CORIA 2009

Actes de la Sixième Conférence Francophone en Recherche d'Information et Applications

5 - 7 mai 2009

Presqu'île de Giens
Var

Organisée par

Université du Sud Toulon-Var

Laboratoire des Sciences de l'Information et des Systèmes
UMR 6168 CNRS

Avec le concours de

ARIA

French Chapter
acm SIGAPP.fr
Special Interest Group on Applied Computing

cnrs I3 orange TOULON PROVENCE MÉDITERRANÉE Communauté d'Agglomération Région PACA

Sixième Conférence Francophone Recherche d'Information et Applications

Nos partenaires

Université du Sud Toulon-Var

Orange Innovation

Laboratoire des Sciences de l'Information et des Systèmes

Centre National de la Recherche Scientifique

GDRI^3

Association Francophone de Recherche d'Information et Applications

Toulon Provence Méditerranée

Conseil régional PACA

ISBN 2-9524747-1-0
EAN 9782952474719
Editeur : LSIS-USTV

Actes de la sixième **CO**nférence francophone en

Recherche d'**I**nformation et **A**pplications

CORIA 2009

5-7 mai 2009

Hyères, France

En couverture : photos du site de la presqu'île de Giens

ISBN : 2-9524747-1-0
EAN : 9782952474719

Avril 2009

Editeur : LSIS-USTV
BP 20132
83957 La Garde - Cedex 20
France

Sommaire

Préface

Comités

Conférenciers invités

Collaborative searching : Social searching, together 1
Alan F. Smeaton

Mining opinions in blogs, the good, the bad and the ugly 2
Marie-Francine Moens

Traitement du langage et résumé automatique

Utilisation de la syntaxe pour valider les réponses à des questions par plusieurs documents .. 5
Véronique Moriceau, Xavier Tannier, Brigitte Grau

Evaluation de diverses stratégies de désambiguïsation lexicale 19
Claire Fautsch, Jacques Savoy

Reconnaissance de critères de comparabilité dans un corpus multilingue spécialisé 33
Lorraine Goeuriot, Emmanuel Morin, Béatrice Daille

Recherche d'information multimodale

Une étude de l'impact de la structure sur la recherche multimédia 51
Mouna Torjmen, Karen Pinel-Sauvagnat

Recherche par le contenu dans des documents audiovisuels multilingues 67
Georges Quénot, Tien Ping Tan, Viet Bac Le, Stéphane Ayache, Laurent Besacier, Philippe Mulhem

Utilisation de concepts visuels et de la diversité visuelle pour améliorer la recherche d'images 83
Sabrina Tollari, Marcin Detyniecki, Ali Fakeri-Tabrizi, Christophe Marsala, Massih-Reza Amini, Patrick Gallinari

Modèle de langue visuel pour la reconnaissance de scènes 99
Trong-Ton Pham, Loïc Maisonnasse, Philippe Mulhem, Eric Gaussier

Recherche et distribution de l'information

Clustering en recherche d'information: concentration *vs* distribution de l'information pertinente 115
Sylvain Lamprier, Tassadit Amghar, Bernard Levrat, Frédéric Saubion

Routage sémantique des requêtes dans les systèmes pair-à-pair 131
Taoufik Yeferny, Khedija Arour, Yahya Slimani

Recherche d'information sémantique

Indexation semi-automatique de textes: thésaurus et transducteurs 151
Laurent Kevers

Modèle d'indexation dynamique à base d'ontologies 169
Gilles Hubert, Josiane Mothe, Bachelin Ralalason, Bertin Ramanonjisoa

Indexation et représentation comparative: application au discours électoral 185
Jacques Savoy

Apprentissage et *clustering*

Catégorisation automatique de pages web chinoises, documents spécialisés *vs* grand public sur le tabagisme 203
Guiyao Ke, Pierre Zweigenbaum

Apprentissage supervisé pour la catégorisation de documents manuscrits en-ligne 219
Sebastian Peña Saldarriaga, Emmanuel Morin, Christian Viard-Gaudin

SRI à base d'inclusion graduelle 235
Laurent Ughetto, Olivier Pivert, Vincent Claveau, Patrick Bosc

Interactions entre le calcul de collocations et la catégorisation automatique de textes 251
Rémi Lavalley, Patrice Bellot, Marc El-beze

Rôle de la matrice d'information et pondération des composantes dans les noyaux de Fisher pour PLSI 267
Jean-Cédric Chappelier, Emmanuel Eckard

Recherche d'information dans les documents structurés

Identification et structuration hiérarchique des titres dans les documents HTML 285
Thierry Waszak, Claude De Loupy, Patrice Bellot

RI structurée, RI et XML, RI précise 301
Ali Aïtelhadj, Mohamed Mezghiche, Fatiha Souam

Utilisation des liens entre documents structurés pour la recherche d'information 319
Philippe Mulhem, Delphine Verbyst

Impact précoce du poids des balises pour la recherche d'information ciblée 333
Mathias Géry, Christine Largeron, Franck Thollard

Articles courts

GraphDuplex : visualisation simultanée de N réseaux couplés 2 par 2 351
Martine Hurault-Plantet, Elie Naulleau, Bernard Jacquemi

Prise en compte des liens pour améliorer la recherche d'information structurée 363
Mataoui M'hamed, Mohamed Mezghiche

Structure et proximité pour la recherche documentaire 373
Michel Beigbeder

REVISE, un outil d'évaluation précise des systèmes questions-réponses 385
Sarra El Ayari, Brigitte Grau, Anne-Laure Ligozat

Les traces d'interactions humaines, un nouveau domaine d'application pour la RI 397
Gregory Dyke, Michel Beigbeder, Kristine Lund, Jean-Jacques Girardot

Proposition de cadres d'évaluation d'un système de RI personnalisé 409
Mariam Daoud, Lynda Tamine-Lechani, Mohand Boughanem

Recherche d'Entités Nommées dans les journaux radiophoniques par contextes hiérarchique et syntaxique 421
Azeddine Zidouni, Hervé Glotin, Mohamed Quafafou

Introduction de la sémantique d'un document sous le modèle de langage 433
Arezki Hammache, Mohand Boughanem, Rachid Ahmed-ouamer

Survey of the adequate descriptor for content-based image retrieval : Global versus local features 445
Hichem Bannour, Lobna Hlaoua, Ayeb Bechir

Aide à l'interprétation de documents juridiques - Une approche centrée utilisateur 457
Youssouf Saidali, Julien Lecanu, Eric Trupin, Jacques Labiche

Articles Jeunes Chercheurs

Une approche sémantique basée sur l'apprentissage pour la recherche d'image par contenu 471
Hichem Bannour

Recherche contextuelle d'information dans un environnement mobile 479
Ourdia Bouidghaghen

Recherche d'information textuelle et phonétique pour le contrôle de l'étiquetage automatique d'émissions dans un flux télévisuel 487
Camille Guinaudeau

Aggregated search : From information nuggets to aggregated documents information 495
Arlind Kopliku

6IR: un index paramétrable pour les requêtes ramifiées 503
Youen Peron

Classement collaboratif de manuscrits 511
Pierre-Edouard Portier

Extraction des connaissances à partir du Web pour la recherche des images géoréférencées 519
Houda Bouamor

Index des auteurs

Préface

La recherche d'information (RI) ne cesse de se développer depuis une vingtaine d'années, stimulée par la puissance des processeurs, la qualité et le volume croissant des corpus d'évaluation, et par l'énorme gisement d'informations qu'offre Internet. La RI intègre de plus en plus les aspects multimodaux et multilingues des documents, s'étendant du traitement du signal à la linguistique, en passant par les statistiques. Le comité de programme de CORIA 2009 s'est d'ailleurs enrichi de nouvelles compétences sur ces thématiques. Les équipes de RI deviennent de fait interdisciplinaires. Cette édition de CORIA représente un bon panorama de ces recherches en pays francophones (Algérie, Belgique, France, Suisse, Tunisie).

Chacun des 60 articles soumis a été évalué par trois rapporteurs, 21 ont été retenus en présentation longue et 10 en présentation courte. De plus ces actes intègrent 7 articles sélectionnés parmi ceux soumis à la session parallèle Jeunes Chercheurs.

Nous sommes très reconnaissants aux deux conférenciers de renommée internationale, Alan F. Smeaton de l'université de Dublin en Irlande, et Marie-Francine Moens de l'université de Leuven en Belgique, d'avoir accepté notre invitation à exposer leur dernière synthèse.

Nous tenons à remercier :
- les membres du comité de programme pour leur travail de sélection, et leurs suggestions constructives aux auteurs,
- les membres du comité d'organisation pour s'être pleinement investis pour la réussite de cette belle manifestation,
- le trésorier de l'ARIA pour son efficacité,
- les organisateurs de la session Jeunes Chercheurs,
- nos partenaires académiques (USTV, CNRS, GDR I3, LSIS, ARIA), industriel (Orange), et régionaux (Région PACA, CG 83, TPM), pour leur soutien financier, matériel et moral.

Il ne nous reste plus qu'à souhaiter bon vent à CORIA 2009 sur cette magnifique presqu'île de Giens, espérant que cette édition contribuera à renforcer les liens entre les acteurs de la RI.

La Gardo, avril 2009

Hervé Glotin
Président du comité d'organisation

Mohand Boughanem
Président du comité de programme

Comité d'Organisation

Président

Hervé Glotin	LSIS, UMR CNRS, univ. du Sud Toulon-Var

Membres

Frédéric Caudal	LSIS, UMR CNRS, univ. du Sud Toulon-Var
Nicolas Faessel	LSIS, UMR CNRS, univ. Paul-Cézanne
Salam Fraihat	LSIS, UMR CNRS, univ. du Sud Toulon-Var
Jacques Le Maitre	LSIS, UMR CNRS, univ. du Sud Toulon-Var
Elisabeth Murisasco	LSIS, UMR CNRS, univ. du Sud Toulon-Var
Azeddine Zidouni	LSIS, UMR CNRS, univ. du Sud Toulon-Var

Comité de Programme

Président

Mohand Boughanem	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)

Membres

Rachid Ahmed Ouamer	LARI, univ. Tizi Ouzou	(Algérie)
Massih Amini	LIP6, UMR CNRS, univ. P. & M. Curie, Paris	(France)
Laurent Amsaleg	IRISA, CNRS INRIA, Rennes	(France)
Micheline Beaulieu	University of Sheffield	(UK)
Michel Beigbeder	ENS des Mines de Saint-Etienne	(France)
Patrice Bellot	LIA, univ. Avignon	(France)
Abdelmajid Ben Hamadou	MIRACL ISIM, univ. Sfax	(Tunisie)
Catherine Berrut	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Laurent Besacier	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Jean-François Bonastre	LIA, univ. Avignon	(France)
Azedine Boulmakoul	LIM/LIST, univ. Hassan II, Mohammedia	(Maroc)
Sylvie Calabretto	LIRIS, UMR CNRS INSA, Lyon	(France)
Max Chevalier	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Jean-Pierre Chevallet	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Yves Chiaramella	CLIPS-IMAG, univ. J. Fourier, Grenoble	(France)
Boris Chidlovskii	Xerox, Grenoble	(France)
Claude Chrisment	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Vincent Claveau	IRISA, CNRS INRIA, Rennes	(France)
Nathalie Denos	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Ludovic Denoyer	LIP6, UMR CNRS, univ. P. & M. Curie, Paris	(France)
Habiba Drias	USTHB, Alger	(Algérie)
Elöd Egyed-Zsigmond	LIRIS, UMR CNRS INSA, Lyon	(France)
Omar El Beqqali	GRMS2I FSDM, univ. Fès	(Maroc)
Rim Faiz	IHEC de Carthage	(Tunisie)
Sami Faiz	Institut National des Sciences Appliquées & Tech.	(Tunisie)
Jérôme Farinas	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Patrick Gallinari	LIP6, UMR CNRS, univ. P. & M. Curie, Paris	(France)
Eric Gaussier	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Mathias Géry	Université St-Etienne	(France)
Hervé Glotin	LSIS, UMR CNRS, univ. du Sud Toulon-Var	(France)
Cyril Goutte	GTLI ITI, Conseil National de Recherches	(Canada)
Brigitte Grau	LIR-LIMSI, UMR CNRS, ENSIIE, Evry	(France)
Guillaume Gravier	IRISA, CNRS INRIA, Rennes	(France)
Patrick Gros	IRISA, CNRS INRIA, Rennes	(France)
Christopher Kermorvant	A2IA.COM R&D, Paris	(France)
Mounia Lalmas	Queen Mary univ. of London	(UK)
Edmond Lassalle	Orange Labs R&D	(France)
Jacques Le Maitre	LSIS, UMR CNRS, univ. du Sud Toulon-Var	(France)
Lynda Lechani-Tamine	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Sébastien Marcel	IDIAP EPFL, Martigny	(Suisse)
Marie-Francine Moes	Katholieke Universiteit Leuven	(Belgique)
Josiane Mothe	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Philippe Mulhem	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Adeline Nazarenko	LIPN, univ. Paris-Nord	(France)
Jian-Yun Nie Rali	Université de Montréal	(Canada)
Iadh Ounis	University of Glasgow	(UK)
Jean-Marie Pinon	LIRIS, UMR CNRS, INSA, Lyon	(France)
Benjamin Piwowarski	University of Glasgow	(UK)
Andrei Popescu-Belis	IDIAP EPFL, Martigny	(Suisse)
Bruno Pouliquen	JRC, Commission Européenne	(Italie)
Georges Quenot	LIG, UMR CNRS, univ. J. Fourier, Grenoble	(France)
Catherine Roussey	LIRIS, UMR CNRS, INSA, Lyon	(France)
Béatrice Rumpler	LIRIS, UMR CNRS, INSA, Lyon	(France)
Jacques Savoy	Université de Neuchâtel	(Suisse)
Florence Sedes	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Malika Smail-Tabbone	LORIA, UMR INRIA, univ. H. Poincaré, Nancy	(France)
Chantal Soule-Dupuy	IRIT, UMR CNRS, univ. P. Sabatier, Toulouse	(France)
Isabelle Tellier	LIFL, UMR CNRS, Lille	(France)
Mohamed Tmar	ISIM, univ. Sfax	(Tunisie)
Tanguy Urvoy	France Telecom R&D	(France)
Pierre Zweigenbaum	LIR-LIMSI, UMR CNRS, ENSIIE, Evry	(France)

Conférences invitées

Collaborative Searching: Social Searching, Together

Alan F. Smeaton
CLARITY Centre for Sensor Web Technologies
Dublin City University Glasnevin,
Dublin 9, Ireland

Information Retrieval (IR) is typically an individual pursuit where an individual searcher will engage with a search system, working alone, until their information need is satisfied. Yet in the real world there are many scenarios, both work-related and related to leisure, entertainment or hobbies, where we want to search as part of a team, maybe even a group of only two people. Collaborative Information Retrieval (CIR) refers to technologies which support collaboration in the retrieval process. In this presentation we will present both synchronous and asynchronous CIR as well as covering remote and co- located search, and the various combinations of these. In our work we are particularly interested in synchronous collaborative IR (SCIR) where a group of users work collectively to address some shared information need. We describe two systems we have developed to demonstrate SCIR, one on a gesture-based tabletop computer and the other on touch-based mobile devices (iPODs). We believe SCIR to be an important kind of social search even though the tools to support this are neither widespread nor reliable and are limited by the technology we currently use. Despite this we expect the importance of SCIR to grow as a consequential fallout of growth in social networks and the trend towards social networks now acting as platforms for applications, like search.

Conférences invitées

Mining Opinions in Blogs, The Good, the Bad and the Ugly

Marie-Francine Moens
Universiteit Leuven,
Belgium

Blogs offer a variety of opinions on politics, consumer products, persons and other subjects. We cannot neglect this wealth of information. No wonder that there are multiple research and commercial efforts to make this information accessible. In a first part the lecture gives an overview of state of art technologies, their results and their applications. In a second part we describe and discuss our own research in opinion mining and focus on the following problems. As blogs interweave many different opinions expressed towards a variety of targets, it is important to correctly attribute a certain opinion to the right subject of discussion. Moreover, the language used in blogs is very diverse demanding automated methods for efficiently selecting examples when acquiring or annotating the necessary opinion patterns (e.g., use of active learning techniques). Related to the foregoing is that blog languages diverge largely from standard language and evolve continuously. We need here very adaptive mining systems. In a third part of the lecture, we go deeper into the current phenomenon of spamming and manipulation of opinions and how text mining could be of help in order to protect sensitive users such as children. Throughout the lecture, we point to many promising research avenues.

Chapitre 1

Traitement du Langage et Résumé Automatique

Utilisation de la syntaxe pour valider les réponses à des questions par plusieurs documents.

Véronique Moriceau[*,**] — **Xavier Tannier**[*,**] — **Brigitte Grau**[*,***]

[*] *LIMSI-CNRS, Orsay, France*

[**] *Université Paris-Sud 11, Orsay, France*

[***] *ENSIIE, Evry, France*

RÉSUMÉ. *Cet article présente FIDJI, un système de questions-réponses pour le français, combinant des informations syntaxiques sur la question et les documents avec des techniques plus traditionnelles du domaine, telles que la reconnaissance des entités nommées et la pondération des termes. Notamment, nous expérimentons dans ce système la validation des réponses dans plusieurs documents, ainsi que des techniques spécifiques permettant de répondre à différents types de questions (comme les questions attendant des réponses multiples (liste) ou une réponse booléenne).*

ABSTRACT. *This article presents FIDJI, a question-answering system for French, combining syntactic information with traditional QA techniques such as named entity recognition and term weighting. Among other uses of syntax, we experiment in this system the validation of answers through different documents, as well as specific techniques for answering different types of questions (*e.g. *yes/no or list questions).*

MOTS-CLÉS : *Systèmes de questions-réponses, analyse syntaxique, validation des réponses.*

KEYWORDS: *Question-answering, syntactic analysis, answer validation.*

1. Introduction

Cet article présente le système FIDJI[1] (Finding In Documents Justifications and Inferences), un système de questions-réponses en domaine ouvert pour le français.

Le projet dans son ensemble a pour objectif de valider des réponses en vérifiant que toutes les informations données dans une question sont bien retrouvées dans les passages de texte supportant la réponse. Cette vérification repose sur le fait de retrouver les différentes entités précisées dans la question correctement reliées entre elles, soit dans une même phrase, un même passage, ou dans plusieurs documents.

Lorsqu'une information est recherchée, elle peut être présente sous différentes formulations et peut requérir, pour être reconnue, l'utilisation de bases de connaissances sémantiques et la réalisation d'inférences avec plusieurs pas de raisonnement pour relier les différentes parties de la réponse. Or, si l'on dispose bien de bases lexicales contenant des variations sur les termes (synonymes par exemple), des bases conceptuelles permettant de relier les concepts pour décrire des événements ne sont pas disponibles pour le français et on ne peut donc envisager une analyse sémantique.

Aussi, notre propos consiste à retrouver les informations précisées par la question en se reposant sur la syntaxe, et notamment la reconnaissance des dépendances entre syntagmes et leurs différentes paraphrases possibles pour identifier les relations cherchées. Nous décrivons dans cet article le fonctionnement général du système, de l'analyse des questions à l'extraction de la réponse. Nous détaillons et évaluons notamment une technique permettant de valider une réponse à travers plusieurs documents.

L'article présente tout d'abord la stratégie générale mise en œuvre dans le système FIDJI (section 2), puis l'analyse syntaxique (section 3) et l'extraction et la justification de la réponse (section 4). Enfin, la section 5 décrit une évaluation et en étudie les résultats.

2. Présentation de FIDJI

La plupart des systèmes de questions-réponses peut extraire une réponse à une question factuelle quand celle-ci est explicitement présente dans le texte, mais ils ne sont pas capables de combiner plusieurs informations pour produire une réponse. FIDJI (Finding In Documents Justifications and Inferences), un système de questions-réponses en domaine ouvert pour le français, vise à introduire des mécanismes de compréhension reposant sur des inférences.

L'objectif est de produire des réponses qui sont entièrement validées par des extraits de textes (des passages).

1. Ce travail est financé partiellement par le projet CONIQUE ANR-05-BLAN-008501 et par OSEO dans le cadre du programme Quaero.

La principale difficulté est qu'une réponse (ou des informations composant une réponse) peut être validée par plusieurs documents. Par exemple :
Question : Quel premier ministre français s'est suicidé ?
Réponse : Pierre Bérégovoy
Passage 1 : Le premier ministre français Pierre Bérégovoy a mis en garde Bill Clinton contre...
Passage 2 : Deux ans plus, Pierre Bérégovoy s'est suicidé après avoir été impliqué...

Dans cet exemple, la réponse *Pierre Bérégovoy* ne peut être entièrement validée par un seul passage : les informations *premier ministre français* et *s'est suicidé* sont validées par deux passages différents. En fait, la réponse est ici l'intersection de deux ensembles de réponses obtenues aux deux questions *Qui s'est suicidé ?* et *Qui sont les premiers ministres français ?* Dans ce cas, pour pouvoir proposer une réponse entièrement validée, il est nécessaire de décomposer la question en deux sous-questions.

Une analyse syntaxique peut fournir de telles décompositions pour les questions. Beaucoup de systèmes de questions-réponses utilisent des informations syntaxiques, en particulier les relations de dépendance, principalement pour l'extraction des réponses. Deux approches émergent : la première consiste à rechercher un appariement exact entre les relations de dépendance de la question et celles d'un passage [KAT 03], tandis que la seconde approche calcule une distance d'édition entre les arbres représentant la question et le passage [LIG 07].
[KAT 05] propose une stratégie pour décomposer les questions à un niveau syntaxique et sémantique : ceci permet à leur système de rechercher des informations dans plusieurs ressources. Il utilise un certain nombre d'"annotations paramétrées" et de patrons sémantiques appliqués à toute la collection de documents afin de relier une question aux informations d'un ou plusieurs documents. Ce système est principalement construit pour répondre aux questions portant sur des objets ou des propriétés (par exemple, une date de naissance, la population d'une ville, etc.). Le système de questions-réponses IRSAW pour l'allemand [HAR 08] adopte lui aussi une stratégie de décomposition des questions en s'appuyant sur un analyseur syntactico-sémantique.

Notre but est d'extraire et de valider des réponses en allant au-delà d'un appariement syntaxique exact entre la question et le passage, et cela sans utiliser de ressources sémantiques. Dans ce contexte de validation de réponse, nous avons remarqué que la stratégie de validation à appliquer (validation par un seul ou plusieurs documents) peut être guidée par la question, et en particulier par le type de réponse attendu. En effet, beaucoup de questions factuelles attendent une réponse d'un type qui peut être :

– une entité nommée comme dans *Qui a obtenu le Prix Nobel de la paix en 1995 ?* qui attend une réponse de type PERSONNE ;

– un type plus précis comme dans *Quel président russe a assisté au G7 en 2007 ?*, qui attend aussi une réponse de type PERSONNE mais dont le type est précisé explicitement dans la question (*président russe*). Le type précis n'est pas issu d'une liste

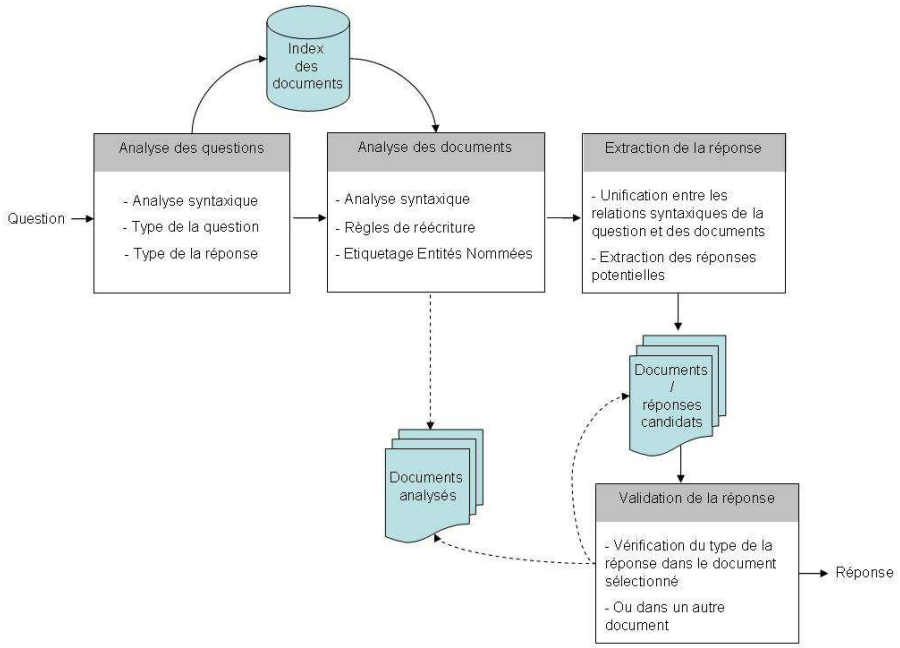


Figure 1. *Architecture de FIDJI*

prédéfinie : il est identifié automatiquement lors de l'analyse de la question.

A la vue des différents exemples étudiés, nous faisons l'hypothèse que le type de la réponse est un élément qui peut être validé dans des documents autres que ceux d'où la réponse est effectivement extraite. Dans cet article, nous présentons cette stratégie. Ainsi, FIDJI utilise des informations syntaxiques, en particulier des relations de dépendances, qui vont permettre notamment de décomposer les questions. Le but est alors de vérifier la présence des relations syntaxiques de la question dans un passage justificatif et de confirmer le type de la réponse potentielle dans ce même passage ou dans un autre document, ceci afin de valider entièrement la réponse. La figure 1 présente l'architecture de FIDJI.

3. Analyse des questions et des documents

Notre système s'appuie sur les analyses syntaxiques produites par l'outil Syntex [BOU 00], un analyseur robuste pour le français. Syntex est utilisé pour analyser à la fois les questions et la collection de documents d'où sont extraites les réponses.

3.1. *Analyse des questions*

L'analyse des questions consiste à identifier :

– les dépendances syntaxiques : elles sont fournies par Syntex ;

– le type de la question : factuelle, définition, booléenne, liste ;

– le type de réponse attendu : soit une entité nommée et/ou un type plus précis.

Les questions attendant des réponses de type liste sont celles qui contiennent explicitement un type de réponse au pluriel (par exemple *Quelles planètes… ?*, *Qui sont les… ?*, etc.).
Lors de l'analyse syntaxique de la question, la réponse à extraire est représentée par une variable (notée REPONSE) introduite dans les relations de dépendance. Le type de la question est principalement déterminé grâce à la forme de la question (pronom interrogatif, etc.). Le type spécifié de réponse attendu, s'il existe, est quant à lui le groupe nominal lié à la variable réponse par une relation *attribut* (équivalent à une relation *est_un*). Par exemple :

Q : Quand fut construite la Tour Eiffel ?
Dépendances : `DATE(REPONSE)`
`SUJ(construire, Tour Eiffel)`
`AUX(construire, être)`
Type de la question : factuelle
Type de réponse attendu : DATE (entité nommée)

Q : Quelle déclaration fut adoptée par l'ONU en 1948 ?
Dépendances : `attribut(REPONSE, déclaration)`
`AUX(être, adopter)`
`SUJ(adopter, REPONSE)`
`modif_par(adopter, ONU)`
`DATE(1948)`
Type de la question : factuelle
Type de réponse attendu : déclaration (type plus précis)

Dans le premier exemple, le type de la réponse est une entité nommée alors que dans le second exemple, c'est un mot ou groupe de mots directement extrait de l'analyse syntaxique de la question.

3.2. *Analyse des documents*

Notre approche consiste à déterminer, pour une question donnée, si toutes les caractéristiques de la question (en l'occurrence les dépendances syntaxiques) peuvent être trouvées dans un ou plusieurs documents. Dans ce but, FIDJI détecte les implica-

tions syntaxiques entre la question et les passages. Toute la collection de documents est donc également analysée syntaxiquement par Syntex.

Comme les informations dans les documents ne sont pas toujours exprimées de la même façon que dans les questions (par exemple, par le biais de variations syntaxiques), il est indispensable de raisonner sur les relations de dépendance syntaxique. De la même façon que [BOU 07], nous avons implémenté environ 40 règles de réécriture pour tenir compte, entre autres, des variations comme les changements voix passive/active, les nominalisations de verbes [JAC 96], les appositions, coordinations, etc. Ces règles de réécriture sont appliquées à toute la collection analysée.

De cette façon, quelle que soit la forme syntaxique de la question, le système est capable de trouver une formulation équivalente dans un passage justificatif : il est ainsi possible d'avoir un appariement exact soit entre les dépendances syntaxiques de la question et d'un passage, soit entre les dépendances syntaxiques de la question et celles d'un passage obtenues par réécriture.

L'exemple suivant illustre la règle de réécriture pour la reformulation passif/actif (les relations obtenues par réécritures sont en italique).

Q : Quelle ville a été secouée par un tremblement de terre le 17 janvier ?

```
DATE(17 janvier)
attribut(REPONSE, ville)
SUJ(secouer, REPONSE)
AUX(être, secouer)
modif_par(secouer, tremblement)
attribut_de(tremblement, terre)
```

Text : Le tremblement de terre qui a secoué, lundi 17 janvier, Los Angeles ne serait pas associé directement à la fameuse faille de San-Andreas...

```
SUJ(secouer, tremblement)
OBJ(secouer, Los Angeles)
SUJ(secouer, Los Angeles)
AUX(être, secouer)
modif_par(secouer, tremblement)
attribut_de(tremblement, terre)
DATE(17 janvier) ...
```

Les relations de dépendance réécrites sont obtenues par application de la règle de la figure 2. Dans cet exemple, toutes les relations de la question sont présentes dans l'analyse du passage.

```
SUJ(Verbe, NP1)        SUJ(Verbe, NP2)
OBJ(Verbe, NP2)   ⇒    AUX(être, Verbe)
                       modif_par(Verbe, NP1)
```

Figure 2. *Exemple de règle de réécriture : voix active vers voix passive.*

3.3. *Autres ressources : les entités nommées*

Les entités nommées des documents sont étiquetées en utilisant un ensemble d'environ 160 types [ROS 07] (*e.g.* personne, organisation, lieu, nationalité, date, nombre, etc.). Cet étiquetage, associé aux résultats de l'analyse de la question, est utile pour vérifier que le type d'entité nommée attendu par la question est bien le même que celui de la réponse extraite dans le document. Par exemple, la question suivante attend une réponse de type LIEU :

Question : Où Barbara Hendricks a-t-elle donné son premier concert de l'année ?

```
LIEU(REPONSE)
SUJ(donner, Barbara Hendricks)
OBJ(donner, concert)
attribut_de(concert, année)
```

Passage étiqueté : <pers>Barbara Hendricks</pers> a donné son premier concert de l'Année nouvelle à <ville>Sarajevo</ville>.

```
SUJ(donner, Barbara Hendricks)
OBJ(donner, concert)
attribut_de(concert, année)
...
```

Dans cet exemple, toutes les relations de dépendance syntaxique de la question sont présentes dans le passage et le type de réponse attendu (LIEU) correspond bien au type de la réponse extraite *Sarajevo* (le type VILLE est un sous-type de LIEU dans la hiérarchie des entités nommées).

4. Recherche de la réponse

4.1. *Sélection des passages*

Les documents sont indexés par le moteur de recherche Lucene [HAT 04]. Les mots-clés utilisés pour interroger l'index sont l'ensemble des mots significatifs de la

question[2] ; les 100 premiers documents sont conservés (ainsi, seuls ces documents sont considérés par la suite).

4.2. *Extraction de la réponse*

La stratégie employée pour l'extraction de la réponse dépend du type de la question. L'utilisation de l'analyse syntaxique se situe au niveau de la phrase. Les informations syntaxiques, lorsqu'elles sont utiles, sont combinées avec d'autres paramètres.

4.2.1. *Questions factuelles*

Les phrases candidates sélectionnées sont celles qui comptent le plus de relations en commun avec la question. Une fois ces phrases obtenues, deux cas sont à considérer :

1) Au moins une dépendance de la question comportant la variable 'REPONSE' s'unifie avec une dépendance de la phrase. Dans ce cas, cette variable est instanciée par un lemme, qui est considéré comme la tête de la réponse. La réponse complète est ensuite composée en ajoutant les modifieurs présents dans la phrase (compléments du nom et adjectifs). Si cette réponse est du type d'entité nommée attendu, un poids plus important lui est attribué (voir la section 4.4.1).

2) Si la variable 'REPONSE' ne trouve pas d'instanciation dans la phrase, des éléments ayant le type d'entité nommée approprié sont recherchés. Ceci a pour but de contrebalancer les erreurs d'analyse. Il arrive bien entendu régulièrement que la réponse soit présente dans le passage mais que les seules relations syntaxiques n'y conduisent pas.

Dans le cas d'absence d'entité nommée du type recherché dans la phrase considérée, elles peuvent être collectées dans les 2 phrases précédentes (avec un score moindre). Cette heuristique compense (très imparfaitement) l'absence d'une résolution anaphorique dans le système ; ceci produit inévitablement un certain bruit, mais qui est généralement masqué par la redondance des bonnes réponses (la fréquence d'extraction d'une réponse augmente en effet son score – section 4.4.1).

4.2.2. *Questions listes*

L'analyse de Syntex, enrichie par certaines des règles de réécriture, fournit des informations sur la coordination des syntagmes.

Si une réponse est trouvée (par les techniques indiquées à la section précédente), les éventuelles relations de coordination avec cette réponse sont recherchées, et une liste est construite à partir des éléments ainsi collectés.

---

2. C'est-à-dire les mots étiquetés comme nom, verbe, adjectif ou adverbe par l'analyseur syntaxique.

Si les réponses multiples sont préférées, les réponses atomiques sont retournées malgré tout ; en effet, une question étiquetée “liste” peut être satisfaite par un autre moyen. Par exemple, à la question *Qui sont les Dalton ?*, le pluriel indique certes que l’on attend une liste de noms, mais la réponse *“4 bandits”* est également tout à fait valable.

4.2.3. *Questions booléennes*

Répondre à des questions booléennes est relativement proche de la tâche de validation de réponse, comme pratiquée par exemple lors des campagnes AVE (Answer Validation Exercise [ROD 08]). Dans cette tâche, les participants doivent considérer des questions, des réponses de systèmes et un texte de justification, et décider si la réponse à la question est à la fois correcte et validée entièrement par le texte.

Nous avons participé à AVE en langue française en 2008 : FIDJI s’est classé premier pour le français et second sur l’ensemble des candidats toutes langues confondues [MOR 08].

Nous utilisons la même technique pour les questions booléennes : si la part de dépendances de la question trouvées dans la phrase dépasse un certain seuil (déterminé empiriquement à 70 %), la réponse retournée est ’oui’.

4.3. *Validation du type de la réponse*

Lorsqu’une phrase est sélectionnée et une réponse extraite, il arrive souvent que toutes les relations recherchées ne soient pas satisfaites. Dans certains cas, il est alors possible de vérifier la présence des dépendances manquantes dans d’autres documents. Dans l’état actuel du système, la seule validation effectuée par ce biais est celle du type précis de la réponse.

Ce type est fourni, lorsqu’il existe, par l’analyse de la question (voir section 3.1). Si le passage justificatif ne contient pas les informations permettant de le vérifier, une nouvelle question est construite pour valider la réponse candidate.

Dans notre exemple *“Quel premier ministre s’est suicidé...”*, le type de réponse est *ministre* (tandis que le type d’entité nommée est PERSONNE). Le type *étendu* est *premier ministre*.

Q : Quel premier ministre s’est suicidé en 1993 ?
Dépendances : `SUJ(se suicider, REPONSE)`
`DATE(se suicider, 1993)`
`attribut(REPONSE, ministre)`
`attribut(ministre, premier)`

Si on retrouve dans un texte la phrase *Pierre Bérégovoy s’est suicidé en 1993*, la variable REPONSE s’unifie avec *Pierre Bérégovoy* qui devient une réponse candi-

date : les deux premières dépendances de la question sont ainsi vérifiées dans cette phrase. Il manque les deux suivantes, concernant le type précis (Pierre Bérégovoy était-il premier ministre ?).

La validation est opérée en deux étapes. Tout d'abord, le système vérifie que la réponse candidate est bien un *ministre*, en recherchant une relation nommée 'attribut' (`attribut(Bérégovoy, ministre)`). Si cela est confirmé, le type étendu est également vérifié et les deux relations `attribut(Bérégovoy, ministre)` et `attribut(français, ministre)` sont attendues dans la même phrase.

4.4. *Justification et classement des réponses candidates*

4.4.1. *Classement des réponses*

À chaque réponse est associé un quadruplet :

– la présence de tous les noms propres de la question dans la phrase (0 ou 1),

– la réponse a le bon type d'entité nommée (0 ou 1),

– la réponse a le bon type précis (0, 1 pour le type simple, 2 pour le type étendu),

– le nombre de fois où la réponse a été extraite.

Les critères suivants sont utilisés pour classer les réponses (du plus important au moins important) :

1) Si le passage justificatif ne contient pas tous les noms propres de la question, alors la réponse est disqualifiée ;

2) Une réponse possédant le bon type d'entité nommée est préférée (quels que soit les scores ci-dessous) ;

3) Une réponse ayant validé le type précis est préférée également (le type étendu étant idéal) ;

4) Enfin, si tous les critères ci-dessus sont équivalents, une réponse trouvée plusieurs fois additionne les poids obtenus par chaque occurrence. La redondance est une information importante qui permet de masquer bon nombre d'erreurs.

4.4.2. *Passage justificatif*

Le passage justificatif est produit de la façon suivante :

– Pour chaque réponse, la phrase ayant le meilleur score est sélectionnée.

– Les phrases précédentes sont également inclues dans le passage (dans le but de collecter le contexte et d'éventuels antécédents anaphoriques) dans une limite de 256 caractères (limite classique des campagnes d'évaluation).

CLEF 2005		
Type de question	**FIDJI**	QRISTAL
Factuelle	**53%**	59%
Définition	**78%**	86%
TOTAL	**59.5%**	64%
CLEF 2006		
Type de question	**FIDJI**	QRISTAL
Factuelle	**46%**	64%
Définition	**56.5%**	83%
Liste	**50%**	50%
TOTAL	**48.5%**	68%

Tableau 1. *Résultats de FIDJI sur les données de CLEF 2005 et 2006.*

5. Evaluation et résultats

Nous avons évalué notre système sur les données de test des campagnes d'évaluation CLEF 2005 et 2006 [VAL 05, MAG 06]. La collection de documents est composée d'environ 177000 articles de journaux en français (Le Monde et ATS 1994-1995 (environ 2 Go)). Ces documents sont censés être syntaxiquement corrects. Les questions de CLEF 2005 sont factuelles ou de type définition tandis que les questions de CLEF 2006 sont factuelles, de type définition ou liste.
Lors de ces campagnes, les sytèmes sont autorisés à proposer 3 réponses pour chaque question, ces réponses devant être classées par ordre de confiance.

Plusieurs points doivent être évalués :

– les performances du système sur différents types de questions,

– l'apport de la stratégie de décomposition des questions dans le but de valider une réponse grâce à plusieurs passages.

5.1. *Evaluation de FIDJI*

Le tableau 1 présente les résultats obtenus par FIDJI sur les données de CLEF 2005 et 2006 (nombre de réponses correctes proposées en première position). Nous comparons ces résultats avec ceux de QRISTAL [LAU 05, LAU 06], le meilleur système de questions-réponses pour le français lors de ces campagnes.

Le tableau 2 précise le rang des réponses correctes proposées par FIDJI.

FIDJI obtient de moins bons résultats sur les données de CLEF 2006. Ceci est dû au fait que de nouveaux types de questions ont été introduits lors de CLEF 2006 : des questions de type définition et liste. En effet, pour les questions de type défini-

Position de la réponse correcte	Rang 1	Rang 1 à 3
CLEF 2005	59.5%	68.5%
CLEF 2006	48.5%	57%

Tableau 2. *Position des réponses correctes.*

tion, de nouvelles catégories (par exemple, des définitions d'objets : *Qu'est-ce qu'un t-shirt ?*) ont été ajoutées afin de réduire le nombre de questions auxquelles il est facile de trouver une réponse (par exemple, les acronymes (*Qu'est-ce que RKA ?*) et les descriptions de personnes (*Qui est Bill Clinton ?*) sont souvent des appositions aux noms propres) [MAG 06]. Une autre difficulté se situe au niveau de l'identification des questions de type liste : des questions telles que *Citez le nom de tous les aéroports de Londres* sont facilement identifiées par les systèmes comme étant des questions de type liste alors que des questions telles que *Qui a découvert la comète Shoemaker-Levy ?* sont plus difficiles à analyser.

Les résultats de FIDJI sont inférieurs à ceux de QRISTAL qui utilise aussi une approche syntaxique mais qui bénéficie de l'utilisation de nombreuses ressources telles que des dictionnaires et des ontologies. Cependant, notre système se place "virtuellement" à la deuxième place de ces campagnes d'évaluation puisque les systèmes qui ont atteint la deuxième place ont obtenu un score de 35% de réponses correctes pour CLEF2005 et de 46% pour CLEF2006.

5.2. *Evaluation de la stratégie de décomposition des questions*

Dans l'ensemble de questions de CLEF 2005, il y a 51 questions (25%) qui peuvent être décomposées en sous-questions en appliquant notre stratégie (à noter que seulement 17% des questions sont décomposables pour [HAR 08], qui utilise aussi une stratégie de décomposition similaire). FIDJI trouve une réponse correcte (sans tenir compte du rang) pour 32 questions. Chaque fois que la décomposition des questions est appliquée, le système peut rechercher des justifications à la réponse dans des documents différents.

Parmi les 32 réponses correctes pour les questions qui ont été décomposées en sous-questions, 22% d'entre elles ont une justification dans plusieurs documents. En revanche, si FIDJI n'utilise pas la stratégie de décomposition, seulement 64% des réponses sont correctes (au lieu de 68.5%).

Dans l'ensemble de questions de CLEF 2006, 56 questions peuvent être décomposées. FIDJI trouve une réponse correcte pour 29 d'entre elles : 14% de ces réponses correctes ont une justification dans plusieurs documents. Si FIDJI n'utilise pas la stratégie de décomposition, seulement 55% des réponses sont correctes (au lieu de 57%).

Le tableau 3 montre le nombre de réponses correctes dont la justification a été trouvée dans un ou plusieurs documents.

Parmi 32 réponses correctes	**CLEF 2005**
Justification dans 1 document	78%
Justification dans plusieurs documents	22%
Parmi 29 réponses correctes	**CLEF 2006**
Justification dans 1 document	86%
Justification dans plusieurs documents	14%

Tableau 3. *Evaluation de la stratégie de décomposition des questions.*

Ces résultats montrent une amélioration des performances du système lorsque l'on utilise une stratégie de décomposition des questions afin de valider les réponses par l'intermédiaire de plusieurs documents.

6. Conclusion

Dans cet article, nous avons présenté comment l'utilisation de la syntaxe peut aider un système de questions-réponses à produire de bons résultats. Les performances de notre système, évalué sur les collections CLEF, sont proches des meilleurs systèmes, et ceci sans avoir recours à de nombreuses ressources externes exploitées par ces systèmes. Les résultats montrent aussi que la validation de réponses par plusieurs documents améliore le traitement de certains types de questions. D'autres expérimentations sont en cours pour mesurer l'apport exact de l'analyse syntaxique dans chaque module du système.

Nous allons maintenant nous attacher à tester notre système sur une collection de documents provenant du Web dans le contexte du projet Quaero[3]. Un work package de Quaero est en effet consacré aux questions-réponses sur le Web. Un corpus de 2 millions de pages a été créé, et les questions sont définies et évaluées par un partenaire indépendant. Des questions de type liste, booléennes et complexes ('pourquoi', 'comment') seront proposées. Cette étude aura pour but d'étudier si des techniques de TAL peuvent être appliquées avec succès à de très grandes collections composées de documents de styles très différents. La taille de la collection de documents étant trop importante, nous modifions actuellement l'architecture afin d'éviter d'analyser entièrement le corpus.

Par ailleurs nous nous proposons aussi d'étudier la vérification de relations manquantes autres que le type de la réponse. Cela nécessitera sans doute de contextualiser la recherche afin de contrôler les inférences recherchées. Nous allons pour cela utiliser un nouvel ensemble de questions plus approprié à la recherche multi-documents.

3. http ://www.quaero.org

Véronique Moriceau, Xavier Tannier, Brigitte Grau.

7. Bibliographie

[BOU 00] BOURIGAULT D., FABRE C., « Approche linguistique pour l'analyse syntaxique de corpus », *Cahiers de Grammaire*, vol. 25, 2000, Université Toulouse Le Mirail.

[BOU 07] BOUMA G., FAHMI I., MUR J., VAN NOORD G., VAN DER PLAS L., TIEDEMANN J., « Linguistic knowledge and question answering », *Traitement automatique des langues*, vol. 46, 2007, Hermes-Lavoisier.

[FOR 08] FORNER P., PEÑAS A., ALEGRIA I., FORASCU C., MOREAU N., OSENOVA P., PROKOPIDIS P., ROCHA P., SACALEANU B., SUTCLIFFE R., SANG E. T. K., Eds., *Working Notes for the CLEF 2008 Workshop*, Aarhus, Denmark, septembre 2008.

[HAR 08] HARTRUMPF S., GLÖCKNER I., LEVELING J., « University of Hagen at QA@CLEF 2008 : Efficient Question Answering with Question Decomposition and Multiple Answer Streams », Forner et al. [FOR 08].

[HAT 04] HATCHER E., GOSPODNETIĆ O., *Lucene in Action*, Manning, 2004.

[JAC 96] JACQUEMIN C., « A symbolic and surgical acquisition of terms through variation », *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, Heidelberg, 1996.

[KAT 03] KATZ B., LIN J., « Selectively using relations to improve precision in Question Answering », *Proceedings of workshop on Natural Language Processing for Question Answering, EACL*, Budapest, 2003.

[KAT 05] KATZ B., BORCHARDT G., FELSHIN S., « Syntactic and Semantic decomposition Strategies for Question Answering from Multiple Resources », *Proceedings of the AAAI 2005 Workshop on Inference for Textual Question Answering*, Pittsburgh, 2005.

[LAU 05] LAURENT D., SÉGUÉLA P., NÈGRE S., « Cross Lingual Question Answering using QRISTAL for CLEF 2005 », *Working Notes QA@CLEF*, Vienna, 2005.

[LAU 06] LAURENT D., SÉGUÉLA P., NÈGRE S., « Cross Lingual Question Answering using QRISTAL for CLEF 2006 », *Working Notes QA@CLEF*, Alicante, 2006.

[LIG 07] LIGOZAT A., « Apport de l'analyse syntaxique des phrases dans un système de questions-réponses », *Traitement automatique des langues*, vol. 46, 2007, Hermes-Lavoisier.

[MAG 06] MAGNINI B., GIAMPICCOLO D., FORNER P., AYACHE C., OSENOVA P., PEÑAS A., JIJKOUN V., SACALEANU B., ROCHA P., SUTCLIFFE R., « Overview of the CLEF 2006 Multilingual Question Answering Track », *Working Notes QA@CLEF*, Alicante, 2006.

[MOR 08] MORICEAU V., TANNIER X., GRAPPY A., GRAU B., « Justification of Answers by Verification of Dependency Relations - The French AVE Task », Forner et al. [FOR 08].

[ROD 08] RODRIGO A., PEÑAS A., VERDEJO F., « Overview of the Answer Validation Exercise 2008 », Forner et al. [FOR 08].

[ROS 07] ROSSET S., GALIBERT O., ADDA G., BILINSKI E., « The LIMSI Qast systems : comparison between human and automatic rules generation for question-answering on speech transcriptions », *ASRU*, Kyoto, 2007.

[VAL 05] VALLIN A., GIAMPICCOLO D., AUNIMO L., AYACHE C., OSENOVA P., PEÑAS A., DE RIJKE M., SACALEANU B., SANTOS D., SUTCLIFFE R., « Overview of the CLEF 2005 Multilingual Question Answering Track », *Working Notes QA@CLEF*, Vienna, 2005.

Evaluation de diverses stratégies de désambiguïsation lexicale

Claire Fautsch, Jacques Savoy

Institut d'informatique
Université de Neuchâtel, rue Emile Argand 11, 2009 Neuchâtel (Suisse)
Claire.Fautsch@unine.ch, Jacques.Savoy@unine.ch

RESUME. *Dans la campagne d'évaluation CLEF-2008, la tâche « robuste » fournissait un corpus enrichi en langue anglaise. Pour chaque mot, le lemme, la partie du discours et le numéro Synsets de WordNet™ (numéro de classe d'un thésaurus) étaient fournis. Sur cette base, nous avons testé plusieurs approches afin de lever, en partie pour le moins, l'ambiguïté lexicale. Recourant au modèle vectoriel tf idf, ainsi qu'à trois approches probabilistes et un modèle de langue, cet article évalue leur performance en fonction de diverses techniques d'enracineur. Un enracineur léger permet d'obtenir des performances similaires à des approches plus agressives ou à celle obtenue par une analyse morphologique. L'indication de la partie du discours permet d'améliorer significativement la qualité de la réponse tandis que les numéros de classes d'un thésaurus n'ont pas permis une amélioration.*

ABSTRACT. *In the robust track of the 2008 CLEF evaluation campaign an enlarged English corpus was provided. For each term, the lemma, the part-of-speech (POS) and the Synset number extracted from WordNet™ (class number of the corresponding thesaurus) are given. Based on this corpus we tested several approaches to remove at least partially the underling lexical ambiguity. Using different IR models such as the vector-space model tf idf as well as three probabilistic models and a language model, we want to evaluate their performance when using different algorithmic or morphological stemming approaches. The inclusion of the part-of-speech information improves the retrieval performance significantly, while the inclusion of the synset number does not show any improvement.*

MOTS-CLES : *Analyse morphologique, enracineur, thésaurus, partie du discours, évaluation.*

KEY WORDS: *Morphological Analysis, Stemmer, Thesaurus, Part-Of-Speech, Evaluation.*

Claire Fautsch, Jacques Savoy

1. Introduction

En recherche d'information (RI), l'emploi d'enracineurs représente une technique fréquente qui, en général, améliore la qualité des résultats obtenus [MAN 08]. Le but recherché est de permettre des appariements entre des formes sémantiquement reliées mais dont l'orthographe varie. Par exemple, si une requête contient le mot "chat" et le document renferme la forme "chats", l'enracineur devrait regrouper ces deux formes de surface sous la même entrée dans l'index de même que pour les formes "chattes" ou "chaton".

Pour la langue anglaise, différents enracineurs algorithmiques ont été proposés basés sur des règles morphologiques de la langue comme, par exemple, celui de Porter [POR 80] ou Lovins [LOV 68]. Contrairement à ceux-ci, nous pouvons procéder à une analyse morphologique plus profonde, requérant certes plus de ressources (dictionnaire), mais permettant de retourner le lemme du mot (on son entrée dans le dictionnaire). Cette première étape nous permet d'éliminer toutes les marques liées aux flexions. De plus en se basant sur la partie du discours (POS) nous pourrions améliorer la qualité des appariements entre formes identiques (e.g., le sens de "*mean*" comme nom et comme verbe est différent) ou utiliser l'information POS pour mieux contrôler l'élimination des suffixes dérivationnels [SAV 93]. Comme approche supplémentaire pour améliorer le dépistage de documents pertinents, nous pourrions utiliser le numéro de la classe Synset du thésaurus WordNet™ [FEL 98] afin de faciliter l'appariement entre formes différentes mais ayant un sens proche.

Cet article a pour objectif d'évaluer et d'analyser l'impact de divers traitements morphologiques (enracineur, lemmatisation, partie du discours et thésaurus) disponible avec la tâche « robuste » de la campagne d'évaluation CLEF 2008. La suite de cette communication est organisée de la manière suivante. Dans la deuxième section nous décrivons brièvement le corpus utilisé tandis que la troisième section présente les enracineurs et les modèles de dépistage utilisés. Les évaluations faites sont exposées dans la quatrième section et une conclusion résume les principales contributions dans une cinquième section.

2. Regard sur le corpus d'évaluation

La tâche « robuste » de la campagne d'évaluation CLEF-2008 a décidé de former un large ensemble de requêtes en regroupant l'ensemble des collections rédigées en langue anglaise et couvrant les campagnes de 2001 à 2006 (Peters *et al*. 2008). Ce corpus se compose d'articles de journaux du *Los Angeles Times* publiés durant l'année 1994 ainsi que des documents du *Glasgow Herald* parus en 1995. Ce corpus comprend un total de 169 477 documents (correspondant à environ 579 MB de données). En moyenne, chaque article contient environ 250 mots pleins (médiane:

191) (ce calcul ne tient pas compte des mots outils comme “the”, “of” ou “in”). Un document caractéristique possède un titre bref suivi d'un à quatre paragraphes de texte pouvant être rédigés selon l'orthographe anglaise ou américaine.

La figure 1 illustre avec quelques détails les diverses composantes de notre corpus d'évaluation. Par exemple, pour l'année 2003, les requêtes disponibles débutent au numéro 141 et s'achève avec le numéro 200. Sur cet ensemble, nous disposons de 54 interrogations avec au moins un document pertinent. Ces bonnes réponses doivent être dépistées selon l’année dans un ou deux des journaux *Los Angeles Times* et *Glasgow Herald* comme indiquée dans la figure 1. Par exemple en 2004 on ne tient compte que du *Glasgow Herald* tandis qu’en 2005 on recherche l’information dans les deux journaux.

	2001	2002	2003	2004	2005	2006
Source	*LA Times*	*LA Times*	*LA Times Glasgow H.*	*Glasgow H.*	*LA Times Glasgow H.*	*LA Times Glasgow H.*
Taille	425 MB	425 MB	579 MB	154 MB	579 MB	579 MB
# docs	113 005	113 005	169 477	56 472	169 477	169 477
Nb req.	47	42	54	42	50	49
Requête	n^{o}41-n^{o}90	n^{o}91-n^{o}140	n^{o}141-n^{o}200	n^{o}201-n^{o}250	n^{o}251-n^{o}300	n^{o}301-n^{o}350

***Figure 1** : Caractéristiques essentielles des diverses parties de notre corpus d'évaluation*

Suivant le modèle des campagnes TREC, chaque requête possède principalement trois champs logiques, à savoir un titre bref (T), une phrase décrivant le besoin d’information (D) et une partie narrative (N). La figure 2 présente, dans sa partie supérieure, un exemple. Pour l’essentiel de nos évaluations, nous avons retenu uniquement la partie “titre” (T) pour construire les requêtes (pour, en moyenne, 2,91 termes d’indexation par requête) ou les parties “titre” et “descriptif” (TD) (moyenne, 7,51 termes par requête).

Lors de la campagne d’évaluation CLEF 2008, les organisateurs ont ajouté des informations tant au niveau des documents que des requêtes. La partie inférieure de la figure 2 illustre un exemple d’une requête élargie. Ainsi, avec chaque mot, on retrouve sa forme actuelle (sous l'étiquette <WF>), sa partie du discours (étiquette <POS>), son lemme (ou son entrée dans le dictionnaire avec l'étiquette <LEMA>) et finalement le ou les numéros de classes dans le thésaurus WordNet™ (étiquette <SYNSET>). L'ensemble de ces informations a été ajouté afin de mesurer l'efficacité de diverses stratégies pouvant éliminer ou, pour le moins réduire, l'ambiguïté lexicale en recherche d'information.

Ainsi avec le lemme fourni nous pouvons travailler directement avec le résultat d'une analyse morphologique. Le recours à un enracineur léger éliminant les flexions morphologiques s'avère superflu. La partie du discours peut fournir des indications pertinentes pour procéder à la suppression des suffixes de dérivation ou pour favoriser des appariements entre mots de même nature. Le nombre associé à l'entrée dans le thesaurus WordNet™ (version 1.6) permet de définir les synonymes d'un terme. Evidement pour les noms propres (nom de personne, lieu ou de produit comme "Haiti" ou "Kaurismäkis"), cette information n'existe pas. Dans le corpus ce nombre peut être unique si le terme n'apparait que dans une seule classe. Parfois plusieurs numéros de classe du thésaurus sont indiqués avec un score représentant la probabilité que le Synset correspondant soit correct (par exemple, dans la figure 2, deux numéros de classe sont attribués pour le mot "Bankruptcy").

```
<NUM> C180 </NUM>
<EN-TITLE> Bankruptcy of Barings </EN-TITLE>
<EN-DESC> What was the extent of the losses in the Barings bankruptcy case?
 </EN-DESC>
<EN-NARR> Relevant documents must quantify in some way the losses caused by
the collapse of the oldest bank in Great Britain </EN-NARR>
...
<NUM> C180 </NUM>
<EN-TITLE>
 <TERM ID="10.2452/180-AH-1" LEMA = "bankruptcy" POS = "NNP">
 <WF> Bankruptcy </WF>
   <SYNSET SCORE = "0.4819665883771086" CODE = "10386276-n"/>
   <SYNSET SCORE = "0.5180334116228914" CODE = "10386165-n"/> </TERM>
<TERM ID = "10.2452/180-AH-2" LEMA = "of" POS = "IN">
 <WF> of </WF> </TERM>
<TERM ID = "10.2452/180-AH-3" LEMA = "baring" POS = "NNPS">
 <WF> Barings </WF>
 <SYNSET SCORE = "1" CODE = "00819570-n"/> </TERM>
</EN-TITLE>
...
```

***Figure 2** : Exemple d'une requête avec les indications du lemme, de sa partie du discours, et des numéros de classe de WordNet associées*

L'ensemble des ces informations n'a pas été ajouté manuellement mais en recourant à différents traitements automatiques. En premier lieu, le système MXPOST (*Maximum Entropy POS Tagger*[1]) [RAT 96] a été utilisé afin de déterminer la partie du discours (POS) pour chaque terme. Lors d'une deuxième étape, le lemme correspondant est extrait de WordNet™ en utilisant JWNL (*Java WordNet Library*), une API permettant un accès facile au thésaurus. Finalement, en

[1] Téléchargeable sur http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html

se basant sur ces informations, des collocations locales et le contexte, le système de désambiguïsation NUS-PT [CHA 07] détermine la ou les classes du thésaurus correspondant au mot étudié. Cette affectation s'effectue, pour l'essentiel, avec un algorithme de type SVM (*Support Vector Machine*) entrainé sur différents corpus dont des collections d'articles de journaux, type d'information que l'on retrouve dans les collections CLEF. Ces divers traitements sont pas excepte d'erreurs comme le fait que l'étiquette POS associé au mot “Bankruptcy” dans la figure 2 est “NNP” (nom propre) et non “NN” (nom).

3. Modèles de recherche d'information

Afin d’obtenir une vision assez large de la performance de divers traitements lexicaux, nous avons retenus différentes approches. Comme modèle de base, nous avons indexé les documents (et les requêtes) selon la formulation classique $tf \cdot idf$ qui tient compte de la fréquence d'occurrence (ou fréquence lexicale notée tf_{ij} pour le j^e terme dans le i^e document) et de la fréquence documentaire d'un terme (df_j, ou plus précisément de l’$idf_j = \log(n/df_j)$ avec n indiquant le nombre de documents inclus dans le corpus).

L'évaluation avec ce modèle vectoriel sera complétée par celles obtenues par des approches probabilistes. Dans ce cadre, nous avons considéré le modèle Okapi (ou BM25) [ROB 00] utilisant la formulation suivante :

$$w_{ij} = [(k_1+1) \cdot tf_{ij}] / (K + tf_{ij}) \quad \text{avec } K = k_1 \cdot [(1-b) + ((b \cdot l_i) / mean\ dl)] \qquad (1)$$

dans laquelle l_i est la longueur du i^e article (mesurée en nombre de termes d’indexation), et b, k_1 des constantes fixées empiriquement à $b = 0{,}55$, $k_1 = 1{,}2$ et *mean dl* la longueur moyenne d’un document.

Comme autres approches probabilistes, nous avons implémenté le modèle DFR-PL2 et le modèle DFR-I(n_e)C2, issus de la famille *Divergence from Randomness* (DFR) [AMA 02]. Pour ces modèles, la pondération w_{ij} combine deux mesures d'information, à savoir :

$$w_{ij} = Inf^1_{ij} \cdot Inf^2_{ij} = Inf^1_{ij} \cdot (1 - Prob^2_{ij}) \qquad (2)$$

Pour le modèle DFR-PL2, ces deux mesures d’informations sont estimées en recourant à la formulation suivante :

$$Prob^2_{ij} = tfn_{ij} / (tfn_{ij} + 1) \quad \text{avec } tfn_{ij} = tf_{ij} \cdot \ln[1+((c \cdot mean\ dl) / l_i)]$$
$$Inf^1_{ij} = -\log_2[(e^{-\lambda_j} \cdot \lambda_j^{tf_{ij}})/tf_{ij}!] \quad \text{avec } \lambda_j = tc_j / n \qquad (3)$$

dans laquelle tc_j représente le nombre d'occurrences du j^e terme dans la collection et c une constante fixée empiriquement à 1,5.

Le modèle DFR-I(n_e)C2 se base sur la formulation suivante.

$$Prob^{2}_{ij} = 1 - [(tc_j + 1) / (df_j \cdot (tfn_{ij}+1))]$$
$$Inf^{1}_{ij} = tfn_{ij} \cdot \log_2[(n+1) / (n_e+0{,}5)] \text{ avec } n_e = n \cdot [1 - [(n-1)/n]^{tc_j}] \quad (4)$$

Enfin, nous avons repris un modèle de langue (LM) [HIE 00], [HIE 02] dans lequel les probabilités sont estimées directement en se basant sur les fréquences d'occurrences dans le document D ou dans le corpus C dans son ensemble. Dans cet article, nous avons repris le modèle de Hiemstra [HIE 00] décrit dans l'équation 5 combinant une estimation basée sur le document (soit $Prob[t_j | D_i]$) et sur le corpus ($Prob[t_j | C]$).

$$Prob[D_i | Q] = Prob[D_i] \cdot \prod_{t_j \in Q} [\lambda_j \cdot Prob[t_j | D_i] + (1-\lambda_j) \cdot Prob[t_j | C]] \quad (5)$$
$$\text{avec } Prob[t_j | D_i] = tf_{ij} / nt_i \quad \text{et } Prob[t_j | C] = df_j / lc \quad \text{avec } lc = \sum_k df_k \quad (6)$$

dans laquelle λ_j est un facteur de lissage (une constante pour tous les termes t_j, fixée à 0,35) et *lc* correspond à une estimation de la taille du corpus C.

Afin de comparer les résultats d'une analyse morphologique avec divers enracineurs, nous avons implémenté quatre algorithmes. Le premier correspond à un un enracineur léger visant à éliminer la marque du pluriel dans la langue anglaise (soit le suffixe '-s'). Cette solution comportant trois règles simples a été proposée par Harman [HAR 91] et sera noté "S-stemmer" dans la suite de cet article. D'autres enracineurs cherchent également à éliminer les suffixes dérivationnels (par exemple avec les suffixes '-ship', '-ability' ou '-ment') et nécessitent un nombre plus important de règles comme l'approche proposée par Lovins [LOV 68] (260 règles) et celle suggérée par Porter [POR 80] (60 règles). Le dernier enracineur testé a été repris du système de recherche d'information SMART [SAL 81]. Il correspond essentiellement à une version améliorée de l'algorithme de Lovins.

4. Evaluation

Pour mesurer la performance des différents modèles de recherche d'information, nous avons utilisé la précision moyenne (MAP ou *mean average precision*) obtenue par le système `trec_eval` [BUC 05]. Cette mesure a été adoptée par diverses campagnes d'évaluation pour mesurer la qualité de la réponse calculé par un système de dépistage de l'information. Afin de savoir si une différence entre deux modèles s'avère statistiquement significative, nous avons opté pour un test bilatéral non-paramétrique (basée sur le ré-échantillonnage aléatoire ou *bootstrap* [SAV 97], avec un seuil de signification $\alpha = 5$ %).

4.1 Evaluation des modèles de recherche et des enracineurs

Se basant sur la méthodologie décrite ci-dessus, les résultats obtenus en utilisant quatre enracineurs et cinq modèles de dépistage sont décrits dans la table 1. Dans la deuxième colonne (dénommée "aucun") nous avons indiqué la performance obtenue en l'absence de tout traitement morphologique. Les quatre colonnes subséquentes représentent les quatre enracineurs choisis tandis que la dernière colonne (avec

l'étiquette “lemme”) montre les précisions moyennes que l'on obtient en utilisant une analyse morphologique. Toutes ces performances sont calculées avec des requêtes très courtes correspond aux titres des besoins d'information.

Dans les tables de cette étude, le meilleur résultat dans des conditions données est écrit en gras. En prenant cette valeur comme base pour notre test statistique, nous avons souligné dans chaque colonne les valeurs montrant une différence statistiquement significative. On remarque que, sauf pour l’enracineur léger de Harman (“S-stemmer”), la meilleure performance s'obtient toujours avec le modèle Okapi. Toutefois cette supériorité n'est pas statistiquement significative avec les modèles probabilistes DFR-PL2 et DFR-I(n_e)C2 qui offrent des performances similaires. D'un autre côté, le modèle de langue (LM) et le modèle vectoriel fournissent des différences statistiquement significatives comparées à la précision moyenne du modèle Okapi.

	Précision moyenne (MAP)					
	aucun	S-stemmer ‡	Porter	Lovins	SMART	lemme
Okapi	**0,3743**	0,4044†	**0,4150**†‡	**0,3930**	**0.4152**†‡	0,3988†
DFR-PL2	0,3703	0,4006†	0,4116†‡	0,3927†	0,4096†‡	**0,3994**†
DFR-I(n_e)C2	0,3731	**0,4054**†	0,4141†‡	0,3894 ‡	0,4139†‡	0,3988†
LM	0,3445	0,3709†	0,3809†‡	0,3522 ‡	0,3760†‡	0,3602
tf idf	0,2230	0,2393†	0,2399†	0,2194 ‡	0,2431†‡	0,2308
Moyenne	0,3370	0,3641	0,3723	0,3493	0,3716	0,3576
% différence		+8,0 %	+10,5 %	+3,6 %	+10,2 %	+6,1 %

Table 1. Précision moyenne (MAP) obtenues avec différents modèles et enracineurs (284 requêtes, format T)

Si on prend comme base la précision moyenne obtenue sans aucun traitement lexical (colonne “aucun”), nous avons indiqué avec le symbole “†” les différences statistiquement significatives. Nous pouvons en conclure que dans presque tous les cas, le recours à un enracineur améliore significativement les performances.

Dans un troisième temps, on souhaite connaître s'il existe des différences significatives entre les performances de divers enracineurs ou par rapport à une analyse morphologique. Dans ce but, nous sélectionnons comme valeurs de référence la performance de l'enracineur léger (“S-stemmer”) et toute différence significative sera signalée par le symbole “‡”. Dans ce cas, on observe que l'algorithme proposé par Porter ou celui utilisé par SMART fournissent une meilleure performance. Par contre, l’approche plus agressive suggérée par Lovins donne, significativement, de moins bons résultats en moyenne. Enfin, les différences de

performance entre l'enracineur léger et l'analyse morphologique ne s'avèrent pas statistiquement significatives.

A titre de comparaison, Harman [HAR 91] indique que, basé sur le modèle *tf idf*, aucune différence de performance statistiquement significative n'a pu être détectée entre les enracineurs proposés par Porter [POR 80], Lovins [LOV 68] ou le S-stemmer [HAR 91]. Pour Hull [HUL 96] s'appuyant sur une variante du modèle classique *tf idf*, une amélioration faible (de l'ordre de 1 % à 3 %) peut être attribuable à l'emploi d'enracineurs. Selon cette étude, tous les enracineurs proposent une précision moyenne significativement supérieure à un modèle renonçant à cette forme de normalisation lexicale. De plus, le S-stemmer offrirait une qualité inferieure à celle obtenue par les algorithmes de Porter [POR 80] ou de Lovins [LOV 68]. Ces deux études se limitaient au modèle classique *tf idf* et plus, elles se basaient sur des collections différentes et disposaient d'un nombre de requêtes plus restreint.

4.2 Analyse morphologique, partie du discours et thésaurus

Un des avantages indéniables de la campagne CLEF 2008 consistait à mesurer l'impact de l'analyse morphologique mais également à vérifier l'influence de deux informations pouvant, du moins en partie, désambiguïser la sémantique attachée aux mots. En effet, nous pouvons utiliser les diverses étiquettes concernant la partie du discours (POS) et celles associées au numéro de la classe du thésaurus (synset).

Afin de disposer d'une base de comparaison plus large, nous avons évalué nos approches avec des requêtes courtes (T) dans la table 2 d'une part et, d'autre part, avec des requêtes de taille moyenne (TD), valeurs de performances reportées dans la table 3. Dans les deux cas, la deuxième colonne indique la précision moyenne (MAP) obtenue en utilisant le lemme lors de l'indexation des documents et des requêtes (même valeur que dans la table 1, dernière colonne).

Dans la troisième colonne, le score des documents dépistés sera augmenté si le lemme commun entre eux et la requête possède la même partie du discours (POS). Cette information peut déterminer plus précisément le sens attaché à un terme. En anglais par exemple, le même mot peut avoir des sens différents, comme par exemple "lean" (maigre) comme adjectif ou comme verbe (incliner). Le mot "face" (ou "form", "mean") possède aussi un sens quelque peu distinct lorsque on l'utilise en tant que nom (visage) ou que verbe (assumer, faire face). Dans le but de tenir compte de cette information, nous rajoutons pour chaque terme d'indexation une deuxième unité d'indexation composée du lemme et de son étiquette POS (dérivé du projet *Penn Treebank* [MAR 93]). Par exemple pour l'adjectif "white" on collerait son POS (JJ) et obtiendrait l'unité d'indexation "whiteJJ". Si par contre on rencontre le nom propre "White" ou le nom commun on obtiendrait l'unité d'indexation "whiteNNP" respectivement "whiteNN" où NNP indique qu'il s'agit un nom propre et NN d'un nom commun. Ces deux unités d'indexation diffèrent de "whiteJJ" et permettent alors de distinguer les trois utilisations différentes de la racine "white".

Dans la dernière colonne nous avons reporté la performance obtenue lorsque l'on accroit le score d'un document extrait si le terme commun entre celui-ci et la requête possède le même numéro synset. Dans cette perspective, tous les numéros de classe du thésaurus sont ajoutés à la représentation des documents et des requêtes.

Comme dans la table 1, les valeurs imprimées en gras signalent la meilleure performance pour une colonne donnée, et les performances soulignées indiquent les différences statistiquement significatives par rapport au meilleur score d'une colonne. Les modèles probabilistes DFR ou Okapi proposent une meilleure qualité qui s'avère statistiquement différente de celle obtenue par le modèle de langue (LM) ou l'approche vectorielle.

Si l'on adopte comme référence les performances de la deuxième colonne, les différences statistiquement significatives seront signalées par le symbole "†" placé après la valeur analysée. Dans la table 2 (requêtes T), le modèle Okapi ou LM propose une variation significative de la performance moyenne lorsque l'on tient compte de la partie du discours (POS).

	Précision moyenne (MAP)		
	lemme	lemme & POS	lemme & synset
Okapi	0,3988	**0,4053**†	0.3986
DFR-PL2	**0,3994**	0,4013	0,3918
DFR-I(n_e)C2	0,3988	0,4047	**0,4018**
LM	0,3602	0,3659†	0,3546
tf idf	0,2308	0,2315	0,2325
Moyenne	0,3576	0,3617	0,3559
% différence		+1,2 %	-0,5 %

Table 2. Précision moyenne (MAP) pour différents modèles de RI et variantes d'analyse morphologique (284 requêtes, format T)

Si l'on analyse quelques requêtes, nous pouvons mieux comprendre l'effet de cette information lors de la recherche. Avec le modèle Okapi par exemple, on observe une amélioration de la moyenne de 0,3988 à 0,4053 lors de la prise en compte des parties du discours. Dans ce cas, on améliore la performance pour 133 requêtes, mais on constate une détérioration pour 108 interrogations (il n'y a pas de changement pour le solde des 44 requêtes). L'interrogation n° 217 ("AIDS in Africa") propose la plus grande variation. Dans ce cas, la précision moyenne passe de 0,2037 lorsque l'on ignore les étiquettes POS à 0,5556. Lors du traitement de la requête, le système convertit "AIDS" en "aid" augmentant ainsi le nombre de correspondances possibles ("aid" possédant d'autre sens dans le corpus). Avec la

prise en compte de la partie du discours, “aid” est signalé comme nom propre (étiquette NNP), et ainsi les documents contenant cette abréviation verront leur similarité avec la requête s'accroitre et leur classement s'améliorer.

	Précision moyenne (MAP)		
	lemme	lemme & POS	lemme & synset
Okapi	0,4663	0,4720†	0,4395†
DFR-PL2	0,4608	0,4634	0,4365†
DFR-I(n_e)C2	**0,4671**	**0,4740**†	**0,4665**
LM	0,4444	0,4562†	0,4342†
tf idf	0,2778	0,2879†	0,2834
Moyenne	0,4597	0,4664	0,4442
% différence		+1,5 %	-3,4 %

Table 3. Précision moyenne (MAP) pour différents modèles de RI et variantes d’analyse morphologique (284 requêtes, format TD)

Les évaluations reportées dans la table 3 ont été obtenues avec des requêtes de taille moyenne (TD). Dans ce contexte, le modèle DFR-I(n_e)C2 fournit la meilleure performance (précision moyenne imprimée en gras). Comme dans les évaluations précédentes, les différences avec le modèle de langue (LM) ou l'approche vectorielle *tf idf* sont statistiquement significatives (valeurs soulignées). Si les valeurs moyennes obtenus par l’analyse morphologique (colonne “lemme”) servent de référence, on observe que, exception faite du modèle DFR-PL2, toutes les autres modèles recourant aux étiquettes POS apportent des performances moyennes statistiquement significatives (ajout du symbole “†”), même si ces différences demeures faibles en valeur absolue.

Si on analyse quelques cas, nous constatons que pour le modèle DFR-I(n_e)C2 l'emploi des informations POS permet d'accroitre la MAP de 0,4671 à 0,4740. Derrière cette variation, on observe une amélioration pour 138 interrogations et une dégradation pour 98 requêtes. Comme pour les requêtes courtes, l'interrogation n° 217 (“AIDS in Africa”) voit sa précision moyenne passer de 0,1944 (“lemme”) à 0,5526 (“lemme & POS”).

L'ajout des numéros de classe du thésaurus (“lemme & synset”) apporte des dégradations de performance qui s'avèrent significatives pour les modèles Okapi, DFR-PL2 et LM. Pour expliquer ce phénomène, nous pouvons analyser quelques requêtes. Par exemple pour le modèle Okapi et l'interrogation n° 76 (“Solar Energy”), la précision moyenne passe de 0,663 (“lemme”) à 0,0722 avec la prise en compte des numéros de classe du thésaurus. Pour cette requête, sa partie descriptive contient la forme “is” et deux occurrences de “being”. Le lemme correspondant “be” engendre dix numéros de synsets qui s'ajoutent à la représentation interne. Ainsi chaque document contenant une forme verbale du verbe “to be” aura à chaque

fois dix appariements avec la requête rendant plus difficile la discrimination entre les documents pertinents et ceux qui en le sont pas. *A posteriori* on pourrait imaginer utiliser une liste de mots outils afin d'éliminer les termes ayant une haute fréquence dans la collection avec leur synset pour éviter ce genre de problème. Or on doit aussi noter que ceci aurait probablement un grand effet sur un nombre très restreint de requêtes, mais sur l'ensemble de requêtes l'amélioration serait probablement négligeable.

5. Conclusion

Sur la base d'un corpus rédigé en langue anglaise et enrichi d'information morphologique, nous avons démontré que les meilleures performances s'obtiennent avec le modèle Okapi ou DFR-I(n_e)C2. Cependant, les différences de précision moyenne entre ces deux modèles ou la variante DFR-PL2 ne sont pas significatives. D'un autre côté, le modèle de langue (LM) et le modèle vectoriel *tf idf* produisent des différences statistiquement significatives comparées à la précision moyenne la plus élevée.

Quelque soit le modèle de recherche considéré, l'emploi d'un enracineur ou d'un traitement morphologique permet d'améliorer significativement la performance moyenne. Entre ces diverses approches, nous avons observé une différence significative entre l'algorithme de Porter ou celui du système SMART et un enracineur léger noté S-stemmer. Avec l'approche suggérée par Lovins, la performance moyenne se détériore comparée aux autres enracineurs. L'indexation par lemmes donne des résultats similaires aux algorithmes de Porter ou du système SMART. On constate en particulier que l'approche très simple de l'enracineur léger basé sur trois règles ayant comme but d'éliminer les marques du pluriel (S-stemmer), s'avère au moins aussi efficace que des approches plus agressives comme celles proposés par Lovins (260 règles) et par Porter (60 règles).

Pour favoriser la transparence de ce processus envers l'utilisateur, il serait avantageux d'utiliser une approche simple, produisant des performances similaires à des approches plus agressives. Ces dernières tendent à réduire plus fortement les formes de surface, rendant parfois moins compréhensible l'appariement entre formes de surface et racine pour l'utilisateur. Dans cette optique, le moteur de recherche Google applique un enracineur léger. Ainsi si l'on soumet la requête "computes" le système retourne des documents contenant aussi les formes "compute", "computers" ou "computing" mais pas la forme "computable".

L'adjonction de la partie du discours permet, dans le cadre de requête moyenne (TD), d'augmenter significativement la performance moyenne. Avec des interrogations de faible longueur (T), cette amélioration s'avère significative pour le modèle de langue et l'approche Okapi. L'inclusion des numéros de classe d'un thésaurus dans les documents et les requêtes a tendance à diminuer la précision

moyenne. En présence de requêtes de taille moyenne (TD), la différence de performance s'avère souvent significative. Notre solution doit toutefois être vue comme un premier essai qui mériterait quelques améliorations comme la sélection d'une seule classe par mot au lieu de laisser l'ensemble des possibilités. Comme autre possibilité, nous pourrions tenir compte du thésaurus uniquement pour certaines parties du discours comme les noms par exemple.

En plus il faut remarquer que les résultats obtenus sont liés à la langue anglaise, et pourrait s'appliquer à d'autres langues possédant une morphologie flexionnelle simple. Tomlinson [TOM 04] par exemple compare les enracineurs lexicaux et algorithmiques pour neuf langues européennes (allemand, français, italien, espagnol, néerlandais, finnois, suédois, russe et anglais). Pour le finnois et l'allemand, l'analyse morphologique (enracineur lexical) apporte des performances significativement supérieures tandis que pour les autres langues, les différences de performances ne sont pas significatives. La présence d'une morphologie flexionnelle complexe semble être à l'origine de ces différences.

Remerciements

Cette recherche a été financée en partie par le Fonds national suisse pour la recherche scientifique (subside n^0 200021-113273).

6. Bibliographie

[AMA 02] Amati, G., & van Rijsbergen, C.J. "Probabilistic models of information retrieval based on measuring the divergence from randomness", *ACM-Transactions on Information Systems*, vol. 20, n^o 4, 2002, p. 357-389.

[BUC 05] Buckley, C., & Voorhees, E.M. "Retrieval system evaluation", E.M. Voorhees, D.K. Harman (Eds): *TREC. Experiment and Evaluation in Information Retrieval* (pp. 53-75). The MIT Press, Cambridge (MA), 2005.

[CHA 07] Chan, Y.S., Ng, H.T., & Zhong, Z. "NUS-PT: Exploiting Parallel Texts for Word Sense Disambiguation in the English All-Words Tasks", *Proceedings of the 4th International Workshop on Semantic Evaluations* (SemEval 2007), Prague, p. 253–256.

[FEL 98] Fellbaum, C. "*WordNet. An Electronic Lexical Database*", The MIT Press, Cambridge (MA), 1998.

[FOX 90] Fox, C. "A stop list for general text", *ACM-SIGIR Forum*, vol. 24, n^o 1, 1990, p. 19-35.

[HAR 91] Harman, D. "How effective is suffixing?", *Journal of the American Society for Information Science*, vol. 42, n^o 1, 1991, p. 7-15.

[HIE 00] Hiemstra, D. "Using language models for information retrieval. CTIT Ph.D. Thesis, 2000.

[HIE 02] Hiemstra, D. "Term-specific smoothing for the language modeling approach to information retrieval. The importance of a query term", *Proceedings of the ACM-SIGIR'2002*, Tempere, p. 35-41.

[HUL 96] Hull, D. “Stemming algorithms: A case study for detailed evaluation”, *Journal of the American Society for Information Science*, vol. 47, n° 1, 1996, p. 70-84.

[LOV 68] Lovins, J.B. “Development of a stemming algorithm”, Mechanical Translation and Computational Linguistics, vol. 11, n° 1, 1968, p. 22-31

[MAN 08] Manning, C.D, Raghavan, P., & Schütze, H. “*Introduction to Information Retrieval*”, Cambridge University Press, Cambridge (UK), 2008.

[PER 08] Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A. & Santos, D. (Eds.). “*Advances in Multilingual and Multimodal Information Retrieval*”, LNCS #5152, Springer-Verlag, Berlin, 2008.

[POR 80] Porter, M.F. “An algorithm for suffix stripping”, *Program*, vol. 14, n° 3, 1980, p. 130-137.

[RAT 96] Ratnaparkhi, A. “A maximum entropy part-of-speech tagger”, *Proceedings of the Empirical Methods in Natural Language Processing Conference*, 1996, p. 133-142.

[ROB 00] Robertson, S.E., Walker, S., & Beaulieu, M. “Experimentation as a way of life: Okapi at TREC”, *Information Processing &Management*, vol. 36, n° 1, 2000, p. 95-108.

[SAL 71] Salton, G. “*The SMART Retrieval System - Experiments in Automatic Document Processing*”, Prentice-Hall Inc., Englewood Cliffs (NJ), 1971.

[SAV 93] Savoy, J. “Stemming of French words based on grammatical category”, *Journal of the American Society for Information Science*, vol. 44, n° 1, 1993, p. 1-9.

[SAV 97] Savoy, J. “Statistical inference in retrieval effectiveness evaluation”, *Information Processing & Management*, vol. 33, n° 4, 1997, p. 495-512.

[TOM 04] Tomlinson, S. “Lexical and algorithmic stemming compared for 9 European languages with Hummingbird SearchServer™ at CLEF 2003”, *Comparative Evaluation of Multilingual Information Access Systems* LNCS #3237, Springer-Verlag, Berlin, 2004. p. 286-300.

Reconnaissance du type de discours dans des corpus comparables spécialisés

Lorraine Goeuriot, Emmanuel Morin et Béatrice Daille

Université de Nantes, LINA - UMR CNRS 6241
{lorraine.goeuriot,emmanuel.morin,beatrice.daille}@univ-nantes.fr

RÉSUMÉ. *Notre objectif est d'automatiser la construction de corpus comparables spécialisés à partir du Web. La comparabilité se base sur trois niveaux : le domaine, le thème et le type de discours. Le domaine et le thème peuvent être filtrés grâce aux mots-clés utilisés lors de la recherche. Nous présentons dans cet article la reconnaissance automatique du type de discours dans des documents spécialisés français et japonais, qui nécessite une analyse linguistique poussée. Une analyse contrastive des documents nous permet de déterminer quelles informations paraissent discriminantes. En s'inspirant des travaux classiques de recherche d'information, nous créons une typologie robuste et linguistiquement motivée basée sur trois niveaux d'analyse : structurel, modal et lexical. Cette typologie nous permet d'apprendre des modèles de classification qui donnent de bons résultats, ce qui montre l'efficacité de cette typologie.*

ABSTRACT. *Our goal is to automate the compilation of smart specialized comparable corpora. The comparability is based on three levels: domain, topic and type of discourse. Domain and topic can be filtered with the keywords used through web search. We present in this paper the automatic detection of the type of discourse in French and Japanese documents, which needs a wide linguistic analysis. A contrastive analysis of the documents leads us to specify which information is relevant to distinguish them. Referring to classical studies on information retrieval, we create a robust and linguistically motivated typology based on three analysis levels: structural, modal and lexical. This typology is used to learn classification models using shallow parsing. We obtain good results, that demonstrates the efficiency of this typology.*

MOTS-CLÉS : *classification automatique, type de discours, typologie multilingue, corpus comparables*

KEYWORDS: *automatic classification, type of discourse, multilingual typology, comparable corpora*

1. Introduction

L'exploitation de corpus comparables est un domaine de recherche récent, qui vise à suppléer les inconvénients liés à l'utilisation de corpus parallèles notamment lorsqu'il s'agit de travailler avec un couple de langues ne faisant par intervenir l'anglais. Les corpus comparables sont principalement utilisés pour extraire des terminologies multilingues (Déjean *et al.*, 2002, Morin *et al.*, 2007) ou des lexiques multilingues (Fung *et al.*, 1998, Rapp, 1999). Ils représentent aussi une ressource précieuse dans le cadre d'études contrastives multilingues (Peters *et al.*, 1997) et permettent aux traducteurs (Laviosa, 1998) et enseignants d'observer la langue dans son usage.

La profusion de documents accessibles dans des langues variées sur le web incite à puiser dans ce réservoir pour constituer des corpus comparables. Néanmoins, cette tâche ne saurait se réduire à la simple collecte de documents partageant un vocabulaire commun. Il est nécessaire de respecter des caractéristiques communes telles que le thème et le domaine (Bowker *et al.*, 2002) qui sont fixées avant la construction du corpus et qui sont fonction de sa finalité (McEnery *et al.*, 2007). De nombreux travaux traitent de la construction de corpus à partir du Web (Baroni *et al.*, 2006, Chakrabarti *et al.*, 1999) mais aucun, à notre connaissance, n'est consacré à celle des corpus comparables, qui doit répondre à différentes contraintes. Nous fixons ainsi la comparabilité à trois niveaux : le domaine, le thème et le type de discours.

L'objectif que nous poursuivons dans cette étude vise la constitution automatique de corpus comparables spécialisés à partir de documents issus du web pour des couples de langues à grande distance linguistique. Plus précisément, nous cherchons à rendre opérationnelle la précédente notion de comparabilité. Le domaine et thème d'un document pouvant être filtrés grâce aux mots-clés lors de la recherche (Chakrabarti *et al.*, 1999), nous nous concentrons ici sur la reconnaissance automatique des types de discours des domaines de spécialité : scientifique et vulgarisé. Pour ce faire, nous mettons en évidence un ensemble de critères, linguistiquement motivés, discriminants et opératoires pour caractériser les types de discours scientifique et vulgarisé. Ces critères implémentés au sein d'un système de classification automatique permettent de créer un corpus comparable français/japonais spécialisé dont la qualité avoisine celle obtenue manuellement.

La suite de cette étude est structurée comme suit. Après une introduction des travaux relatifs à l'exploitation de corpus comparables dans la section 2, l'analyse stylistique effectuée sur notre corpus d'étude est présentée dans la section 3. Elle nous permet de créer une typologie des types de discours scientifique et vulgarisé dans des domaines de spécialité. L'application d'algorithmes d'apprentissage à celle-ci est décrite dans la section 4 et ses résultats dans la section 5.

2. Contexte

« A comparable corpus can be defined as a corpus containing components that are collected using the same sampling frame and similar balance and representative-

ness » (McEnery *et al.*, 2007, p. 20). La comparabilité est garantie grâce à des caractéristiques pouvant référer au contexte de création des documents (période, auteur...) ou aux documents eux-mêmes (thème, genre...). Le choix des caractéristiques communes, qui définissent le contenu du corpus, influent sur le *degré de comparabilité*, notion permettant de quantifier dans quelle mesure deux corpus sont comparables. Ce choix dépend des objectifs applicatifs du corpus, nous distinguons dans les travaux sur ces corpus deux types :

Les corpus comparables généralistes : composés généralement d'articles de journaux. Les documents sont souvent extraits de journaux nationaux, et portent sur une même période, voire une même thématique. Fung *et al.* (1997), par exemple, utilisent un corpus anglais/japonais composé d'articles extraits du Wall Street Journal et du Nikkei Financial News (journaux traitant du domaine financier) sur une même période. Rapp (1999) utilise lui aussi des articles extraits de grands journaux nationaux allemand et anglais sur une même période, mais sans cibler de domaine particulier.

Les corpus comparables spécialisés : composés de documents émanant d'un domaine spécialisé, souvent scientifique, faisant appel à un langage spécialisé. Déjean *et al.* (2002) utilisent par exemple un corpus composé de documents médicaux tirés de la base de données médicales MEDLINE, ainsi que Chiao (2004) , utilisant les bases CISMEF, CLINIWEB et OSHUMED.

Dans les travaux sur la langue générale, les documents partagent souvent des caractéristiques telles que le domaine et le thème. Étant souvent extraits de journaux périodiques, il est important de les limiter à une certaine période afin de garantir la comparabilité. Dans les travaux sur les langues de spécialité, un premier niveau de comparabilité peut être assuré grâce au domaine ou au thème. De plus, différentes situations de communication peuvent être observées dans les langues de spécialité (Pearson, 1998, p. 36) : la communication d'expert à expert, d'expert à initié, de semi-expert à non-initié, d'enseignant à élève... Malrieu *et al.* (2002) relèvent différents niveaux de classification de textes, chaque niveau correspondant à une certaine granularité. Le premier niveau est le *discours*, défini comme un ensemble d'énoncés d'un énonciateur caractérisé par une unité globale de thème (Ducrot *et al.*, 1999). Le second niveau est le *genre*, défini comme les catégories de textes distinguées spontanément par les locuteurs d'une langue. Par exemple, au discours littéraire correspondent les genres : théâtre, poésie, récit... En s'inspirant de ces situations de communication et ces niveaux de classification de textes, nous avons choisi de distinguer deux situations de communication, que nous appelons *types de discours*, dans les domaines de spécialité : scientifique (textes écrits par des spécialistes à destination de spécialistes) et vulgarisé (textes écrits pour des non-spécialistes par des non-spécialistes, semi-spécialistes ou spécialistes). Ce niveau de comparabilité, le type de discours, reflète le « contexte de production ou d'usage » des documents (Habert *et al.*, 2001) et garantit une homogénéité lexicale dans le corpus (Bowker *et al.*, 2002, p. 27). De plus, Morin *et al.* (2007) montrent qu'un corpus comparable dont les documents partagent un thème et un type de discours est très adapté à l'extraction de terminologies multilingues.

Dans cette étude, nous nous intéressons à la catégorisation automatique de documents selon leur type de discours. Elle est basée sur une typologie composée de critères caractérisant ce type de discours, élaborée grâce à une analyse stylistique contrastive (Karlgren, 1998). Son but est de trouver des critères linguistiquement motivés, correspondant à différents niveaux d'analyse, dont la combinaison caractérise un type de discours.

3. Analyse des types de discours

La première étape de cette analyse des types de discours est une analyse stylistique manuelle, basée sur les méthodes déductives et contrastives, dont le but est de mettre en évidence des critères discriminants et linguistiquement motivés caractérisant les types de discours scientifique et vulgarisé. La principale difficulté de cette tâche réside dans la recherche de critères pertinents adaptés à chaque langue. Ces critères sont ensuite rassemblés dans une typologie qui sera utilisée afin d'apprendre des modèles de classification. Celle-ci se devra d'être robuste, générique et extensible à d'autres langues. La généricité sera garantie par une typologie couvrant une grande variété de caractéristiques textuelles, la robustesse par des critères opératoires et un traitement aisément adaptable aux textes comme aux documents du Web.

Sinclair (1996) distingue deux niveaux d'analyse dans son rapport sur les typologies textuelles : un niveau externe, concernant le contexte de création des textes et un niveau interne, correspondant aux caractéristiques linguistiques des textes. Nos corpus étant composés de documents extraits du Web, nous considérons les critères du niveau externe comme tous les critères liés à la création des documents et à leur structure (critères non-linguistiques), nous les appelons *critères structurels*. Ke *et al.* (2009) utilisent les fréquences des mots et caractères ainsi que certaines informations lexicales et structurelles afin de distinguer les types de discours en chinois. Notre analyse stylistique a permis de mettre en évidence différents niveaux de granularité dans les critères linguistiques, le niveau d'analyse interne est donc composé de deux catégories. Pour distinguer les deux niveaux de communication que sont nos types de discours, nous devons tout d'abord considérer le locuteur dans son discours : les critères modaux (Nakao, 2008). De plus, le discours scientifique peut être caractérisé par son vocabulaire, la longueur des mots et autres critères lexicaux. Notre typologie est donc composée de trois niveaux d'analyse : structurel, modal et lexical.

3.1. *Les critères structurels*

Nos documents étant extraits du Web, nous devons considérer leur structure et le contexte de leur création. Dans le cadre de la classification de documents du Web, différents éléments apportent des informations pertinentes : les images, les vidéos et d'autres contenus multimédia (Asirvatham *et al.*, 2001) ; les méta-informations, le titre et la structure `HTML` (Riboni, 2002). Les critères structurels de notre typologie sont : le patron d'URL, le format du document, les balises `META`, la balise `TITLE`, la mise en

page (utilisation de CSS, cadres, tableaux…), le fond des pages, les images, les liens, les paragraphes, les listes, le nombre de phrases, la typographie (italique, gras…) et la longueur des documents (nombre de caractères).

3.2. *Les critères modaux*

Le degré de spécialisation requis par le lecteur ou l'interlocuteur est caractérisé par la relation établie dans l'énoncé entre le locuteur ou l'auteur et l'interlocuteur ou le lecteur[1]. Cette relation est caractérisée par le ton du locuteur et par l'emploi de certains traits linguistiques. La modalisation est une interprétation de l'attitude du locuteur vis-à-vis du contenu de son discours (Querler, 1996), elle est caractérisée par différents marqueurs textuels : les verbes, les adverbes, les formules de politesse… La plupart des théories de la modalisation sont dépendantes de la langue et font appel à une description des phénomènes spécifiques à chaque langue. Pourtant, la théorie de Charaudeau (1992) semble indépendante de la langue et opérationelle pour les langues française et japonaise. Selon lui (Charaudeau, 1992, p. 572), la modalisation permet d'expliciter au sein d'un énoncé la position du locuteur par rapport à l'interlocuteur, à lui-même et à son discours. Elle est composée d'actes locutifs, qui sont des positions particulières du locuteur dans son discours. Chacun d'entre eux est caractérisé par différentes modalités. Nous en observons deux dans cette théorie :

L'acte allocutif : le locuteur implique l'interlocuteur dans son discours (ex. : « *Tu dois le faire* ») ;

L'acte élocutif : le locuteur est impliqué dans son discours, il révèle sa propre position (ex. : « *J'aimerais le faire* »).

Les modalités sont présentées dans le tableau 1 avec des exemples en français. Certaines d'entre elles ne sont pas utilisées dans un langage ou l'autre si elles sont trop peu fréquentes ou trop ambiguës.

3.3. *Les critères lexicaux*

Biber (1988, 1989) utilise des informations lexicales afin d'observer les variations entre des textes et plus particulièrement entre leurs genres et leurs types. Karlgren *et al.* font appel aux critères lexicaux afin de caractériser les genres textuels et observer les variations stylistiques entre textes. Nous considérons ainsi que les informations lexicales peuvent être pertinentes dans la distinction des types de discours scientifique et vulgarisé. La première raison est que le vocabulaire spécialisé est l'une des principales caractéristiques des textes issus de domaines de spécialité (Bowker *et al.*, 2002, p. 26). De plus, les documents scientifiques contiennent davantage d'unités lexicales complexes, groupes nominaux ou phrases nominales que les documents vulgarisés

1. Comme nous travaillons sur des domaines de spécialité, nous considérons le locuteur comme auteur des textes et l'interlocuteur comme lecteur.

Modalité	Exemple	Français	Japonais
Acte allocutifs			
Pronoms personnels allocutifs	*Tu, vous*	×	
Injonction	*Ne fais pas ça*	×	×
Autorisation	*Tu peux le faire*	×	
Jugement	*Bravo, tu as réussi !*	×	
Suggestion	*Tu devrais le faire*	×	×
Interrogation	*Quand arrives-tu ?*	×	×
Interpellation	*Comment allez-vous, monsieur ?*	×	
Requête	*S'il vous plaît, faites-le*	×	×
Actes élocutifs			
Pronoms personnels élocutifs	*Je, nous, on*	×	×
Constat	*Je remarque qu'il est parti*	×	×
Connaissance	*Nous savons qu'il est parti*	×	×
Opinion	*Je pense qu'il est parti*	×	×
Volonté	*Je voudrais qu'il parte*	×	×
Promesse	*Je te promets qu'il sera là*	×	×
Déclaration	*Je t'assure qu'il est parti*		×
Appréciation	*Je l'aime bien*	×	
Obligation	*Nous devons le faire*	×	
Possibilité	*Je peux le leur dire*	×	

Tableau 1. *Critères modaux*

(Sager, 1990). Nous présentons dans le tableau 2 les critères lexicaux. Il est à noter que ceux-ci sont plus dépendants de la langue que les critères présentés précédemment.

Critère	Français	Japonais
Vocabulaire spécialisé	×	×
Caractères numériques	×	×
Unités de mesure	×	×
Longueur des mots	×	
Bibliographie	×	×
Citations bibliographiques	×	×
Ponctuation	×	×
Fins de phrase		×
Parenthèses	×	×
Autres alphabets (latin, hiragana, katakana)		×
Symboles		×

Tableau 2. *Critères lexicaux*

4. Reconnaissance automatique du type de discours

L'élaboration d'un système de classification automatique est réalisée en trois étapes : l'indexation des documents, l'apprentissage du classifieur et son évaluation (Sebastiani, 2005, p. 112, 113). L'indexation des documents consiste à générer une représentation compacte des documents pouvant être interprétée par un classifieur. Dans notre cas, chaque document d_i est représenté par un vecteur de poids des critères :

$$\vec{d_i} = \{w_{1i}, \ldots, w_{ni}\}$$

où n représente le nombre de critères de la typologie et w_{ji} représente le poids du j^{eme} critère dans le i^{eme} document. Chaque poids de critère est normalisé, en le divisant par le total. L'indexation des documents est ici effectuée grâce à la typologie (cf. section 3) et à l'implémentation de ces critères.

L'implémentation des critères de notre typologie se fait au moyen de patrons lexico-syntaxiques (*i.e.* des expressions régulières).

4.1. *Critères structurels*

La majorité des critères structurels (présentée en section 3.1) est implémentée par des opérations de recherche de motifs. Par exemple, le patron d'URL permet de déterminer si un document est issu d'un site hospitalier (`http://www.chu-***.fr`) ou d'un site universitaire (`http://www.univ-***.fr`)... Quant aux images, paragraphes, liens, etc., une simple recherche de balises a été effectuée.

4.2. *Critères modaux*

Les marqueurs de présence du locuteur dans un texte peuvent être implicites ou ambigus. Nous avons préféré utiliser des marqueurs simples afin d'éviter d'introduire trop de bruit dans notre système (précision forte). Nous y introduisons toutefois du silence (rappel fort) : toutes les occurrences d'une modalité ne sont pas détectées mais celles qui le sont sont correctes. Certains pronoms sont spécifiques à l'acte locutif : par exemple, les pronoms français *je* et *nous*, et les japonais 私 (je), 私達 (nous) et 我々 (nous) sont caractéristiques de l'acte élocutif. Nous utilisons de plus des marqueurs lexicaux : par exemple, la modalité du savoir peut être détectée en français grâce aux verbes *savoir, connaître* et en japonais avec le verbe 知る (savoir), dans la forme polie 知っています et dans la forme neutre 知っている.

4.3. *Critères lexicaux*

Nous présentons ici l'implémentation des onze critères lexicaux introduits dans le tableau 2. Certains d'entre eux sont spécifiques aux documents scientifiques, comme

les bibliographies, citations bibliographiques ou vocabulaire spécialisé. Pour mesurer la densité terminologique en français (proportion de vocabulaire spécialisé dans un texte), nous recherchons des affixes gréco-latins (Namer *et al.*, 2007) et des adjectifs relationnels particulièrement fréquents dans les domaines scientifiques (Daille, 2000). Nous avons dénombré près de 50 affixes tels que *inter-*, *auto-* ou *nano-* et 10 suffixes relationnels tels que *-ique* ou *-al*. Ces affixes peuvent être présents dans les deux types de discours, mais dans des proportions différentes. Par exemple, le terme *ovariectomie* peut être fréquent dans un document scientifique tandis qu'il sera très rarement employé dans un document vulgarisé, et ce au profit du terme *ablation des ovaires*. Les fins de phrases sont des particules de terminaison spécifiques, par exemple la particule か qui est souvent utilisée à la fin des phrases interrogatives.

4.4. *Algorithmes de classification automatique*

La classification automatique est un processus qui, partant d'un ensemble de vecteurs dans une classe c ou $\overline{c}$, détermine quelles caractéristiques doit avoir un nouveau document pour être classé dans l'une de ces classes[2]. À partir d'une indexation de documents, il existe plusieurs algorithmes permettant de réaliser ce processus (les réseaux de neurones, les classifieurs bayesiens, les machines à vecteurs de support…) dont (Sebastiani, 2002) a mené une comparaison. Appliquées à des corpus de dépêches Reuters, ces méthodes donnent des résultats variables selon le nombre de classes, de critères… Dans cette étude, les systèmes *SVMlight* (Joachims, 2002) et *C4.5* (Quinlan, 1993) donnent de très bons résultats dans un contexte similaire au notre : petits corpus, classification binaire, moins de 100 critères.

5. Expérimentations

Nous décrivons dans cette section les deux corpus comparables que nous avons utilisés et présentons les expériences menées sur ceux-ci. Le premier corpus est utilisé afin d'apprendre un modèle de classification basé sur notre typologie (la phase d'apprentissage), tandis que le second corpus sert à évaluer ce modèle de classification sur de nouveaux documents (la phase d'évaluation).

5.1. *Corpus comparables*

Les corpus utilisés dans nos expériences sont composés de documents français et japonais extraits du Web. Ils sont issus du domaine médical, sur les thématiques *diabète et alimentation* pour la phase d'apprentissage et *cancer du sein* pour la phase d'évaluation. La collecte des documents a été menée manuellement. Leur domaine et leur thématique ont été filtrés grâce aux mots-clés : par exemple, *alimentation, diabète*

2. Dans le cas binaire ; voir (Sebastiani, 2005) pour les autres cas.

et *obésité* pour la partie française et 糖尿病 (*diabète*) et 肥満 (*surpoids*) pour la partie japonaise du corpus d'apprentissage. Les documents ont ensuite été manuellement sélectionnés puis classés par des locuteurs natifs de chaque langue, qui ne sont pas des spécialistes du domaine médical, selon leur type de discours : scientifique (SC) ou vulgarisé (VU). La classification manuelle se base sur les heuristiques suivantes :

– Un document scientifique est écrit par des spécialistes, à destination de spécialistes.

– En ce qui concerne les documents vulgarisés, nous distinguons deux niveaux de vulgarisation : les documents écrits par des spécialistes pour le grand public et les documents écrits par le grand public pour le grand public. Nous ne distinguons pas ici ces deux niveaux mais accordons toutefois plus d'importance aux documents écrits par des spécialistes, potentiellement plus riches en contenu et en vocabulaire (les conseils d'un médecin à ses patients peuvent être plus riches qu'une discussion de forum).

Notre classification manuelle des documents se base donc sur ces deux heuristiques, ainsi que sur différents éléments empiriques : l'origine du site Web, le vocabulaire employé... Pour quelques documents, il a été difficile de déterminer le type de discours (par exemple des documents écrits par des personnes dont le degré de spécialisation n'était pas clair). Ils n'ont pas été conservés dans le corpus.

Nous avons donc créé deux corpus comparables :

– [DIABÈTE] portant sur le thème *diabète et alimentation* et utilisé lors de la phase d'apprentissage.

– [CANCER] portant sur le thème *cancer du sein* et utilisé lors de la phase d'évaluation.

Le tableau 3 présente les principales caractéristiques de chaque corpus : le nombre de documents et le nombre de mots[3] pour chaque langue et chaque type de discours.

			# doc.	# mots
[DIABÈTE]	FR	SC	65	425 781
		VU	183	267 885
	JP	SC	119	234 857
		VU	419	572 430
[CANCER]	FR	SC	50	443 741
		VU	42	71 980
	JP	SC	48	211 122
		VU	51	123 277

Tableau 3. *Principales caractéristiques des deux corpus comparables*

3. Pour le japonais, le nombre de mots est le nombre d'occurrences reconnues par ChaSen (Matsumoto *et al.*, 1999)

5.2. *Résultats de la phase d'apprentissage*

Dans cette première expérience, nous entraînons et testons nos classifieurs sur le corpus [DIABÈTE]. Nous utilisons la méthode par validation croisée (*N-fold cross validation method*) qui consiste à diviser le corpus en n partitions de même taille. Si nous fixons $n = 5$, à chaque itération, le sous-corpus d'apprentissage compte 80 % des documents du corpus initial (en terme de caractères) et les 20 % restants (correspondant à la i^{eme} partition) sont utilisés pour l'évaluation. Les résultats que nous donnons sont des moyennes sur ces 5 partitions et nous utilisons les métriques de précision et de rappel pour évaluer l'efficacité des classifieurs :

$$\text{Précision} = \frac{\text{\# doc. correctement classés dans } c}{\text{\# doc. classés dans c}}$$

$$\text{Rappel} = \frac{\text{\# doc. correctement classés dans } c}{\text{\# doc. appartenant à c}}$$

Les résultats obtenus avec les systèmes *SVMlight* et *C4.5* sur le corpus [DIABÈTE] sont présentés dans le tableau 4. Notre mesure de référence est la suivante : nous considérons pour chaque classe (scientifique ou vulgarisée) que 50 % des documents lui appartenant sont correctement classés. Ainsi, le rappel est toujours de 50 % tandis que la précision varie : elle est faible pour les documents scientifiques et satisfaisante pour les documents vulgarisés. Nous pouvons constater que quels que soient la langue et le système de classification, notre méthode donne des résultats corrects sur les documents vulgarisés. Nous améliorons le rappel et la précision de la méthode de référence dans quasiment tous les cas de figure. En ce qui concerne les documents scientifiques, nos résultats sont bien meilleurs que ceux de référence pour le français comme pour le japonais avec le système *C4.5*. En revanche, les résultats obtenus avec le système *SVMlight* sont plus diffus, notamment en ce qui concerne le rappel des documents français.

Si nous ne tenons pas compte de la distinction des documents selon le type du discours, les résultats obtenus en français sont globalement satisfaisants avec un rappel moyen de 87 % et une précision moyenne de 90 % avec le système *C4.5* (plus de 215 documents sur 248 sont correctement classés). Les résultats de la classification des documents japonais sont bons avec le classifieur *C4.5* : plus de 90 % des documents sont correctement classifiés et la précision atteint en moyenne 80 %. Les résultats les plus faibles obtenus sur les documents japonais peuvent s'expliquer par la grande variété de genres dans ce corpus (articles de recherche, de journaux, recettes de cuisine, offres d'emploi, discussions de forums...).

Nous présentons dans le tableau 5 les résultats de la classification obtenus pour chaque catégorie de critères considérée indépendamment, avec les deux systèmes de classification sur le corpus [DIABÈTE]. Dans chaque cas, les représentations vectorielles des documents ne contiennent que les poids des critères de la catégorie concernée. Quel que soit le classifieur, nous n'observons pas de grande baisse des résultats

		Français		Japonais	
		Préc.	Rapp.	Préc.	Rapp.
Mesure de référence	SC	0,26	0,50	0,22	0,50
	VU	0,74	0,50	0,78	0,50
SVMlight	SC	1,00	0,36	0,70	0,65
	VU	0,80	1,00	0,72	0,80
C4.5	SC	0,89	0,80	0,76	0,96
	VU	0,91	0,94	0,95	0,99

Tableau 4. *Précision et rappel pour chaque langue et classifieur pour le corpus* [*DIABÈTE*]

en ne conservant qu'une seule catégorie de critères. Par contre, les résultats sur les documents japonais sont inférieurs. Nous pouvons en déduire que la combinaison de chacune de ces catégories de la typologie permet d'améliorer les résultats de la classification. Cependant, aucune catégorie ne se distingue clairement dans cette expérience, les plus efficaces ne sont pas les mêmes selon le système utilisé et la langue. Avec *SVMlight*, les critères lexicaux et structuraux semblent les plus discriminants. Avec *C4.5*, les critères modaux donnent de meilleurs résultats sur les documents français, tandis que les critères lexicaux améliorent les résultats pour le japonais. Chaque catégorie semble discriminante pour une langue ou un système de classification et les expériences sur la typologie complète montrent que leur combinaison améliore les résultats.

		Français		Japonais	
		Préc.	Rapp.	Préc.	Rapp.
SVMlight	Structurel	0.90	0.67	0.59	0.71
	Modal	0.60	0.50	0.50	0.49
	Lexical	0.91	0.75	0.58	0.53
C4.5	Structurel	0.85	0.85	0.41	0.44
	Modal	0.89	0.91	0.39	0.44
	Lexical	0.85	0.85	0.47	0.45

Tableau 5. *Résultats de chaque catégorie de critères sur le corpus* [*DIABÈTE*]

5.3. *Résultats de la phase d'évaluation*

Afin d'évaluer l'impact de l'application des modèles de classification générés sur de nouveaux documents, une nouvelle expérience a été menée : les classifieurs ap-

pris sur le corpus [DIABÈTE] sont testés sur le corpus [CANCER]. Les résultats sont présentés dans le tableau 6.

Nous notons une baisse globale des résultats de la classification sur ce corpus d'évaluation bien qu'ils restent satisfaisants. Les documents français sont classés avec une précision supérieure à 75% et un rappel de plus de 75%, ce qui représente plus de 70 documents correctement classés sur 92. La classification des documents japonais donne de bons résultats, avec une précision de 76% et un rappel de 77% en moyenne, ce qui représente 23 documents mal classés sur 99. Ces modèles de classification semblent donc efficaces pour distinguer les types de discours scientifique et vulgarisé dans des documents spécialisés français et japonais.

Selon les objectifs applicatifs du corpus, il peut être souhaitable de privilégier la précision ou le rappel. Par exemple, (Morin *et al.*, 2007) montrent qu'un corpus composé de documents scientifiques est plus adapté à l'extraction de termes complexes bilingues dans des domaines de spécialité qu'un corpus mêlant les deux types de discours. Dans ce cas, la précision doit être privilégiée au rappel et *SVMlight* le permet.

		Français		Japonais	
		Préc.	Rapp.	Préc.	Rapp.
SVMlight	SC	0,92	0,53	0,90	0,61
	VU	0,64	0,95	0,66	0,98
C4.5	SC	0,70	0.92	0,76	0,70
	VU	0,87	0,56	0,75	0,80

Tableau 6. *Précision et rappel pour chaque langue et classifieur pour le corpus* [*CANCER*]

6. Conclusion

Dans cet article nous avons décrit une première étape de la construction automatique de corpus comparables spécialisés en français et en japonais. Une qualité proche des corpus construits manuellement est garantie par le choix des caractéristiques communes aux textes : un domaine, un thème et un type de discours. Une analyse stylistique contrastive nous a permis de créer une typologie composée de critères caractérisant les types de discours scientifique et vulgarisé dans des documents spécialisés issus du Web. Cette typologie est basée sur trois niveaux d'analyse des documents : le niveau structurel, le niveau modal et le niveau lexical. Ses critères ont été mis en œuvre et des modèles de classification ont été générés avec les systèmes *SVMlight* et *C4.5*. Ces derniers donnent de bons résultats, sur le corpus d'apprentissage ainsi que sur le corpus d'évaluation avec une précision moyenne de 80 % et un rappel moyen de 70 %.

Toutefois, l'aspect binaire de notre classification nous parait discutable. Il peut être en effet intéressant de considérer les classes scientifique et vulgarisée comme un continuum, ce qui nous mènerait à évaluer pour chaque document un degré de spécialisation plutôt qu'une appartenance à une classe. De plus, *SVMlight* attribue à chaque document un score, que nous interprétons comme l'appartenance à l'une des deux classes. Nous envisageons de considérer ces scores du point de vue du continuum. Nous pourrions ainsi distinguer un plus grand nombre de situations de communication : de spécialiste à spécialiste, de spécialiste à non-spécialiste, de non-spécialiste à non-spécialiste… (Pearson, 1998). Nous avons souhaité que notre typologie soit générique, de façon à pouvoir être adaptée à d'autres langues. Les critères étant déjà définis, il sera nécessaire pour ajouter une langue de trouver les marqueurs pour chacun des critères et de créer un corpus sur lequel un modèle sera appris pour cette nouvelle langue.

Remerciements

Ce travail a été mené dans le cadre du projet ANR C-Mantic 2007-2009. Nous remercions Yukie Nakao pour son travail sur la typologie et les marqueurs japonais.

7. Bibliographie

Asirvatham A. P., Ravi K. K., « Web Page Classification Based on Document Structure », *IEEE National Convention*, 2001.

Baroni M., Kilgarriff A., « Large Linguistically-Processed Web Corpora for Multiple Languages », *EACL'06*, The Association for Computer Linguistics, p. 87-90, 2006.

Biber D., *Variation across Speech and Writing*, Cambridge University Press, 1988.

Biber D., « A typology of English texts », *Linguistics*, vol. 27, p. 3-43, 1989.

Bowker L., Pearson J., *Working with Specialized Language : A Practical Guide to Using Corpora*, London/New York, Routeledge, 2002.

Chakrabarti S., van den Berg M., Dom B., « Focused crawling : a new approach to topic-specific Web resource discovery », *Computer Networks (Amsterdam, Netherlands : 1999)*, vol. 31, n°11–16, p. 1623-1640, 1999.

Charaudeau P., *Grammaire du sens et de l'expression*, Hachette, 1992.

Chiao Y.-C., Extraction lexicale bilingue à partir de textes médicaux comparables : application à la recherche d'information translangue, PhD thesis, Université Pierre et Marie Curie (Paris 6), juin, 2004.

Daille B., « Morphological Rule Induction for Terminology Acquisition », *COLING'00*, Sarrbrucken, Germany, p. 215-221, 2000.

Déjean H., Gaussier E., Sadat F., « An Approach Based on Multilingual Thesauri and Model Combination for Bilingual Lexicon Extraction », *COLING'02*, 2002.

Ducrot O., Todorov T., *Dictionnaire encyclopédique des sciences du langage*, Éditions du Seuil, 1999.

Fung P., McKeown K., « Finding terminology translations from non-parallel corpora », *Proceedings of the 5th annual workshop on very large corpora (VLC 97)*, Hong Kong, p. 192-202, 1997.

Fung P., Yee L. Y., « An IR Approach for Translating New Words from Nonparallel, Comparable Texts », *in* , C. Boitet, , P. Whitelock (eds), *COLING'98*, vol. 1, Montreal, Quebec, Canada, p. 414-420, 1998.

Habert B., Grabar N., Jacquemart P., Zweigenbaum P., « Building a text corpus for representing the variety of medical language », *in* , P. Rayson, , A. Wilson, , T. McEnery, , A. Hardie, , S. Khoja (eds), *Corpus Linguistics 2001*, Lancaster, p. 245-254, february, 2001.

Joachims T., *Learning to Classify Text using Support Vector Machines*, Kluwer Academic Publishers, 2002.

Karlgren J., *Natural Language Information Retrieval*, Tomek, Kluwer, chapter Stylistic Experiments in Information Retrieval, 1998.

Karlgren J., Cutting D., « Recognizing Text Genres with Simple Metrics Using Discriminant Analysis », *COLING'94*, vol. 2, Kyoto, Japan, p. 1071-1075, 1994.

Ke G., Zweigenbaum P., « Catégorisation automatique de pages web chinoises », *Actes de la 6ème Conférence en Recherche d'Informations et Applications (CORIA'09)*, 2009. À paraître.

Laviosa S., « Corpus-based Approaches to Contrastive Linguistics and Translation Studies », *Meta*, vol. 43, n°4, p. 474-479, 1998.

Malrieu D., Rastier F., « Genres et variations morphosyntaxiques », *Traitement Automatique des Langues (TAL)*, vol. 42, n°2, p. 548-577, 2002.

Matsumoto Y., Kitauchi A., Yamashita T., Hirano Y., Japanese Morphological Analysis System ChaSen 2.0 Users Manual, Technical report, Nara Institute of Science and Technology (NAIST), 1999.

McEnery A., Xiao Z., « Parallel and comparable corpora : What is happening ? », *in* , G. Anderman, , M. Rogers (eds), *Incorporating Corpora : The Linguist and the Translator*, Clevedon : Multilingual Matters, 2007.

Morin E., Daille B., Takeuchi K., Kageura K., « Bilingual Terminology Mining – Using Brain, not brawn comparable corpora », *ACL'07*, Prague, Czech Republic, p. 664-671, 2007.

Nakao Y., « Multilingual modalities for specialised languages », *Workshop on Multilingual and Comparative Perspectives in Specialized Language Resources (MCPSLR 2008), Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC 2008), Marrakech, Morocco, 26 May 2008*, European Language Resources Association (ELRA), 2008.

Namer F., Baud R., « Defining and relating biomedical terms : Towards a cross-language morphosemantics-based system », *International Journal of Medical Informatics*, vol. 76, n°2-3, p. 226-233, 2007.

Pearson J., *Terms in Context*, John Benjamins publishing company, 1998.

Peters C., Picchi E., « Using Linguistic Tools and Resources in Cross-Language Retrieval », *in* , D. Hull, , D. Oard (eds), *Cross-Language Text and Speech Retrieval. Papers from the 1997 AAAI Spring Symposium, Technical Report SS-97-05*, p. 179-188, 1997.

Querler N. L., *Typologie des modalités*, Presses universitaires de Caen, Caen, 1996.

Quinlan J. R., *C4.5 : Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.

Rapp R., « Automatic Identification of Word Translations from Unrelated English and German Corpora », *ACL'99*, College Park, Maryland, USA, p. 519-526, 1999.

Riboni D., « Feature Selection for Web Page Classification », *in* , H. Shafazand, , A. M. Tjoa (eds), *Proceedings of the 1st EurAsian Conference on Advances in Information and Communication Technology (EURASIA-ICT)*, Springer, Shiraz, Iran, p. 473-478, 2002.

Sager J. C., *A Pratical Course in Terminology Processing*, John Benjamins, Amsterdam, 1990.

Sebastiani F., « Machine Learning in Automated Text Categorization », *ACM Computing Surveys*, vol. 34, n°1, p. 1-47, 2002.

Sebastiani F., « Text Categorization », *in* , A. Zanasi (ed.), *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, WIT Press, Southampton, UK, p. 109-129, 2005.

Sinclair J., Preliminary recommendations on Text Typology, Technical report, EAGLES (Expert Advisory Group on Language Engineering Standards), 1996.

Chapitre 2

Recherche d'Information Multimodale

Une étude de l'impact de la structure sur la recherche multimedia

Mouna Torjmen, Karen Pinel-Sauvagnat

IRIT, Université Paul Sabatier
118, Route de Narbonne, 31400, Toulouse

RÉSUMÉ. *Cet article s'inscrit dans le cadre de la recherche XML multimedia, dont l'objectif est de trouver des fragments multimedia pertinents (c'est à dire des fragments XML contenant au moins un autre media que le texte). Dans des travaux précédents, nous avons proposé un modèle pour la recherche de fragments multimedia appliqué au media "image". Ce modèle consiste tout d'abord à trouver les images pertinentes et ensuite, à définir les fragments multimedia pertinents à partir de ces images. Dans cet article, nous nous intéressons plus particulièrement à la première partie du modèle où nous étudions l'impact de différents facteurs structurels pour la recherche d'images. Cette étude comparative est effectuée à travers une approche basée sur une analogie entre un document XML et une ontologie. Les facteurs sont évalués dans le cadre de la tâche Multimedia de campagne d'évaluation INEX 2007, et montrent l'intérêt de l'utilisation de la structure dans le processus de recherche multimedia.*

ABSTRACT. *In this paper, we are interested in XML multimedia retrieval, whose aim is to find relevant multimedia components(i.e XML fragments containing at least another media than text). The work described here is carried out with images, but can be extended to any other media. We proposed in previous work a multimedia fragment retrieval model which consists to retrieve in a first step relevant images and in a second step the best multimedia fragments through the retrieved images. In this paper, we are interested in the first step of our model. In fact, we study the impact of different structural factors on image retrieval. This comparative study is carried out through an approach based on an analogy between XML documents and ontologies. These factors are evaluated in the Multimedia Task of INEX 2007 and show the efficiency of using document structure in multimedia retrieval process.*

MOTS-CLÉS : *recherche d'information, fragment multimedia, texte, structure, image*

KEYWORDS: *information retrieval, multimedia fragment, text, structure, image*

1. Introduction

La recherche d'information dans des documents XML consiste à retrouver des fragments pertinents, c'est à dire des passages ou des éléments XML contenant des informations pertinentes. Bien que le média "*texte*" reste une composante dominante dans la majorité des documents XML, d'autres types de médias peuvent également être présents dans ces documents. Les études existantes concernant la recherche d'information multimédia ont montré qu'elle est loin d'être triviale dans le cas où l'utilisateur cherche une combinaison de médias (par exemple texte et image).

Dans cet article, le travail présenté est appliqué sur le média *"image"*. Toutefois, l'approche proposée peut être utilisée sur n'importe quel autre type de média.

La plupart des travaux existants dans le domaine de recherche d'images sont basés généralement soit sur le contenu textuel des documents contenant les images (Elghazel *et al.,* 2005) (Zhang *et al.,* 2005), soit sur les caractéristiques de bas niveau des images telles que la couleur et la texture (Lew *et al.,* 2006) (on parle alors de recherche d'images basée contenu -CBIR-). D'autres travaux proposent de combiner les deux pour utiliser les avantages de chaque approche (Iskandar *et al.,* 2005) (Tollari *et al.,* 2008). D'autres sources dévidence ont été récemment envisagées. Parmi elles on peut citer l'utilisation de ressources sémantiques comme les ontologies ou encore d'autres facteurs extraits des documents tels que les hyperliens. Dans cet article, nous proposons d'étudier l'impact de la structure sur la recherche d'images.

La principale différence entre la recherche XML adhoc et la recherche XML multimedia concerne les éléments retournés qui sont respectivement des fragments textuels et des fragments multimedia. Un fragment multimedia doit possèder un caractère multimedia, c'est à dire que les éléments retournés doivent être des éléments multimedia ou bien contenant au moins un élément multimedia (Tsikrika *et al.,* 2007b).

La plupart des techniques de recherche XML multimedia ne prennent pas en compte la spécificité multimedia d'une façon explicite : soit elles combinent les résultats XML adhoc avec les résultats de recherche d'images basée contenu (Iskandar *et al.,* 2005) (Tjondronegoro *et al.,* 2005) (van Zwol, 2005), soit elles filtrent les résultats XML adhoc en gardant ceux qui répondent au besoin multimédia (Tsikrika *et al.,* 2007a).

Dans (Torjmen *et al.,* 2008a), nous avons proposé un modèle qui prend en compte le caractère multimedia dans la recherche des fragments XML. Il consiste tout d'abord à rechercher les images pertinentes, et ensuite, à les utiliser pour trouver les bons fragments multimedia.

Dans cet article, nous évaluons et discutons plusieurs paramètres permettant de déterminer la pertinence des images. Nous abordons aussi quelques problématiques liées à l'évaluation des fragments multimedia et comment nous les avons résolues.

La suite de l'article est organisée comme suit : la section 2 présente un état de l'art sur la recherche de fragments XML multimedia. Dans la section 3, nous décri-

vons notre modèle, en se focalisant sur la première partie pour laquelle nous étudions l'impact de plusieurs facteurs sur l'évaluation de la pertinence des images. Des expérimentations et résultats sur la collection INEX 2007 sont présentés dans la section 4. Une discussion générale est menée dans la section 5, et enfin, quelques conclusions et perspectives sont décrites dans la section 6.

2. La recherche multimedia dans des documents semi-structurés

A l'origine, les systèmes de recherche d'information ont été conçus pour rechercher des documents entiers de type textuel, l'utilisateur devant lire toutes les informations des documents afin de trouver les parties qui l'intéressent. La recherche d'information structurée a apporté une réponse à ce problème, en utilisant la structure des documents et en renvoyant des éléments (noeuds) XML se focalisant sur le besoin de l'utilisateur.

Ces dernières années, avec le nombre croissant de média de type image, vidéo et son dans les documents, de nouvelles problématiques liées à l'inclusion de médias autre que le texte dans les documents semi-structurés sont apparues. Des fragments multimédia contenant à la fois du texte et un média autre que le texte doivent pouvoir être renvoyés aux utilisateurs.

Nos travaux se focalisent sur ce besoin, pour lequel nous décrivons quelques approches issues de l'état de l'art ci-dessous, avec des applications sur le média "*image*".

Jusqu'en 2005, où la campagne d'évaluation INEX [1] a donné naissance à une nouvelle tâche appelée tâche multimédia, offrant ainsi une plateforme d'évaluation de traitement de requêtes multimedia, peu de travaux se sont intéressés à la recherche multimedia (et plus précisément à la recherche d'images) dans des documents XML.

Parmi les premiers travaux proposés utilisant la structure XML pour la recherche d'éléments multimédia, citons ceux de (Kong *et al.*, 2005) (Kong *et al.*, 2007) qui consistent à diviser tout le contenu textuel du document XML en plusieurs *Region Knowledge* [2] *RKs* : *Self level RK* : *RK* du nœud multimedia ; *Sibling level RK* : *RK* des nœuds frères du nœud multimedia ; *1st ancestor level RK* : *RK* du premier ancêtre (parent) du nœud multimedia à l'exclusion du texte déjà utilisé ; *2nd ancestor level RK,..., Nth ancestor level RK*. Le modèle vectoriel est ensuite utilisé pour évaluer chaque *Region Knowledge*. Même si cette méthode exploite la structure verticale des documents, elle ne prend pas en compte la distribution des éléments contenus dans une même *Region Knowledge*.

D'autres travaux utilisent une combinaison linéaire des résultats obtenus par une recherche d'images basée sur le contenu (c'est à dire les caractéristiques de bas niveau des images) et une recherche textuelle. Dans (Iskandar *et al.*, 2006) par exemple, les auteurs ont proposé d'utiliser le système de recherche d'images par contenu *GIFT*

1. Initiative for the Evaluation of XML Retrieval. http ://inex.is.informatik.uni-duisburg.de/
2. Le contenu textuel de l'objet multimédia et des éléments l'entourant hiérarchiquement.

d'une part et le système de recherche textuel *Zettair* d'une autre part. La combinaison de ces deux systèmes n'a pas montré son intérêt dans la campagne d'évaluation INEX.

Une autre méthode proposée par l'équipe *CWI/UTwente* (Tsikrika *et al.*, 2007a) consiste à utiliser une méthode de recherche traditionnelle basée sur le modèle de langage en évaluant plusieurs priorités de longueur. Afin de respecter la spécificité multimedia, les résultats obtenus sont filtrés en ne gardant que les fragments contenant au moins une image. Ainsi, aucun traitement multimedia supplémentaire n'est effectué. Les meilleurs résultats retournés par cette méthode sont obtenus en ne renvoyant que des documents entiers.

Une autre approche proposée dans (Szlávik *et al.*, 2007) consiste à utiliser un réseau bayésien intégrant un modèle de langage, appliqué aux éléments et non aux documents, pour la recherche de texte et d'images. Cette méthode a été évaluée avec une petite collection (Lonely Planet d'INEX Multimedia 2005) et a montré son intérêt, même si des expérimentations avec une plus grosse collection (telle que la collection Wikipedia d'INEX, Tâche Multimedia Fragment 2006-2007) seraient nécessaires.

En conclusion, la recherche de fragments multimedia se réalise soit en combinant une recherche adhoc XML et une recherche d'images basée-contenu, soit par filtrage des résultats d'une recherche adhoc XML en ne gardant que les fragments contenant au moins une image. Peu de travaux tiennent compte à la fois de la structure des documents et de la spécificité multimédia. Nous présentons dans ce qui suit notre modèle qui vise à utiliser ces deux sources d'évidence.

3. Un modèle de recherche de fragments multimedia basée sur l'information textuelle et structurelle des documents

Dans (Torjmen *et al.*, 2009), nous avons proposé un modèle dédié à la recherche de fragments multimedia. La recherche s'effectue en deux étapes : (1) recherche des éléments images en utilisant le contenu textuel et structurel des documents, (2) détermination des fragments multimedia pertinents à partir de ces images. Le défi ici est de sélectionner le meilleur fragment multimedia qui doit être retourné.

3.1. *Représentation textuelle et structurelle d'éléments multimedia dans des documents semi-structurés*

Un document XML peut être représenté par un arbre où la racine est le document, les nœuds internes sont les nœuds représentant les éléments ou les attributs, et les nœuds feuilles sont les nœuds contenant les valeurs (texte, nom-image).

Dans des travaux précédents, nous avons proposé deux approches pour définir les éléments multimedia pertinents. La première (Torjmen *et al.*, 2009) consiste à évaluer un score pour les images en utilisant trois sources d'évidence : ses descendants, ses frères et ses ascendants ayant déjà des scores de pertinence précalculés par un sys-

tème de recherche XML adhoc. L'inconvénient de cette méthode est qu'elle est très dépendante du système de recherche adhoc utilisé.

La deuxième (Torjmen *et al.,* 2008a) consiste à représenter l'image via les nœuds textuels en se basant sur une analogie entre un document XML et une ontologie.

Cette dernière approche est basée sur deux intuitions : (1) chaque nœud textuel porte des informations permettant de représenter sémantiquement une image. Par conséquent, chaque élément textuel contenant des informations pertinentes doit participer à représenter l'image sémantiquement ; (2) certains nœuds textuels du document doivent participer plus que d'autres dans la représentation de l'image. En effet, l'apport de chaque nœud dans la représentation de l'image doit se calculer en fonction de la position hiérarchique de ce nœud par rapport à l'image.

La question qui s'impose derrière cette idée est : comment calculer la participation de chaque nœud textuel dans la représentation sémantique de l'image ?

Afin de répondre à la première intuition, pour utiliser l'information textuelle du document XML, nous avons calculé un score de pertinence pour chaque nœud feuille à partir d'un système de recherche XML classique, basé sur la formule *tf*idf*ief*.

Pour prendre en compte la deuxième intuition, nous avons utilisé l'information structurelle du document XML. La représentation arborescente d'un document XML nous permet de le considérer comme une ontologie très simplifiée où les nœuds sont les concepts qui sont organisés hiérarchiquement avec la relation ***est partie de***. Par exemple, *section* ***est partie de*** *article* et *paragraphe* ***est partie*** de *section*.

L'idée consiste à transposer une mesure de similarité sémantique utilisée entre les termes d'une ontologie pour calculer l'apport de chaque nœud à la représentation de l'image. Nous considérons le nœud image comme un concept C1, et le nœud à utiliser comme un autre concept C2 (Torjmen *et al.,* 2008b) (Torjmen *et al.,* 2008a).

Etant donné que l'image peut ne pas avoir ou en avoir très peu de contenu textuel, nous nous intéressons aux mesures de similarité basées sur les arcs et pas sur le contenu. Plusieurs mesures de similarités basées sur le nombre d'arcs entre les concepts sont proposées dans la littérature telles que celle de (Rada *et al.,* 1989), celle de (Hirst *et al.,* 1997) et celle de (Wu *et al.*, 1994).

La mesure de Wu-Palmer (Wu *et al.*, 1994), prenant en compte la position des concepts par rapport à la racine de l'ontologie est à la fois la plus simple à implémenter et la plus performante (Lin, 1998). Elle est définie comme suit :

$$Sim_{WP}(C_1, C_2) = \frac{2 * N_3}{(N_1 + N_2 + 2 * N_3)} \quad [1]$$

où N_1 et N_2 sont le nombre d'arcs qui séparent C_1 et C_2 de leur ascendant commun le plus spécifique *C*. N_3 est le nombre d'arcs qui séparent *C* de l'élément racine.

Cette mesure n'a cependant pas montré son intérêt dans des travaux précédents (Torjmen *et al.*, 2008a). Dans ce qui suit, nous proposons d'autres facteurs permet-

tant de prendre en compte la différence d'importance des nœuds dans l'arbre du document. Nous souhaiterions que les descendants de l'image participent plus que les descendants de son premier ancêtre puisqu'ils sont les plus spécifiques [3], que les descendants du premier ancêtre participent plus que les descendants du deuxième ancêtre puisqu'ils ont une forte probabilité de partager le même sujet avec l'image, etc. Les nœuds qui participeraient le moins à la représentation de l'image sont les descendants de l'élément racine puisqu'ils sont les plus loins de l'image.

La mesure de Wu-Palmer a été utilisée dans l'indexation sémantique des documents XML dans (Zargayouna, 2004). Cependant, les auteurs ont constaté qu'elle représente une limite car il est possible d'avoir la similarité entre un concept et son fils inférieure à la similarité entre ce concept et son frère alors qu'il était envisagé de ramener tous les fils d'un concept avant ses frères.

Pour éviter cela, les auteurs ont proposé de pénaliser les scores des frères en ajoutant une fonction $spec(C_1, C_2)$ qui calcule la spécificité de deux concepts par rapport au concept le plus bas (*bottom*) (voir Figure 1).

$$Sim_{WP}(C_1, C_2) = \frac{2 * N_3}{(N_1 + N_2 + 2 * N_3 * spec(C_1, C_2))} \quad [2]$$

$$où\ spec(C_1, C_2) = depth_b(C) * N_1 * N_2 \quad [3]$$

avec *C* est l'ancêtre commun le plus spécifique, $depth_b$ est le nombre maximum d'arcs qui séparent *C* de *bottom* (figure 1) et N_1 (N_2) est la distance en nombre d'arcs entre *C* et C_1 (C_2).

Comme le montre la figure 1, le facteur $depth_b$ utilise la structure hiérarchique verticale afin de différencier la participation des descendants de chaque ancêtre de l'image. Dans la figure 1, les descendants *F, M et S* de l'image *I* ont un $depth_b$ ($Depth_b$I) plus petit que celui de *B* ($Depth_b$B). Ce facteur $depth_b$ semble donc adéquat pour prendre en compte notre intuition dans la représentation sémantique de l'image.

Cependant, l'utilisation seule de ce facteur comme information structurelle conduit à égaliser la participation des descendants du même ancêtre de l'image. Afin de palier cet inconvénient, nous avons décidé de conserver aussi les facteurs N_1 et N_2 privilégiant ainsi les nœuds descendants les plus proches de l'image : plus les nœuds textuels sont loins de l'image, moins ils participent à sa représentation.

Prenons l'exemple des nœuds textuels *H* et *K* dans la figure 1, ils participent avec le même $depth_b$ ($Depth_b$B) et la même distance entre l'image *I* et l'ancêtre commun *B* (N_2^I) dans la représentation du nœud image *I*, mais le nœud *K* participe plus que le nœud *H* puisque que la distance N_1^K est plus petite que la distance N_1^H.

3. Les éléments images, outre le fait de donner l'URL du fichier image concerné, peuvent contenir d'autres éléments trés spécifiques, comme le nom de l'image et sa légende.

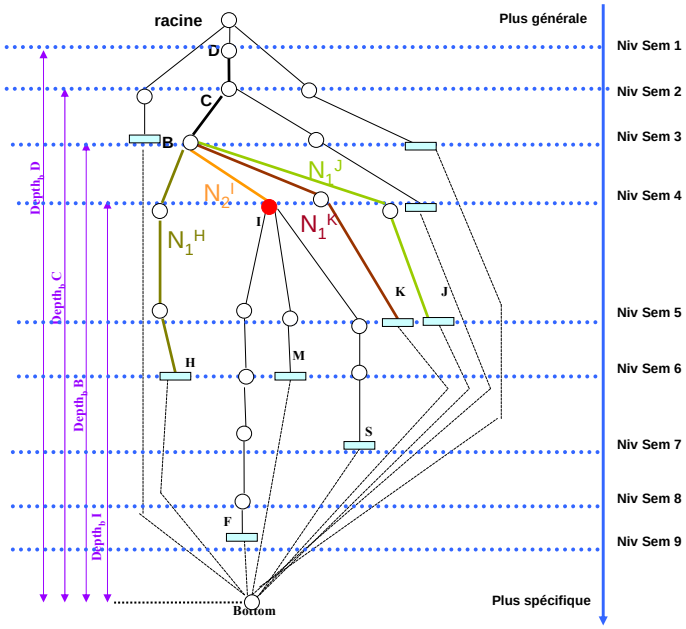


Figure 1. *Représentation d'un élément image en se basant sur l'information structurelle*

Dans nos travaux, nous proposons d'utiliser cette mesure pour calculer la participation de chaque nœud pertinent dans la représentation sémantique de l'image. Nous définissons la mesure de représentation sémantique comme suit :

$$Rep_{SPEC}(I, NT) = \frac{S_{NT}}{depth_b(C) * N_1 * N_2} \quad [4]$$

où *I* est le nœud image, *NT* est le nœud textuel qui participe à la représentation de l'image et S_{NT} est le score du nœud *NT*, calculé à l'aide d'un système de recherche d'information structurée classique et C est l'ancêtre commun entre *I* et *NT*.

Chacun des facteurs de cette formule sera évalué séparément dans la partie *Evaluation* (section 4.3).

Le score final de chaque image est calculé comme suit :

$$S(I) = \sum_{i=1}^{|NT|} Rep(I, NT_i) \quad [5]$$

avec NT_i un nœud textuel du document et $|NT|$ le nombre des noeuds textuels du document contenant l'image.

3.2. *Recherche de fragments multimédia à travers les images*

En recherche multimedia, le besoin utilisateur peut être un media comme une image, ou bien un fragment de document contenant au moins une image (Tsikrika *et al.,* 2007b). Par conséquence, dans notre cas, les résultats à retourner à l'utilisateur ne sont pas obligatoirement des images, mais ils peuvent être aussi des fragments multimedia (image + texte pertinent).

La problématique ici est de décider quels éléments doivent être retournés en se focalisant sur le besoin de l'utilisateur (*Focused Retrieval* dans la terminologie INEX). Les éléments retournés doivent être les plus exhaustifs et spécifiques possibles et ne doivent pas être imbriqués les uns dans les autres. Ce type de recherche suppose que l'utilisateur préfère l'élément (un seul) le plus pertinent d'un sous arbre pertinent (Kamps *et al.,* 2007).

Pour réaliser cet objectif, une méthode a déjà été proposée dans (Torjmen *et al.,* 2009). Elle consiste à définir pour chaque image retrouvée dans la première étape, un ensemble de fragments composé de l'image elle-même et ses ancêtres.

Le score de chaque ancêtre S_a^{im} (des images) est calculé en fonction de son score normalisé obtenu par un système adhoc XML (S_{Adhoc}) et des scores normalisés des images elles-même (S_{im}) contenues par cet ancêtre. La combinaison de ces deux scores se fait à travers une combinaison linéaire :

$$S_a^{im} = \gamma * S_{Adhoc} + (1 - \gamma) * \sum_{i=1}^{|NI|} S_{im_i} \qquad [6]$$

avec γ un pivot $\in$[0..1] et $|NI|$ le nombre d'images contenues dans l'ancêtre *a*.

4. Evaluation

Pour calculer un score de pertinence des nœuds textuels des documents, nous avons utilisé le système XFIRM (Pinel-Sauvagnat *et al.,* 2004) (Sauvagnat *et al.,* 2006). Ce système est aussi utilisé pour calculer un score de pertinence pour les nœuds ancêtres des images, utilisé dans l'équation 6.

4.1. *INEX : Collection et mesures d'évaluation*

INEX (Initiative for the Evaluation of XML Retrieval) est actuellement la seule campagne d'évaluation des différents systèmes de recherche d'information pour des documents XML. Le but principal d'INEX est de promouvoir l'évaluation de la recherche sur des documents XML en fournissant une collection de test, des procédures d'évaluation et un forum pour permettre aux différentes organisations participantes de comparer leurs résultats. La collection de test consiste en un ensemble de documents XML, requêtes et jugements de pertinence. Le langage de requêtes utilisé dans INEX est NEXI (Trotman *et al.,* 2005). Nous nous intéressons ici à la tâche multimedia qui

a eu lieu en 2007 pour la troisième fois, et plus particulièrement à la sous tâche Multimedia Fragments qui consiste à retrouver des fragments XML multimedia (contenant au moins une image). Des détails concernant cette tâche sont présentés dans (Westerveld *et al.*, 2006) (Tsikrika *et al.*, 2007b). La collection de cette tâche est la collection XML Wikipedia (Denoyer *et al.*, 2006), comprenant plus de 650 000 documents.

En 2007, 19 requêtes sont fournies pour la tâche Fragments Multimedia. Ces requêtes comportent plusieurs parties : une représentation textuelle par simples mots clés, une représentation textuelle et structurelle en ajoutant des contraintes structurelles, et finalement une représentation multimedia en ajoutant par exemple des concepts ou des images exemples.

Dans les travaux présentés dans cet article, seule la représentation textuelle simple des requêtes est utilisée.

La première partie de notre modèle consiste à retrouver les images pertinentes à partir d'une base de jugement de pertinence composée seulement d'images. Pour évaluer cette partie, nous avons créé une nouvelle base de jugements de pertinence à partir de la base originale de fragments multimedia, et ceci en gardant seulement les éléments images. Cette partie est évaluée grâce à la moyenne de la précision moyenne (MAP), en utilisant l'outil *trec-eval*.

La deuxième partie de notre approche consiste à retrouver des fragments multimedia pertinents. Elle est évaluée avec les mesures officielles de la tâche Fragments multimedia d'INEX 2007 (Kamps *et al.*, 2007). Deux mesures sont utilisées :

– ***La précision interpolée selon quatre niveaux de rappel sélectionnés :*** $iP[jR], j \in [0.00, 0.01, 0.05, 0.1]$

La précision à un rang r est défini comme suit :

$$P[r] = \frac{\sum_{i=1}^{r} rsize(p_i)}{\sum_{i=1}^{r} size(p_i)} \quad [7]$$

où p_r (p_i) est la partie du document assignée au rang r ($i <= r$) dans la liste de résultats L_q des parties de documents retournées par un système de recherche pour une requête q.

$rsize(p_r)$ est la taille du texte pertinent contenu dans p_r en nombre de caractères et $size(p_r)$ est la taille du texte totale contenu dans p_r en nombre de caractères.

Le rappel à un rang r est défini comme suit :

$$R[r] = \frac{\sum_{i=1}^{r} rsize(p_i)}{Trel(q)} \quad [8]$$

où $Trel(q)$ est la quantité totale du texte pertinent pour une requête q.

La mesure de précision interpolée $iP[x]$ est la suivante :

$$iP[x] = \begin{cases} max_{1 \leq r \leq |L_q|}(P[r] \wedge R[r] \geq x) \;\; if\ x \leq R[|L_q|] \\ \\ 0 \qquad\qquad\qquad\qquad\qquad if\ x \succ R[|L_q|] \end{cases} \quad [9]$$

où $R[|L_q|]$ est le rappel pour tous les documents restitués.

– ***La moyenne des précisions moyennes interpolées selon 101 niveaux de rappel***.

Supposons que nous avons n requêtes, $MAiP$ est calculée comme suit :

$$MAiP = \frac{1}{n}. \sum_{t} AiP(t) \qquad [10]$$

où Aip est la précision moyenne interpolée.

4.2. *Problèmes liés aux jugements de pertinence*

Alors que la différence principale entre la recherche XML adhoc et la recherche XML multimedia est que cette dernière a pour objectif de retourner des fragments documentaires pertinents contenant au moins une image (Tsikrika *et al.*, 2007b), les jugements de pertinence fournis par la campagne d'évaluation INEX 2007, tâche Fragments Multimedia ne respectent pas cette spécificité multimedia. En effet, nous avons constaté que 84,71% de ces jugements concernent des fragments purement textuels (c'est à dire ne contenant aucune image).

Par conséquent, nous ne pouvons pas évaluer la deuxième partie de notre méthode avec cette base de jugements de pertinence. Afin de palier cet inconvénient, nous avons décidé de filtrer ces jugements de pertinence en ne gardant que les fragments ayant au moins une image pertinente.

Conjointement à ce filtrage des jugements de pertinence, nous avons filtré les runs officiels des participants d'INEX 2007 puisque quelques uns renvoient également des fragments textuels purs, et ceci afin de les comparer à notre modèle. Suite à ce filtrage des runs officiels des participants d'INEX, le nombre de résultats retournés par leurs systèmes est diminué, et par conséquent, la comparaison selon la mesure MAiP n'est plus significative. Afin d'effectuer tout de même une comparaison, nous avons décidé de tracer les courbes Rappel/Précision interpolées selon les niveaux de rappel [0.00..0.05] et [0.1..1]. Nous nous intéressons plus particulièrement aux précisions dans les premiers niveaux de rappel puisque le nombre de résultats retournés n'est plus le même pour tous les runs.

4.3. *Résultats de la représentation sémantique des images par les nœuds textuels*

Afin d'étudier l'efficacité de la mesure de similarité Wu-Palmer (Equation 1) dans nos travaux ainsi que l'importance de chaque facteur de la formule 4, nous avons évalué tout d'abord le contenu textuel seul sans utiliser les facteurs de structure (Figure 2, formule *Cont-Text*), et ceci en sommant simplement les scores des nœuds textuels évalués par le système *XFIRM*. Ainsi, les images du même document auront tous le même score qui est la somme des scores du contenu textuel.

Nous avons ensuite évalué les scores des images en multipliant la mesure de Wu-Palmer représentée dans l'équation 1 par la formule *Cont-Text*. La valeur *MAP* de

cette mesure est représentée dans la figure 2 sous le nom *Cont-Text-Wu-Palmer*. Nous constatons que cette mesure permet une amélioration de 14.48% en MAP.

Pour évaluer l'impact de chaque facteur structurel sur la pertinence des images, nous les avons évalués séparemment en les multipliant par le score des nœuds textuels pré-calculé avec le système *XFIRM*.

La figure 2 montre les résultats des différents facteurs selon la mesure *MAP*.

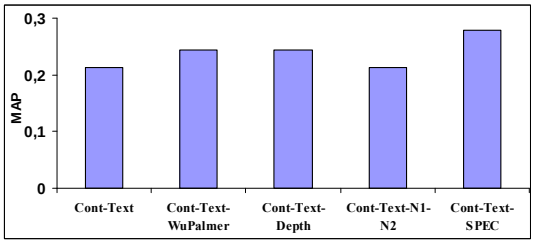


Figure 2. *Comparaison des différents facteurs pour la représentation des images*

Cont-Text-Depth consiste à multiplier le score de contenu textuel (*Cont-Text*) par le facteur $1/depth_b$. *Cont-Text-N1-N2* consiste à multiplier le score de contenu textuel (*Cont-Text*) par le facteur $1/(N_1 * N_2)$, et *Cont-Text-SPEC* consiste à multiplier le score de contenu textuel (*Cont-Text*) par les facteurs $1/depth_b$ et $1/(N_1 * N_2)$ (c'est à dire à utiliser l'équation 4).

Nous constatons tout d'abord que le facteur $1/depth_b$ joue un rôle important dans l'amélioration des résultats (*Cont-Text-Depth*). Ceci illustre l'importance de différencier la participation des nœuds textuels selon leur hiérarchie verticale avec l'image.

Nous constatons d'autre part que le facteur $1/(N_1 * N_2)$ seul (*Cont-Text-N1-N2*) n'apporte pas d'amélioration au contenu textuel, alors que c'est le cas en le multipliant par $1/depth_b$. Ces résultats confirment aussi notre intuition que plus la distance entre le nœud textuel et l'image est petite, plus ce nœud textuel doit participer à la représentation de l'image.

Finalement, en comparant les deux formules Wu-Palmer (*Cont-Text-WuPalmer*) et *spec* (*Cont-Text-SPEC*), nous constatons que cette dernière apporte une amélioration significative (+14.42% en *MAP*).

Le reste de nos expérimentations est basé sur les résultats obtenus par la formule *Cont-Text-SPEC* (Equation 4).

4.4. *Résultats de la recherche de fragments multimédia à travers les images*

Avant d'évaluer notre modèle, nous avons évalué les runs officiels des participants de la tâche Fragment Multimedia d'INEX 2007 avec la nouvelle base de jugements de

pertinence filtrée (en ne gardant que les fragments Multimedia, c'est à dire les fragments contenant au moins une image). Nous avons évalué également le meilleur run adhoc selon la tâche Multimedia en utilisant la mesure MAiP (run *MeilleurRunAdhoc-Indstaint-MAiP*), et le meilleur run adhoc selon la tâche Multimedia en utilisant la mesure iP[0.01] (run *MeilleurRunAdhoc-Mines-iP[0.01]*).

En effet, la différence majeure entre la tâche adhoc et la tâche multimedia dans INEX 2007 étant que les fragments retournés doivent avoir un caractère multimedia dans le second cas, les requêtes de la tâche Multimedia font partie de l'ensemble de requêtes de la tâche Adhoc. Les runs adhoc ont été ainsi évalués dans la cadre de la tâche Multimedia. Les courbes de rappel/précisions interpolées de ces 7 runs sont présentées sur la figure 3. Le meilleur run au niveau de rappel 0.01 est *MeilleurRunAdhoc-Mines-iP[0.01]*.

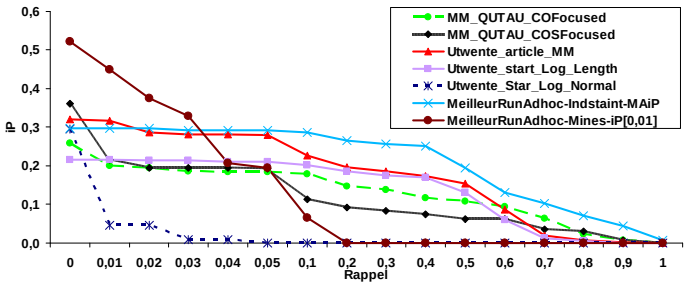


Figure 3. *Comparaison des runs officiels de la tâche Multimedia et des meilleurs runs Adhoc à INEX 2007 selon la nouvelle base de jugements filtrée*

La figure 4 donne les résultats de quelques uns de nos runs, à savoir le meilleur run en renvoyant à la fois des images ou des ancêtres d'images (run *Gamma1-Ancêtres-Images*), le meilleur run en ne renvoyant que des ancêtres d'images et jamais les images elles mêmes (*Gamma1-Ancêtres*), le run obtenu avec $\gamma = 0$, dans lequel seul le score des images est utilisé pour le calcul du score des ancêtres (run *Gamma0*), et enfin le run où nous utilisons seulement la première partie de notre modèle (c'est à dire que seuls les éléments images sont retournés) (*RunImages*).

Nous avons mené plusieurs expérimentations, non présentées ici, pour déterminer la meilleure valeur de γ dans l'équation 6. Selon la mesure officielle d'INEX 2007 (iP[0.01]), le meilleur valeur de γ est 1, lorsque nous renvoyons à la fois des images ou des ancêtres, ou lorsque nous renvoyons seulement des éléments ancêtres (runs *gamma1-Ancêtres-Images* et *gamma1-Ancêtres*). Ceci signifie que l'utilisation seule du score calculé par le système XFIRM est meilleure que la combinaison des deux scores. L'utilisation seule des scores des images n'a pas donné des bons résultats (run *gamma0*). En effet, la mesure iP [0.01] se dégrade de 37.26% par rapport à l'utilisation du score de XFIRM seul en renvoyant des images ou des ancêtres (run *gamma1-Ancêtres-Images*) et de 27.43% dans le cas où seulement des ancêtres d'images sont renvoyés (run *gamma1-Ancêtres*).

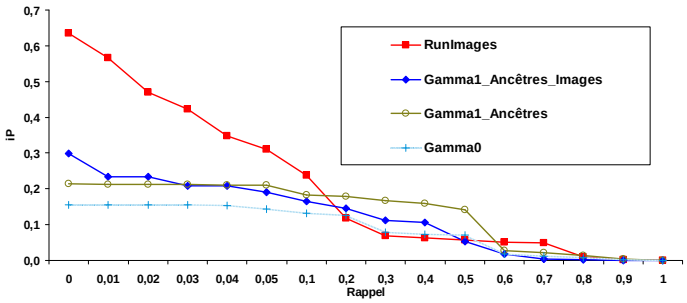


Figure 4. *Comparaison des différents runs de notre modèle*

Ceci peut être expliqué comme suit : chaque image va contribuer aux scores des ancêtres avec le même score (si un document contient une seule image, tous les ancêtres de cette dernière auront le même score). Dans nos expérimentations, si deux ancêtres ont le même score, on renvoie celui qui possède le plus haut niveau, et par conséquent, dans le cas de $\gamma = 0$, l'ancêtre qui possède plus d'images sera classé le premier. Ce comportement va ainsi pousser à renvoyer toujours l'élément ayant le plus haut niveau ("article").

Nous concluons de ces résultats que le score des images n'est pas bien utilisé dans l'évaluation de pertinence des ancêtres. En effet, la participation du score d'une image doit être différent d'un ancêtre à un autre. D'autres expérimentations sont nécessaires pour déterminer la bonne façon de combiner les scores des ancêtres précalculés par XFIRM et les scores des images obtenues dans l'étape 1 de notre modèle.

En comparant les deux runs *gamma1-Ancêtres-Images* et *gamma1-Ancêtres*, nous concluons que renvoyer à la fois des images ou des ancêtres d'images donne des résultats meilleurs que renvoyer seulement des ancêtres. Ceci signifie que l'image toute seule peut satisfaire correctement le besoin de l'utilisateur.

Enfin, en comparant les courbes selon les niveaux de rappel [0.0..0.1], nous constatons que les meilleures précisions interpolées sont obtenues avec le run qui ne renvoie que des images (*RunImages*). Ce résultat, qui peut paraître surprenant, ne l'est en fait pas lorsque on examine les jugements de pertinence. En effet, pour la collection INEX 2007, le pourcentage de fragments multimedia ayant une pertinence supérieure à celle de l'image qu'ils contiennent est de 3.48% seulement. Ceci montre encore un autre problème des jugements de pertinence d'INEX 2007 : la plupart des jugements de pertinence de fragments ont été basés sur la pertinence des images et non sur la pertinence des fragments multimedia (image + texte). Ceci implique que cette base de jugements est plus appropriée pour comparer des systèmes de recherche d'images dans des documents semi-struturées que pour comparer des systèmes de recherche de fragments multimédia. Afin d'évaluer d'une manière satisfaisante la deuxième partie de notre modèle, d'autres expérimentations sont donc nécessaires avec une autre collection prévilégiant suffisamment des fragments multimedia des éléments images.

4.5. *Discussion*

Une conclusion intéressante de ces expérimentations est que l'information structurelle permet bien d'améliorer la recherche d'images dans les documents semi-structurés (31% d'amélioration selon la mesure MAP dans l'étape 1 de notre modèle).

En se basant sur les figures 3 et 4, nous pouvons conclure également que l'utilisation d'une méthode spécifiée multimedia donne des résultats meilleurs qu'une méthode sans spécification multimedia dans le cas de recherche multimedia. Selon la mesure officielle d'INEX 2007, tâche Multimedia, nous obtenons le premier rang avec une amélioration de 78.74% par rapport au meilleur run Multimedia et 26.26% par rapport au meilleur run adhoc (iP[0.01]=0.5668 Versus iP=[0.01]=0.3171 pour le meilleur run multimedia et iP[0.01]=0.4489 pour le meilleur run adhoc). Ceci montre que les sytèmes XML adhoc, même adaptés à la recherche multimédia, ne sont pas suffisants pour répondre aux besoins multimédia des utilisateurs. Cela montre également que partir du contenu textuel et structurel est plus intéressant que chercher des fragments adhocs pertinents et de les filtrer selon un caractère multimédia.

Comme nous l'avons mentionné ci-dessus, la comparaison de notre modèle avec les autres selon la mesure MAiP n'est plus significative étant donné que le nombre de résultats n'est plus le même pour tous les runs. Cependant, à titre indicatif, notre meilleur run (runImages) sur la base de jugements de pertinence filtrée obtient une MAiP de 0.0932, alors que le meilleur run adhoc selon la mesure MAiP (*MeilleurRunAdhoc-Indstaint-MAiP*) obtient 0.1724.

Ces moins bons résultats peuvent être expliqués de la façon suivante : de bons éléments sont retournés par notre modèle aux premiers niveaux de rappel alors qu'ils sont retournés dans des niveaux de rappel plus élevés par le run adhoc, or, l'interpolation des précisions ne pénalise pas les systèmes renvoyant les bons résultats dans des niveaux de rappel éloignés par rapport aux systèmes renvoyant les bons résultats dans les premiers niveaux de rappel.

Afin d'illustrer ce problème, nous avons tracé respectivement la courbe Rappel/Précision non interpolée et la courbe Rappel/Précision interpolée de la requête 529 selon les deux runs (voir figure 5).

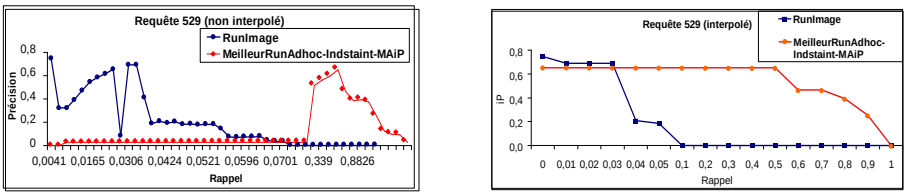


Figure 5. *Courbe Rappel/Précision non interpolée et interpolée de la requête 529*

La mesure MAiP montre ainsi ses limites : la mesure MAP (pas d'interpolation) serait mieux appropriée pour évaluer la performance globale des différents systèmes.

5. Conclusion et perspectives

Dans cet article, nous avons évalué l'impact de différents facteurs pour juger la pertinence des éléments multimedia dans des documents XML. Les expérimentations ont prouvé que l'information structurelle permet d'améliorer significativement la performance de recherche. De plus, les évaluations ont montré quelques problèmes liés à la base de jugement de pertinence. En effet, 84.71% des fragments jugés pertinents sont des fragments purement textuels ce qui ne respecte pas la spécificité de la tâche multimedia. D'autre part, la plupart des jugements de pertinence prévilégient les éléments images par rapport aux fragments multimedia (image et texte). Ceci ne permet pas de comparer les méthodes de recherche de fragments multimédia, mais plutôt des méthodes de recherche d'images dans des documents structurés.

Les perspectives envisageables à nos travaux consistent d'une part à améliorer la deuxième partie de notre modèle en testant d'autres méthodes de combinaison du score de l'image et du score des noeuds ancêtres, et d'une autre part à étudier autres facteurs pour la recherche des images tels que les liens et le nom de l'image.

6. Bibliographie

Denoyer L., Gallinari P., « The Wikipedia XML corpus », *SIGIR Forum 2006, p 64-69*, 2006.

Elghazel H., Idrissi K., Baskurt A., Amar C. B., « Approche textuelle pour la recherche d'image. », *3rd International Conference SETIT'05*, 2005.

Fuhr N., Lalmas M., Malik S., Kazai G., « INEX 2005 », 2005.

Fuhr N., Lalmas M., Trotman A., « INEX 2006 », 2006.

Fuhr N., Lalmas M., Trotman A., Kamps J., « INEX 2007 », 2007.

Hirst G., St-Onge D., « Lexical Chains as representation of context for the detection and correction malapropisms », 1997.

Iskandar D. N. F. A., Pehcevski J., Thom J. A., Tahaghoghi S. M. M., « Combining Image and Structured Text Retrieval », *INEX'05*, p. 525-539, 2005.

Iskandar D. N. F. A., Pehcevski J., Thom J. A., Tahaghoghi S. M. M., « Social Media Retrieval Using Image Features and Structured Text », in *INEX, p358-372* (Fuhr *et al.*, 2006), 2006.

Kamps J., Pehcevski J., Kazai G., Lalmas M., Robertson S., « INEX 2007 Evaluation Measures », in *INEX, p 24-33* (Fuhr *et al.*, 2007), 2007.

Kong Z., Lalmas M., « XML Multimedia Retrieval », *SPIRE 2005, p 218-223*, 2005.

Kong Z., Lalmas M., « Using XML Logical Structure to Retrieve (Multimedia) Objects », *ECDL 2007, p 100-111*, 2007.

Lew M. S., Sebe N., Djeraba C., Jain R., « Content-based multimedia information retrieval : State of the art and challenges », *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, p. 1-19, 2006.

Lin D., « An Information-Theoretic Definition of Similarity », *Proceedings of 15th International Conference On Machine Learning*, 1998.

Pinel-Sauvagnat K., Boughanem M., Chrisment C., « Searching XML documents using relevance propagation », *Symposium on String Processing and Information Retrieval (SPIRE'04)*, p. 242-254, 2004.

Rada R., Mili H., Bicknell E., Blettner M., « Development and application of a metric on semantic nets », *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, n° 1, p. 17-30, 1989.

Sauvagnat K., Boughanem M., Chrisment C., « Answering content and structure-based queries on XML documents using relevance propagation », *Journal Information Systems*, vol. 31, n° 7, p. 621-635, 2006.

Szlávik Z., Tombros A., Lalmas M., « Feature- and Query-Based Table of Contents Generation for XML Documents », *ECIR 2007, p 456-467*, 2007.

Tjondronegoro D., Zhang J., Gu J., Nguyen A., Geva S., « Integrating Text Retrieval and Image Retrieval in XML Document Searching », in *INEX'05, p 511-524* (Fuhr *et al.*, 2005), 2005.

Tollari S., Mulhem P., Ferecatu M., Glotin H., Detyniecki M., Gallinari P., Sahbi H., Zhao Z.-Q., « A Comparative Study of Diversity Methods for Hybrid Text and Image Retrieval Approaches », *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of CLEF*, 2008.

Torjmen M., Pinel-Sauvagnat K., Boughanem M., « Towards a structure-based multimedia retrieval model », *ACM International Conference MIR'08*, 2008a.

Torjmen M., Pinel-Sauvagnat K., Boughanem M., « Une métrique pondérée pour la recherche textuelle d'images dans des documents semi-structurés », *CORIA'08*,, p. 55-70, 2008b.

Torjmen M., Pinel-Sauvagnat K., Boughanem M., « XML Multimedia Retrieval : From relevant textual information to relevant multimedia fragments », *ECIR'09*, 2009.

Trotman A., Sigurbjörnsson B., Fuhr N., Lalmas M., Malik S., Szlavik Z., « Narrowed Extended XPath I (NEXI) », *Lecture Notes in Computer Science*, vol. 3493, Springer Verlag, Heidelberg, p. p 16-40, 2005.

Tsikrika T., Serdyukov P., Rode H., Westerveld T., Aly R., Hiemstra D., de Vries A. P., « Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah », in *INEX, p273-286* (Fuhr *et al.*, 2007), 2007a.

Tsikrika T., Westerveld T., « Report on the INEX 2007 Multimedia Track », in *INEX, p 410-422* (Fuhr *et al.*, 2007), 2007b.

van Zwol R., « Multimedia Strategies for ^{3}-SDR, Based on Principal Component Analysis », in *INEX'05, p540-553* (Fuhr *et al.*, 2005), 2005.

Westerveld T., Zwol R. V., « The INEX 2006 Multimedia Track », in *INEX, p 331-344* (Fuhr *et al.*, 2006), 2006.

Wu Z., Palmer M., « Verb semantics and lexical selection », *Proceedings of the 23nd Annual Meetings of the Associations for Computational Linguistics*, p. 133-138, 1994.

Zargayouna H., « Contexte et sémantique pour une indexation de documents semi-structrés », *COnference en Recherche d'Information et Applications*, p. 571-581, Mars, 2004.

Zhang C., Chai J. Y., Jin R., « User term feedback in interactive text-based image retrieval », *SIGIR'05*, p. 51-58, 2005.

Recherche par le contenu dans des documents audiovisuels multilingues

Georges Quénot* — **Tien Ping Tan*** — **Viet Bac Le*** — **Stéphane Ayache**** — **Laurent Besacier*** — **Philippe Mulhem***

** Laboratoire d'Informatique de Grenoble*
BP 53, 38041 Grenoble Cedex 9, France, Georges.Quenot@imag.fr
*** Laboratoire d'Informatique Fondamentale de Marseille*
163 avenue de Luminy - Case 901, 13288 Marseille Cedex 9, France

RÉSUMÉ. *Nous présentons dans cet article une approche basée sur l'utilisation de l'Alphabet Phonétique International (API) pour l'indexation et la recherche par le contenu de documents audiovisuels multilingues. L'approche fonctionne même si les documents contiennent des langues inconnues. Elle a été validée dans le cadre de la compétition « Star Challenge » sur les moteurs de recherche organisée par l'Agence A*STAR de Singapour. Notre approche comprend la construction d'un modèle acoustique multilingue basé sur l'API et une méthode fondée sur la programmation dynamique pour la recherche de segments de documents par « détection de chaînes API ». La programmation dynamique permet de repérer la chaîne de la requête dans la chaîne du document, même avec un taux d'erreur de transcription au niveau phonétique significatif. Les méthodes que nous avons développées nous ont classés premiers et troisièmes sur les tâches de recherche monolingues (anglais), cinquièmes sur la tâche de recherche multilingue et premiers sur la tâche de recherche multimodale (audio et image).*

ABSTRACT. *We present in this paper an approach based on the use of the International Phonetic Alphabet (IPA) for content-based indexing and retrieval of multilingual audiovisual documents. The approach works even if the languages of the document are unknown. It has been validated in the context of the "Star Challenge" search engine competition organized by the A*STAR Agency of Singapore. Our approach includes the building of an IPA-based multilingual acoustic model and a dynamic programming based method for searching document segments by "IPA string spotting". Dynamic programming allows for retrieving the query string in the document string even with a significant transcription error rate at the phone level. The methods that we developed ranked us as first and third on the monolingual (English) search task, as fifth on the multilingual search task and as first on the multimodal (audio and image) search task.*

MOTS-CLÉS : *Recherche audio, Multilingue, Alphabet Phonétique International, Programmation Dynamique, Star Challenge*

KEYWORDS: *Audio Retrieval, Multilingual, International Phonetic Alphabet, Dynamic Programming, Star Challenge*

.

1. Introduction

Les bases de données audiovisuelles contiennent souvent des documents en plusieurs langues. C'est le cas par exemple pour les archives sur Internet. Il arrive souvent que la langue utilisée dans un document soit inconnue et que le document contient des énoncés prononcés dans différentes langues. Cela complique la recherche par le contenu dans archives. Une possibilité consiste à appliquer un système de reconnaissance de la langue pour ensuite appliquer ls système de transcription approprié mais les détecteurs de langue commettent des erreurs et des langues inconnues peuvent être rencontrées. Une autre approche consiste à transcrire les documents phonétiquement en utilisant un sous-ensemble de l'Alphabet Phonétique International (API), quelle que soit la langue parlée. La recherche par le contenu peut alors être effectuée au niveau des chaînes de caractères en API. Cette approche a été encouragée par l'Agence de la Science, de la technologie et la recherche (A*STAR) de Singapour dans le cadre du défi « Star Challenge » qu'elle a organisé entre Mars et Octobre 2008[1]. Ce défi a également abordé le problème de la recherche dans des documents vidéo en utilisant uniquement l'image ou en utilisant des informations combinées de l'audio et de l'image.

Le défi « Star Challenge » est organisé comme une compétition pour les moteurs de recherche multimédia. Il est un peu différent dans l'esprit des campagnes d'évaluation classiques dans ce domaine telles que celles organisées par le NIST. Il s'agit vraiment d'une compétition dans des conditions proches de celles des applications du monde réel, en particulier en ce qui concerne les aspects temps de traitement (bien plus réduits). Elle est moins orientée vers une mesure précise de la performance des méthodes ou des systèmes. Le défi consiste en une série de trois rounds éliminatoires portant sur la recherche par le contenu dans des documents audiovisuels respectivement par l'audio, l'image et la combinaison des deux. Les cinq meilleures équipes classées après les trois rounds ont été invitées à participer à une épreuve finale « en direct » à Singapour. La tâche de recherche par l'audio existe en deux variantes : dans la première (AT1), la requête est fournie sous la forme d'une chaîne de caractères phonétiques (qui pourrait être entrée telle quelle par un utilisateur ou provenir d'une conversion à partir du texte) ; dans la seconde (AT2), la requête est fournie sous la forme d'un énoncé audio et doit être transcrite de la même manière que les documents audio. Nous avons profité de cette occasion pour développer et tester des approches innovantes pour la recherche par le contenu audio et multimodal dans des bases de recherche audio.

Nous décrivons dans cet article les méthodes que nous avons développés pour cette participation et de la façon dont nous les avons testées dans le cadre de ce défi. L'article est organisé comme suit : dans la section 2, nous décrivons comment nous avons construit nos modèles acoustiques multilingues ; dans la section 3, nous décrivons l'approche que nous avons utilisé pour l'API de recherche ; dans les sections 4 et 5, nous décrivons l'approche que nous avons utilisée pour la recherche visuelle et multimodal ; dans la section 6, nous décrivons les expériences que nous avons effectuées dans le cadre du Star Challenge et nous présentons les résultats obtenus.

1. http ://hlt.i2r.a-star.edu.sg/starchallenge

2. Traitement de l'audio

2.1. *Traitement des documents multilingues – approche générale*

Comme les langues parlées dans les documents audio sont supposées inconnues au départ, nous avons envisagé une approche multilingue pour la transcription automatique de documents audio. En effet, une solution aurait consisté à utiliser en parallèle différents systèmes de reconnaissance monolingues, mais celle-ci n'était pas réaliste dans le contexte de la compétition Star Challenge où le temps de calcul était une contrainte très importante.

Par conséquent, nous avons envisagé une approche plus « bas niveau » où un décodeur phonétique multilingue a été utilisé pour transcrire les documents et les requêtes. Ce décodeur a l'avantage d'être en principe indépendant de la langue et très rapide. En réalité, ce décodeur de phonème n'est pas tout à fait indépendant de la langue car il dépend d'un ensemble de langues cibles utilisées pour entraîner les modèles acoustiques et les modèles de langage « phonémiques ».

2.2. *Système de transcription automatique de la parole*

Pour chaque document audio, le signal audio a été extrait et segmenté en segments homogènes (parlés par un seul locuteur) en utilisant un système de segmentation audio fondé notamment sur le critère BIC (Bayesian Information Criterion (voir [MOR 04] pour plus de détails). En principe, un segment obtenu par ce système correspond à un tour de parole. Ensuite, un décodeur de parole a été appliqué sur chaque segment. Aucun détecteur de musique ou de silence n'a été utilisé ici pour enlever les segments ne contenant pas de parole.

Le décodeur Sphinx-3 [2] de Carnegie Mellon University (CMU) a été utilisé pour transcrire automatiquement des documents audio et des requêtes du round 1 (tâches de recherche vocale monolingue) et de la phase de Qualification (round 3) pour la finale (tâches de recherche vocale/vidéo multilingue). En fait, le décodeur Sphinx-3 est un décodeur rapide qui fonctionne en temps réel. Il implémente une stratégie de recherche en faisceaux via l'algorithme de Viterbi avec un contrôle de la largeur des faisceaux (Beam-Search) à plusieurs niveaux (état HMM, phonème, mot, ...). Sphinx-3 utilise les modèles acoustiques HMM créés et entraînés par SphinxTrain et il accepte en entrée des modèles de langage n-grammes ARPA standard au format binaire.

Un module de paramétrisation du signal a été utilisé pour extraire toutes les 10ms sur une fenêtre d'analyse un vecteur acoustique. Chaque vecteur acoustique consiste en 13 coefficients MFCCs, les dérivées première et seconde de ces coefficients pour obtenir finalement un ensemble de 39 paramètres. Toutes les unités acoustiques ont été construites sur une topologie de HMMs continus gauche-droit d'ordre 1 à 3 états

2. http ://www.speech.cs.cmu.edu/sphinx/

.

où chaque état est une distribution multi gaussienne. Pour apprendre les modèles de langage n-grammes, nous avons utilisé les boîtes à outils SRILM [STO 02] et CMU [CLA 97].

2.3. *Tâche monolingue*

Pour les tâches de recherche vocale (voice search) monolingue (anglais natif et dialectal), des modèles acoustiques anglais de 4000 états (tied-states) ont été utilisés. Chaque état a été modélisé par un mélange de 16 distributions gaussiennes à matrice de covariance diagonale. Ces modèles acoustiques ont été créés par Carnegie Melon University [PLA 97] et ils ont été appris à partir du corpus d'apprentissage broadcast news HUB-4 1996-1997 [LDC 97] qui contient 140 heures de signal de parole. Ensuite, ces modèles natifs anglais ont été adaptés par nos soins avec la méthode d'adaptation supervisée MAP en utilisant une petite quantité de données de parole dialectale de la région de l'Asie du Sud-Est. Par ailleurs, le modèle de langage HUB-4 [3] et le grand dictionnaire de prononciation de CMU [4] avec 125,000 mots ont été utilisés.

2.4. *Tâche multilingue*

Pour les tâches de recherche vocale multilingue, comme les langues parlées dans les documents audio sont supposées inconnues au départ, nous avons décidé de construire des modèles acoustiques multilingues pour 4 langues : anglais, mandarin, vietnamien et malais. Nous pensions que ces 4 langues seraient largement utilisées dans les documents audiovisuels dans la région de l'Asie du Sud-Est et la région Singapourienne en particulier.

Les modèles acoustiques multilingues indépendants du contexte sont entrainés séparément pour le mandarin, le vietnamien et le malais avec un mélange de 16 distributions gaussiennes pour chaque état du modèle HMM. Le modèle acoustique mandarin a été appris à partir du corpus CADCC [CCC 05], le modèle vietnamien a été appris à partir du corpus VnSpeechCorpus [LE 04] et le modèle malais a été appris à partir d'un corpus donné par l'Université Sains Malaysia. Un modèle acoustique anglais indépendant du contexte avec un mélange total de 16 distributions gaussiennes a été combiné à partir de deux modèles acoustiques différents : HUB-4 (issu de CMU, de type broadcast news) avec 8 gaussiennes et WSJ0 avec 8 gaussiennes (de type parole lue) [LDC 93]. Comme le modèle HUB-4 est originalement un modèle dépendant du contexte, nous n'avons extrait que les parties indépendantes du contexte à partir de ce modèle. Le tableau 1 présente le nombre de locuteurs et la taille des corpus de parole utilisés.

---

3. http ://www.speech.cs.cmu.edu/sphinx/models/
4. http ://www.speech.cs.cmu.edu/cgi-bin/cmudict

Enfin, un modèle acoustique multilingue est composé à partir des 4 modèles acoustiques monolingues. Les étiquettes linguistiques ont été ajoutées à chaque modèle de phonème pour que les phonèmes venant de langues différentes puissent être différenciés le cas échéant.

Corpus	Description	Nb. loc.	Heures
HUB4	Anglais, broadcast news	–	140
WSJ0	Anglais, parole lue	123	15
VN	Vietnamien, parole lue	29	15
CADCC	Mandarin	20	5
MSC	Malais	18	5

Tableau 1. *Corpus de parole utilisés pour la modélisation acoustique multilingue*

Pour la modélisation du langage, un modèle de phonème multilingue bigramme a été appris à partir du corpus de texte multilingue pour 4 langues. L'utilisation du modèle de langage phonétique a accéléré significativement le décodage de parole (le temps de calcul était environ de $0.25\times$RT).

3. Recherche par programmation dynamique

La recherche est toujours effectuée au niveau des chaînes de caractères API, indépendamment du fait que la transcription ait été faite au niveau du mot ou du phonème et indépendamment du fait que la requête soit présentée comme une chaîne de caractères API (AT1) ou comme un énoncé vocal (AT2). Dans tous les cas, nous devons déterminer un alignement optimal et un score associé entre les représentations en API des requêtes et des documents. Nous avons pour cela adapté un algorithme de détection de mots dans un flot de parole continue[GAU 82]. La principale différence entre l'algorithme original de détection de mots de notre algorithme de détection de chaînes en API est de remplacer les vecteurs de caractéristiques audio (en général les « Mel Frequency Cepstral Coefficients » ou MFCCs) par des symboles de l'API.

3.1. *Minimisation de la distance d'édition*

À cause de fréquentes erreurs de transcription, soit dans les documents pour les deux tâches, soit dans les requêtes pour la tâche AT2, la recherche de la chaîne phonétique de la requête dans celle d'un document doit permettre une correspondance inexacte. Que la correspondance soit exacte ou inexacte, il faut également lui attribuer un score afin de pouvoir classer en premier les documents pour lesquels la correspondance est la plus exacte. Afin de permettre les correspondances inexactes et d'attribuer un score à celles-ci, nous avons choisi de modifier la « distance » entre la chaîne de la requête et une sous-chaîne d'un document. Toutes les correspondances possibles entre la chaîne de la requête et l'ensemble des sous-chaînes d'un document sont prises en compte

.

et, pour chacune de ces correspondances, une distance est calculée en comptant et en pénalisant l'ensemble des insertions, des suppressions et des substitutions entre la chaîne phonétique de la requête et la sous-chaîne du document. La Figure 1 montre un exemple de modifier le calcul de distance.

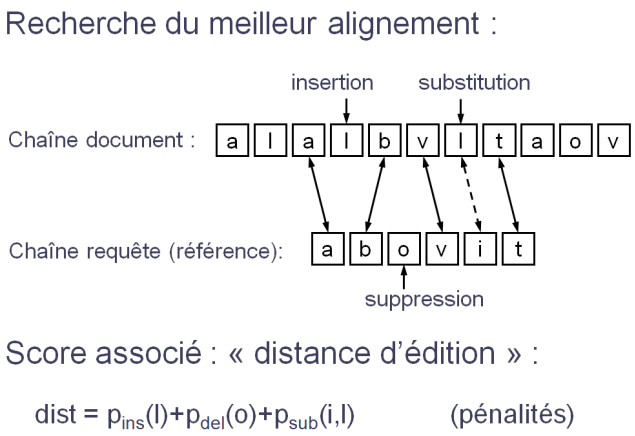


Figure 1. *Calcul du score d'une correspondance entre la chaîne de la requête et une sous-chaîne d'un document*

3.2. *Programmation dynamique*

La programmation dynamique est un moyen de résoudre le problème de trouver le meilleur alignement et le score correspondant entre une chaîne requête et une chaîne document avec un temps de calcul linéaire avec la longueur de la chaîne requête et avec la longueur de la chaîne document.

Considérons la matrice produit de la chaîne de caractères représentant le document (horizontalement) et la chaîne représentant la requête (verticalement). Une correspondance (ou un alignement) valide entre la chaîne de la requête et une sous-chaîne du document est un chemin « continu et croissant » qui relie la rangée du bas de la matrice à la rangée du haut de la matrice (Figure 2). Le meilleur alignement (ou chemin) est celui qui minimise la distance d'édition le long de lui-même. L'astuce de la programmation dynamique est de calculer le meilleur alignement par récurrence.

Si nous considérons la distance d'édition $e(i, j)$ selon le chemin optimal joignant la ligne du bas de la matrice au point (i, j) dans celle-ci, nous avons une équation de récurrence sur $e(i, j)$ ca le chemin optimal arrivant en (i, j) doit :

– soit venir de $(i - 2, j - 1)$ avec une pénalité d'insertion,
– soit venir de $(i - 1, j - 2)$ avec une pénalité de suppression,

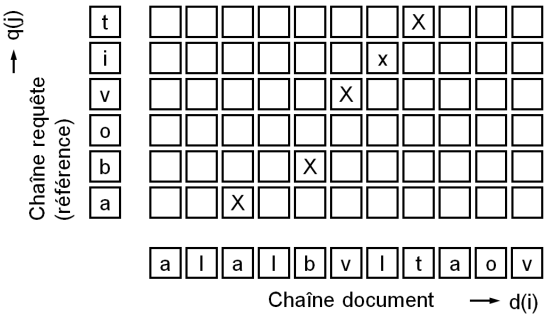


Figure 2. *Chemin d'alignement dans ma matrice de programmation dynamique*

– soit venir de $(i-1, j-1)$ avec une pénalité de substitution éventuelle.

(à moins que l'un de ces points ne soit en dehors de la matrice).

$e(i,j)$ peut être calculé par récurrence dans la matrice complète en initialisant $e(i,j)$ à 0 sur la rangée du bas et à « infini » sur la colonne de gauche (à l'exception de la valeur du bas). L'équation de récurrence effectivement utilisée est donnée en Eq. 1. Les c_{xx} sont des constantes dont les valeurs sont : $c_{ii} = c_{dd} = 2.0$, $c_{sn} = c_{sd} = 1.0$, $c_{si} = 0.5$ (normalisation selon la requête de façon à ce que tous les alignements aient le même poids total et poids identiques pour les pénalités d'insertion, de suppression et de substitution).

$$e(i,j) = \min \left\{ \begin{array}{l} e(i-2,j-1) + c_{si}(p_{sub}(d(i-2), q(i-1)) \\ \qquad\qquad +p_{sub}(d(i), q(i)) + c_{ii}p_{ins}(d(i-1))) \\ e(i-1,j-1) + c_{sn}(p_{sub}(d(i-1), q(i-1)) \\ \qquad\qquad +p_{sub}(d(i), q(i))) \\ e(i-1,j-2) + c_{sd}(p_{sub}(d(i-1), q(i-2)) \\ \qquad\qquad +p_{sub}(d(i), q(i)) + c_{dd}p_{del}(q(i-1))) \end{array} \right\} \quad [1]$$

Une fois terminé, le minimum de $e(i,j)$ sur la rangée du haut donne la meilleure distance d'édition qui est aussi le score du document pour la requête (le document avec le score le plus faible est le meilleur). Le retour en arrière à partir de la position du minimum donne l'alignement complet et donc la position dans le document de l'instance de la requête avec la meilleure correspondance.

3.3. *Pénalités fixes et variables*

Les pénalités d'insertion, de suppression et de substitution peuvent soit être constantes soit dépendre des phonèmes effectivement insérés, supprimés ou substitués, certains

.

phonèmes étant en effet plus susceptibles que d'autres d'être insérés, supprimés ou substitués. Pour les pénalités fixes, nous avons choisi :

- $p_{ins(p_i)} = 1$
- $p_{sub(p_i,p_j)} = 1 - \delta(i,j)$
- $p_{del(p_j)} = 1$

et pour les pénalités variables, nous avons choisi :

- $p_{ins(p_i)} = -\log(\epsilon + prob(insertion(p_i))$
- $p_{sub(p_i,p_j)} = -\log(\epsilon + prob(substitution(p_i,p_j)))$
- $p_{del(p_j)} = -\log(\epsilon + prob(deletion(p_j))$

Les probabilités ont été estimées par comparaison de transcriptions manuelles et automatiques.

4. Recherche par le contenu visuel

Nous utilisons la modalité visuelle pour la classification en concepts via apprentissage supervisé exploitant les annotations fournies par les organisateurs du Star Challenge. Notre système de classification est générique dans la mesure où la même approche est développée pour détecter tous les concepts visés. L'approche met en œuvre des réseaux d'opérateurs qui incluent des extracteurs de descripteurs bas niveau, des détecteurs de concepts intermédiaires et des opérateurs de fusion [AYA 07]. Les sections suivantes décrivent ces étapes.

4.1. *Analyse visuelle*

Le flux visuel est analysé à plusieurs niveaux de granularité ; des descripteurs globaux représentent l'ensemble d'une image, tandis que des descripteurs locaux sont extraits dans des blocs entrelacés, selon une grille de N×M blocs. La participation au Star Challenge posant une contrainte de temps d'exécution, nous avons fixé la granularité par descripteurs empiriquement, de façon à obtenir des taux de classifications satisfaisant avec des temps de calculs réduits. Ces descripteurs sont utilisés pour l'apprentissage et la classification en concepts des séquences vidéo. L'analyse visuelle traite une à plusieurs image clé par séquence vidéo, puis les combine selon les schémas de fusions classiques « précoce », « tardif », ou une combinaison des deux, puis après classification des images clé, attribue un score par concept pour chaque séquence vidéo.

4.1.1. *Descripteurs bas niveau*

Nous considérons des descripteurs de couleur et texture globaux à l'image. La couleur est représentée par un histogramme 3D dans l'espace RGB, où l'espace de couleur est discrétisé de façon à obtenir un histogramme de 4×4×4 dimensions. La texture est

extraite à l'aide de 40 filtres de Gabor sur 8 orientations et 5 échelles. Finalement, un descripteur visuel global est normalisé pour former un vecteur de 104 dimensions.

Pour compléter ces descriptions, nous extrayons d'autres descripteurs visuels dans chaque bloc d'image. Ces descripteurs sont alignés pour former une description visuelle riche pour chaque image clé :

Couleur (1) : décrit par un histogramme 3D de 3×3×3 dimensions, extraits dans une grille de 8×6 blocs. Ce descripteur forme un vecteur de 1296 dimensions.

Couleur (2) : décrit par les deux premiers moments statistiques, extraits dans une grille de 8×6 blocs. Ce descripteur forme un vecteur de 432 dimensions.

Histogramme d'orientation : calculé dans une grille de 4×3 blocs. Chaque dimension correspond à la somme des magnitudes d'une orientation. Nous considérons 50 orientations. Le descripteur EDH forme un vecteur de 600 dimensions, il est connu pour être invariant en échelle et en translation.

Local Binary Pattern : calculé dans une grille de 2×2 blocs et constitue un vecteur de 1024 dimensions. Le descripteur LBP modifie la valeur d'un pixel selon les valeurs des pixels voisins (3×3) pour capter des motifs de texture. LBP est invariant par une variation monotone de la valeur des pixels, ce qui est intéressant pour résister aux variations d'illumination [M¨ 00].

4.1.2. *Descripteurs « sacs de mots »*

La représentation d'images par sac-de-mots consiste à sélectionner un ensemble de régions dans une image (points d'intérêt) puis à décrire chacune d'elles à l'aide d'un descripteur visuel. Ces descripteurs sont alors quantifiés en affectant chaque descripteur à un élément d'un vocabulaire visuel pré-calculé. Cela permet d'obtenir un histogramme qui comptabilise les occurrences des mots visuels (éléments du vocabulaire visuel) dans une image. Combinée avec les descripteurs SIFT, invariants en échelle et en rotation, cette approche constitue l'une des approches les plus discriminantes pour la classification d'images [LOW 04]. Nous avons utilisé un dictionnaire de 1000 mots visuels, fourni par le groupe INRIA-LEAR.

4.2. *Descripteur sémantique*

Ce descripteur vise à modéliser les relations sémantiques entre les concepts, par une approche sac-de-mots. Ce descripteur nécessite une phase d'apprentissage sur les blocs d'image. Pour cela, nous considérons chaque bloc d'image comme positifs relativement à un concept lorsque l'image est annotée positivement. Cette hypothèse est certes très forte mais peut être raisonnable pour certains concepts. Nous entraînons des modèles par concept au niveau bloc sur une partie de l'ensemble d'apprentissage puis classons l'ensemble des blocs, qui aboutie à $nb\_blocs \times nb\_concepts$ scores de classification par image. Le descripteur sémantique est représenté par un histogramme de $nb\_concepts$ dimensions où chaque dimension contient la somme des scores d'un concept sur tous les blocs.

.

4.3. *Classification et fusion*

À partir des descripteurs visuels décrits ci-dessus, nous avons entraîné des classifieurs SVM à noyaux RBF pour la classification par concepts. Le choix des paramètres gamma et C par concept est fixé par validation croisé. Les différents descripteurs sont fusionnés par combinaison des schémas de fusion « précoce » et « tardif ». Une fusion précoce opère dans l'espace des descripteurs, tandis que la fusion tardive combine les scores de classifications obtenus par chaque classifieurs. Une combinaison de ces schémas de fusion est possible lorsque plus de deux descripteurs est disponibles et apporte plus de flexibilité pour combiner les descripteurs. Par exemple, il est possible de fusionner séparément des descritpeurs couleur et texture de façon précoce, puis de fusionner les scores de classifications obtenus de chacun de façon tardive. De telles combinaisons améliorent significativement les performances de classification pour certains concepts.

Nous avons implémenté les opérateurs de fusions tel que l'opérateur de fusion précoce normalise les descripteurs et de les aligne pour former un seul descripteur. L'opérateur de fusion tardive effectue une combinaison linéaire (moyenne) des scores de classification.

Pour notre participation au Star Challenge, nous avons mis en œuvre plusieurs réseaux d'opérateurs (i.e. plusieurs combinaisons de schémas de fusion, faisant intervenir différents descripteurs). Afin d'optimiser le choix du réseau optimal par concept visé, nous avons choisi, pour chaque concept, le réseau qui maximise la performance de classification sur un corpus de développement.

5. Recherche par le contenu multimodale

Les plans vidéo peuvent être évalués et triés en fonction de :

– la probabilité de présence d'un concept donné ;
– la similarité visuelle à une image ou à un plan vidéo donné ;
– la probabilité de présence d'une chaîne phonétique donnée.

Une requête mono ou multimodale peut être définie comme une combinaison de tels critères. Par exemple, dans les tâches vidéo 1 et 2 (VT1 and VT2) du Star Challenge, une requête est définie comme une combinaison d'un concept visuel requis et d'une similarité visuelle à une image (VT1) ou à un plan vidéo (VT2) donné. Dans les tâches de recherche multimodales 1 et 2 (AV1 and AV2), une requête est définie comme une combinaison d'une similarité visuelle à une image et de la présence d'une chaîne phonétique donnée textuelle (AV1) ou parlée (AV2).

Une approche similaire est utilisée dans tous les cas. Un score numérique est obtenu pour chaque critère en utilisant le sous-système approprié : recherche par chaîne phonétique, recherche par énoncé vocal, recherche par concept pré indexés ou recherche

par similarité visuelle à un exemple donné. Ces scores sont normalisés indépendamment pour chaque critère par une simple correction de moyenne et d'écart-type. Une somme pondérée est ensuite calculée et celle-ci est utilisée pour trier les plans vidéo. Les poids optimaux sont choisis comme ceux qui maximisent la performance du système sur l'ensemble de développement.

La similarité visuelle est basée sur une distance Euclidienne sur les mêmes descripteurs de couleur, de texture et de mouvement que ceux qui sont utilisés pour la classification de concepts. La similarité visuelle est calculée séparément pour chaque caractéristique et les scores correspondants sont normalisés et combinés de la même façon que pour les composants multimodaux. Là encore, les poids optimaux sont choisis comme ceux qui maximisent la performance du système sur l'ensemble de développement.

6. Expérimentations

6.1. *Recherche phonétique Monolingue, validation sur la collection de développement du Star Challenge*

L'objectif de la première série d'expériences était d'évaluer la performance relative des recherches basées sur une reconnaissance au niveau du mot et des recherches basées sur une reconnaissance au niveau du phonème ainsi que sur le bénéfice apporté par l'utilisation de pénalités variables dans le second cas. Trois méthodes basées sur la programmation dynamique (PD) sont comparées à quatre méthodes basiques de référence (baselines).

Ces expériences ont été menées sur la collection de développement audio du Star Challenge. Cette collection comprend environ deux heures (233 segments) de données audio monolingues (en anglais) et de 39 requêtes résolues à la fois pour les tâches AT1 (requêtes par chaîne en API) et AT2 (requêtes parlées). Les systèmes doivent retourner une liste de 50 réponses et la mesure d'évaluation est définie comme le MAP du Star Challenge pour la recherche par l'audio (Eq. 2 ; ce MAP est différente de la norme TREC paramètres MAP) :

$$MAP = \frac{1}{L} \sum_{i=1}^{L} \left(\frac{1}{R_i} \sum_{j=1}^{R_i} \delta(i,j) \right) \qquad [2]$$

où L est le nombre total de requêtes, R_i est le nombre total de documents pertinents pour la $i$ème requête, et $\delta(i,j)$ est une fonction indicatrice qui vaut 1 pour les bonnes réponses (i.e. le $j$ème document pertinent est dans la liste de résultats pour la requête i) et 0 sinon.

Les documents (segments audio) et les requêtes AT2 ont été transcrits en API de deux façons. La première est une transcription au niveau mot suivi par une conversion des

.

mots en phonèmes. La seconde est une transcription directement au niveau phonétique. Après cela et dans les deux cas, tous les documents et toutes les requêtes AT2 sont représentés par des chaînes en API.

Plusieurs méthodes basiques de référence ont été utilisées pour la comparaison. Une réponse aléatoire est un choix naturel et ceci constitue la « baseline 2 ». Une autre possibilité est de trier les segments en fonction de leur longueur, les segments les plus longs ayant le plus de chances a priori de contenir la requête. La « baseline 1 » correspond au choix des segments les plus court (pire cas) et « baseline 3 » correspond au choix des segments les plus longs (meilleur cas). Ces trois références ignorent le contenu des documents (segments) comme celui des requêtes et les résultats sont les mêmes sur les tâches AT1 et AT2. La « baseline 4 » consiste en la recherche d'une présence exacte de la chaîne requête dans la chaîne document. Elle est équivalente à la commande Unix « grep » et aussi à une programmation dynamique avec des pénalités d'insertion, de suppression et de substitution infinies. Comme les correspondances exactes sont assez rares, les listes de résultats sont complétées par les segments restants les plus longs.

La programmation dynamique (PD) a été essayée avec une reconnaissance au niveau du mot avec des pénalités fixes et variables et avec une reconnaissance au niveau phonétique avec des pénalités variables.

Méthode	AT1	AT2
Baseline 1 : segments courts	0.024	0.024
Baseline 2 : hasard	0.242	0.242
Baseline 3 : segments longs	0.497	0.497
Baseline 4 : « grep » + segments longs	0.557	0.560
PD, rec. mot, pénalités fixes	0.776	0.632
PD, rec. mot, pénalités variables	0.843	0.636
PD, rec. phon., pénalités variables	0.706	0.650

Tableau 2. *Validation sur la collection de développement du Star Challenge*

Le tableau 2 montre les résultats obtenus pour les méthodes testées et les méthodes de référence. Les observations suivantes peuvent être faites :

– les performances des baselines sont ordonnées comme prévu : courts $<$ hasard $<$ longs $<$ grep+longs ;

– les pénalités variables améliorent les performances de manière significative ;

– comme prévu également, la reconnaissance purement phonétique donne de moins bons résultats car elle ne bénéficie pas du modèle de langue au niveau mot ; elle est cependant la seule disponible pour les documents contenant des langues inconnues et c'est celle qui sera utilisée pour la recherche multilingue.

6.2. *Recherche phonétique Monolingue, évaluation sur la collection « round 1 » du Star Challenge*

Le système ayant eu la meilleure performance sur la collection de développement a été utilisé pour la soumission officielle pour le « round 1 » du Star Challenge. Ce système utilise la programmation dynamique avec des pénalités variables. Une amélioration supplémentaire a été apportée ; elle consiste en l'utilisation de trois transcriptions différentes avec des poids différents pour les bigrammes de phonèmes en faisant une moyenne des trois scores obtenus.

Collection	Pénalités	AT1	AT2	Moyenne
Dével.	fixes	0.760	0.679	0.719
Dével.	variables	0.858	0.728	0.793
Round 1	fixes	0.643	0.319	0.481
Round 1	variables	0.634	0.324	0.479

Tableau 3. *Influence de l'utilisation de pénalités variables*

Le tableau 3 montre les résultats obtenus pas ce système sur la collection du « round 1 ». Cette collection est composée de 25 heures (4300 segments) de documents audio monolingues (en anglais) et de 10 requêtes résolues pour les tâches AT1 et AT2. Les résultats correspondants sont aussi montrés pour le même système avec des pénalités fixes sur les données du round 1 et avec des pénalités fixes et variables sur les données de développement audio. Les observations suivantes peuvent être faites :

– les performances sur les données du round 1 sont très inférieures à celles obtenues sur les données de développement ;

– le gain de performance significatif obtenu par l'utilisation de pénalités variables sur les données de développement data ne se retrouve pas sur les données du round 1 ;

– la chute de performance entre AT1 et AT2 est beaucoup plus importante sur les données du round 1.

Tous ces effets sont probablement liés au fait que les données du round 1 contiennent beaucoup plus de segments dont une proportion beaucoup plus faible est pertinente, ce qui rend la tâche plus difficile. Les segments sont également plus courts. En utilisant cette approche, l'équipe LIG a terminé première sur AT1 et troisième sur AT2 parmi 35 équipes participantes.

6.3. *Recherche phonétique Multilingue, validation sur la collection « round 1 » du Star Challenge*

Le but de cette série d'expériences est de valider la recherche multilingue en utilisant les données d'entraînement disponibles. Puisque les langues cibles ne sont pas connues, nous n'avons pu valider l'approche que sur les données monolingues (en anglais) disponibles. Nous avons toutefois construit des modèles en utilisant d'autres

.

langues et noues les avons testés sur des données en anglais, en considérant que cela serait suffisamment représentatif. Nous avons d'abord essayé de construire des modèles à partir de langues uniques et de les utiliser pour l'indexation et la recherche : l'anglais (EN), mandarin (CH) et le malais (MY). Nous avons aussi essayé un modèle qui est une combinaison de ces trois langues et du Vietnamien (ML4). Nous avons enfin essayé la même chose avec des modèles de langue à base de bigrammes phonétiques (BG) et une combinaison de trois modèles avec des pondérations différentes des modèles de langue (Fuse).

	EN	CH	MY	ML4	BG	Fuse
AT1	0.668	0.476	0.428	0.603	0.615	0.650
AT2	0.585	0.578	0.577	0.568	0.591	0.638

Tableau 4. *Résultats obtenus avec différents modèles de langue.*

Le tableau 4 montre les résultats obtenus avec ces différents modèles. Les observations suivantes peuvent être faites :

– nous avons utilisé un nouveau modèle de l'Anglais qui s'est révélé être meilleur que celui que nous avions utilisé pour notre soumission officielle sur le round 1, en particulier sur la tâche AT2 ;

– les résultats obtenus avec les modèles de langues différents de celui de la langue cible (en Mandarin ou en Malais au lien de l'Anglais) sont très bons bien que les contenus phonétiques soient très différents ;

– les résultats sont meilleurs pour AT2 que pour AT1 dans ce cas, ce qui est probablement dû au fait que des confusions similaires sont faites au cours de la transcription des documents et des requêtes et qu'elles se compensent les unes les autres ;

– le modèle multilingue ML4 est presque aussi bon que le modèle purement Anglais et les modèles BG et Fuse font encore mieux alors que, de par la façon dont ils sont construits, ces modèles devraient être aussi bon pour les langues asiatiques.

L'équipe LIG a utilisé le modèle Fuse pour sa soumission officielle pour le round 3 et a obtenu avec celui-ci la cinquième place sur les tâches audio et la première place sur la tâche multimodale (recherche combinée audio et image de segments vidéo), se qualifiant ainsi pour la finale du Star Challenge à Singapour.

6.4. *Recherche visuelle, évaluation sur la collection « round 2 » du Star Challenge*

La tâche de recherche par le contenu visuel VT1 était de trouver des images (en pratique des images clés extraites de vidéos) qui contenaient un concept donné (parmi les 20 pour lesquels le système a été entraîné) et qui étaient visuellement semblable à une image donnée. Nous avons constaté que les meilleurs résultats ont été obtenus par en pondérant dans un rapport 2 à 1 la similarité visuelle normalisée et la probabilité de présence du concept normalisée.

La tâche de recherche par le contenu visuel VT2 tâche était de trouver des plans vidéo qui contenaient un concept donné (parmi les 10 pour lesquels le système a été entraîné) et qui sont visuellement similaires à un plan vidéo donné. Nous avons encore constaté que les meilleurs résultats ont été obtenus en pondérant dans un rapport 2 à 1 la similarité visuelle normalisée et la probabilité de présence du concept normalisée.

Cette approche nous a classés cinquièmes sur les tâches VT1 et VT2.

6.5. *Recherche multimodale, évaluation sur la collection « round 3 » du Star Challenge*

La tâche de recherche multimodale AV1 (resp. AV2) était de trouver des plans vidéos qui était visuellement similaires à une image donnée et qui contenaient une requête audio définie comme une chaîne en API (resp. comme un énoncé vocal). Nous avons constaté que les meilleurs résultats ont été obtenus par la en pondérant dans un rapport 3 à 7 la similarité visuelle normalisée et le score de détection de chaînes API normalisé.

Cette approche nous a classés premiers sur les tâches multimodales AV1 et AV2.

7. Conclusion

Nous avons présenté une approche fondée sur l'utilisation de l'Alphabet Phonétique International (API) pour la recherche selon le contenu de vidéos multilingues. Une telle approche peut fonctionner même si les langues parlées dans les documents sont inconnues. Notre technique a été validée dans le contexte du « Star Challenge », une compétition de recherche d'information organisée par l'agence Singapourienne A-STAR. L'approche présentée inclut la construction d'un modèle acoustique multilingue à large couverture, contenant des unités API, et sur une méthode de recherche fondée sur la programmation dynamique. La programmation dynamique permet de repérer la chaîne de la requête dans la chaîne du document, même avec un taux d'erreur de transcription au niveau phonétique significatif. Les méthodes que nous avons développées nous ont classés premiers et troisièmes sur les tâches de recherche monolingues (anglais), cinquièmes sur la tâche de recherche multilingue et premiers sur la tâche de recherche multimodale (audio et image).

Les résultats obtenus montrent le potentiel d'une telle approche fondée sur l'API pour indexer et retrouver des documents audiovisuels dans une langue inconnue. Des expériences complémentaires seraient nécessaires sur de plus grands corpus pour confirmer cette tendance. Des améliorations seraient par ailleurs possibles au niveau de la qualité des modèles multilingues et de la recherche fondée sur l'alignement dynamique qui pourrait être améliorée en exploitant des graphes d'hypothèses (treillis) en sortie du système de décodage phonétique.

.

Enfin, la combinaison de l'audio (API), de l'indexation par concepts et de la similarité visuelle, s'est avérée efficace pour la tâche de recherche d'information selon le contenu de vidéos multimodales.

Remerciements

Ce travail a été en partie soutenu par le programme Quaero.

8. Bibliographie

[AYA 07] AYACHE S., QUÉNOT G., « Image and video indexing using networks of operators », *J. Image Video Process.*, vol. 2007, n° 4, 2007, p. 1–13, Hindawi Publishing Corp.

[CCC 05] CCC, « http ://www.dear.com/CCC/resources.htm », 2005.

[CLA 97] CLARKSON P., ROSENFELD R., « Statistical Language Modeling using the CMU-Cambridge Toolkit », *Eurospeech'07*, 1997, p. 2707–2710.

[GAU 82] GAUVAIN J.-L., MARIANI J.-J., « A method for connected word recognition and word spotting on a microprocessor », *Proc. IEEE ICASSP 82*, vol. 2, 3-5 May 1982, p. 891–894.

[LDC 93] LDC, « http ://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp ?catalogId=LDC93S6B », 1993.

[LDC 97] LDC, « http ://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp ?catalogId=LDC98S71 », 1997.

[LE 04] LE V.-B., DO-DAT T., CASTELI E., BESACIER L., SERIGNAT J.-F., « Spoken and written language resources for Vietnamese », *LREC'04*, 2004, p. 599–602.

[LOW 04] LOWE D., « Distinctive image features from scale-invariant keypoints », *International Journal of Computer Vision*, vol. 60, 2004, p. 91-110.

[M¨ 00] MÄENPÄÄ TOPI PIETIKÄINEN MATTI O. T., « Texture classification by multi-predicate local binary pattern operators », *15th International Conference on Pattern Recognition*, vol. 3, 2000, p. 951-95.

[MOR 04] MORARU D., BESACIER L., MEIGNIER S., FREDOUILLE C., BONASTRE J.-F., « Speaker Diarization in the ELISA Consortium over the last 4 years », *RT2004 Fall Workshop*, 13-14 Nov. 2004.

[PLA 97] PLACEWAY P., CHEN S., ESKENAZI M., JAIN U., PARIKH V., RAJ B., RAVISHANKAR M., ROSENFELD R., SEYMORE K., SIEGLER M., STERN R., THAYER, « The 1996 Hub-4 Sphinx-3 System », *In DARPA Speech Recognition Workshop*, Chantilly, VA, February 1997.

[STO 02] STOLCKE A., « SRILM – an extensible language modeling toolkit », *Intl. Conf. on Spoken Language Processing*, 2002.

Utilisation de concepts visuels et de la diversité visuelle pour améliorer la recherche d'images

Sabrina Tollari Marcin Detyniecki, Ali Fakeri-Tabrizi, Christophe Marsala, Massih-Reza Amini, Patrick Gallinari

Université Pierre et Marie Curie - Paris 6, UMR CNRS 7606 - LIP6
104 avenue du président Kennedy, 75016 Paris, France, prénom.nom@lip6.fr

RÉSUMÉ. *Dans cet article, nous étudions (i) comment extraire et exploiter des concepts visuels pour améliorer la recherche d'images basée sur le texte, et (ii) comment diversifier les résultats pertinents obtenus. Nous utilisons d'abord des forêts d'arbre de décisions flous (FFDTs) pour détecter les concepts dans les images, puis nous découvrons à l'aide de l'analyse des cooccurrences des relations d'exclusion mutuelle et d'implication entre les concepts. Ensuite, nous utilisons ces concepts pour améliorer la pertinence des résultats obtenus par un système de recherche d'images par le texte. Enfin, nous appliquons une méthode de diversité visuelle basée sur le partitionnement de l'espace visuel. Ce travail se place dans le cadre de la campagne d'évaluation CLEF. Il montre une nette amélioration des résultats lorsque l'on utilise les concepts apparaissant explicitement dans la requête textuelle, ainsi que l'efficacité du clustering spatial.*

ABSTRACT. *In this article, we study (i) how to automatically extract and exploit visual concepts and (ii) fast visual diversity. First, in the Visual Concept Detection Task (VCDT), we look at the mutual exclusion and implication relations between VCDT concepts in order to improve the automatic image annotation by Forest of Fuzzy Decision Trees (FFDTs). Second, in the ImageCLEFphoto task, we use the FFDTs learnt in VCDT task and WordNet to improve image retrieval. Third, we apply a fast visual diversity method based on space clustering to improve the cluster recall score. This study shows that there is a clear improvement, in terms of precision or cluster recall at 20, when using the visual concepts explicitly appearing in the query and that space clustering can be efficiently used to improve cluster recall.*

MOTS-CLÉS : *recherche d'images basée sur le texte, détection de concepts visuels, arbre de décisions flous, diversification*

KEYWORDS: *text-based images retrieval, visual concepts detection, Fuzzy Decision Trees*

1. Introduction

Les moteurs de recherches d'images sur le web utilisent principalement des informations textuelles, telles que le titre de la page web, le nom de l'image, le texte adjacent, pour tenter de "comprendre" le sens de l'image. Cependant, le texte d'une page web n'est pas toujours en rapport avec le contenu visuel de l'image. De plus, l'utilisateur préfèrant souvent exprimer son besoin d'information à l'aide de quelques mots-clés, il est difficile de trouver des liens avec l'information visuelle contenue dans les images. Il est donc intéressant de trouver des méthodes qui permettent de vérifier l'adéquation visuelle de l'image avec le texte de la requête posée par l'utilisateur. Dans (Yavlinsky *et al.*, 2006), des concepts visuels sont utilisés pour raffiner visuellement les résultats obtenus, cependant, l'utilisateur doit choisir manuellement le concept visuel à appliquer. Nous proposons dans cet article d'étudier une méthode qui permet de choisir automatiquement le concept visuel à appliquer.

D'un autre côté, quand un utilisateur pose une requête, ce qui l'intéresse c'est d'avoir des documents qui soient, certes tous pertinents, mais aussi qui soient les plus dissimilaires les uns des autres (Song *et al.*, 2006, Chen *et al.*, 2006, Zhai *et al.*, 2003). Par exemple, si l'utilisateur cherche des photographies d'animaux en train de nager pour illustrer un article sur la faculté de nager des animaux, toutes les images d'animaux en train de nager seront certes pertinentes, mais afin qu'il ait un aperçu, dès le début de sa recherche, de toute la diversité des images pertinentes, il serait intéressant de lui fournir dans les premiers résultats, des images qui contiennent toutes des animaux différents. Cette diversification des résultats selon le critère *animal* peut permettre à l'utilisateur de trouver plus rapidement ce qu'il recherche. Dans cet article, nous proposons une méthode de diversification basée sur les informations visuelles des images, et nous la comparons avec les résultats obtenus par une diversification aléatoire.

Notre travail se place dans le cadre de deux tâches de la campagne internationale CLEF 2008. La première tâche appelée *Visual Concept Detection Task (VCDT)* (Deselaers *et al.*, 2008) est une tâche de détection de concepts visuels. Lors de cette campagne, notre système de détection de concepts visuels est arrivé 3ième sur 11 équipes. La deuxième appelée ImageCLEFphoto (Arni *et al.*, 2008) est une tâche de recherche d'images basée sur les informations textuelles et visuelles, et propose d'étudier les problèmes soulevés par la diversité.

Dans la première partie, nous détaillons la méthode de détection des concepts visuels proposée, puis nous utilisons l'analyse des cooccurrences pour détecter des relations d'exclusion et d'implication, et nous discutons les résultats obtenus par notre méthode dans la tâche VCDT. Dans la deuxième partie, nous expliquons comment utiliser les concepts visuels de VCDT pour améliorer la recherche d'images basée sur le texte, puis nous présentons notre méthode de diversité visuelle basée sur le partitionnement de l'espace et enfin nous discutons les résultats obtenus. Dans la dernière partie, nous concluons.

2. Détection de concepts visuels

2.1. *Détection de concepts visuels à l'aide de forêts d'arbres de décision*

L'annotation automatique d'images est un problème typique d'apprentissage automatique inductif. Une des méthodes classiques dans ce domaine utilise les arbres de décisions (*Decision Trees (DT)*). Cependant, les arbres de décisions classiques rencontrent des difficultés pour traiter des données numériques ou imprécises. L'introduction de la logique floue a permis de réduire ces difficultés. L'apprentissage inductif consiste à passer du spécifique vers le général. Un arbre est construit, de la racine vers les feuilles, par partitionnements successifs de l'ensemble d'apprentissage en sous-ensembles. Chaque partition est réalisée au moyen d'un test sur un des attributs et amène à la définition d'un noeud de l'arbre (Marsala *et al.*, 1997). (Marsala *et al.*, 2006) montre que quand on considère de grands ensembles (en terme de dimension et de taille) de données non-équilibrés, il est intéressant de combiner plusieurs arbres de décisions, et ainsi d'obtenir une forêt d'arbres de décision (*Forest of Fuzzy Decision Trees (FFDT)*). De plus, la combinaison des résultats de plusieurs arbres de décision permet d'obtenir un degré de confiance dans la classification.

Durant la phase d'apprentissage, une forêt de n arbres est apprise pour chaque concept. Chaque arbre F_j de la forêt est construit en utilisant un sous-ensemble d'apprentissage T_j. Chaque sous-ensemble est un ensemble équilibré constitué d'images de l'ensemble d'apprentissage choisies aléatoirement.

Durant la phase de classification, chaque image I est classée par chaque arbre de la forêt. Le degré $d_j \in [0, 1]$ obtenu pour l'image I représente la présence du concept C pour l'arbre F_j de la forêt. Ainsi, pour chaque image I, n degrés $d_j,\ j = 1 \ldots n$ sont obtenus. Puis, tous les degrés sont agrégés par un vote : $d = \sum_{j=1}^{n} d_j$. Finalement, pour décider si une image contient un concept ou non, nous utilisons un seuil t tel que $t \leq n$: l'image I contient le concept C si $d \geq t$.

2.2. *Analyse des cooccurrences*

Les arbres de décisions apprennent chaque concept de manière indépendante. Cependant, les concepts sont reliés entre eux. Par exemple, une scène ne peut pas être simultanément à l'intérieur (*indoor*) et à l'extérieur (*outdoor*) ; si l'on observe qu'il y a des nuages (*cloudy*), on peut en déduire qu'il y le concept ciel (*sky*). Dans cette partie, nous proposons d'utiliser l'analyse des cooccurences pour déterminer automatiquement les relations entre les concepts. Une fois que nous avons découvert une relation, nous avons besoin d'une règle pour résoudre les conflits d'annotation. Cette règle doit prendre en compte les degrés de confiance donnés par les FFDTs. Par exemple, chaque image sera annotée par *outdoor* avec un certain degré et par *indoor* avec un autre degré. Cependant, les concepts *outdoor* et *indoor* ne peuvent apparaître simultanément. Pour trouver les règles à appliquer, nous étudions deux types de relations entre les concepts : les exclusions et les implications.

Exclusions Pour découvrir automatiquement les *exclusions* entre concepts, nous étudions les concepts qui n'apparaissent jamais ensemble. Pour cela, nous calculons la matrice de cooccurrences COOC entre les concepts. Comme il peut y avoir du bruit (erreurs d'annotation), nous utilisons un seuil α pour décider quels couples de concepts n'apparaissent jamais ensemble. Quand nous savons quels concepts sont reliés, nous appliquons une règle de résolution en fonction des degrés de confiance fournis par les FFDTs. Nous avons choisi une règle qui, pour les concepts mutuellement exclusifs, éliminent (c'est-à-dire donnent un degré de confiance de zéros) aux étiquettes ayant le plus faible degré de confiance. Par exemple, si *outdoor* a un degré de confiance de 42/50 et *indoor* a un degré de 20/50, alors le degré de confiance de *indoor* sera mis à zéro. Pour chaque image de test, soit d(I,C) le degré de l'image I pour le concept C, nous appliquons l'algorithme suivant :

Pour chaque couple (A,B) tel que $COOC(A, B) \leq \alpha$ (*découverte*)

Si d(I,A) $<$ d(I,B) alors d(I,A)=0 sinon d(I,B)=0 (*règle de résolution*)

où COOC est la matrice de cooccurrences des concepts.

Implications Pour découvrir les *implications*, nous étudions, par définition de l'implication, les cooccurrences entre l'absence d'un concept et la présence d'un autre concept. La matrice de cooccurrences résultante COOCNEG est asymétrique, ce qui reflète le fait qu'un concept implique un autre concept, mais cela n'est pas réciproque. La règle de résolution utilisée suppose que si un concept implique un autre concept, alors le degré de confiance de ce dernier doit être au moins égal au premier. Comme il peut y avoir du bruit, nous utilisons un seuil β pour décider quel concept implique un autre concept.

Pour chaque image de test I, soit d(I,C) le degré de l'image I pour le concept C, nous appliquons l'algorithme suivant :

Pour chaque couple (A,B) tel que $COOCNEG(A, B) \leq \beta$ (*découverte*)

d(I,B)=max(d(I,A),d(I,B)) (*règle de résolution*)

où COOCNEG est la matrice asymétrique de cooccurrences entre un concept et la négation d'un autre concept.

2.3. *Expérimentations et résultats*

2.3.1. *Corpus*

Nous appliquons notre méthode de détections de concepts visuels au corpus de la tâche *Visual Concept Detection Task (VCDT)* (Deselaers *et al.*, 2008) de la campagne internationale d'évaluation CLEF 2008. Cette tâche correspond à un problème de classification multi-classes multi-étiquettes. Le corpus de VCDT contient 1827 images d'apprentissage et 1000 images de test. Il y a 17 concepts visuels. Une image d'apprentissage est annotée en moyenne par 5.4 concepts (entre 0 (2 images) et 11 concepts

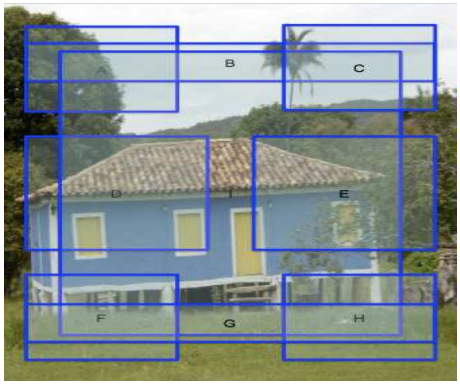


Figure 1. *Les images sont segmentées en 9 régions*

par image). Un concept annote en moyenne 584 images d'apprentissage (entre 68 et 1607 images d'apprentissage par concept).

2.3.2. *Descripteurs visuels*

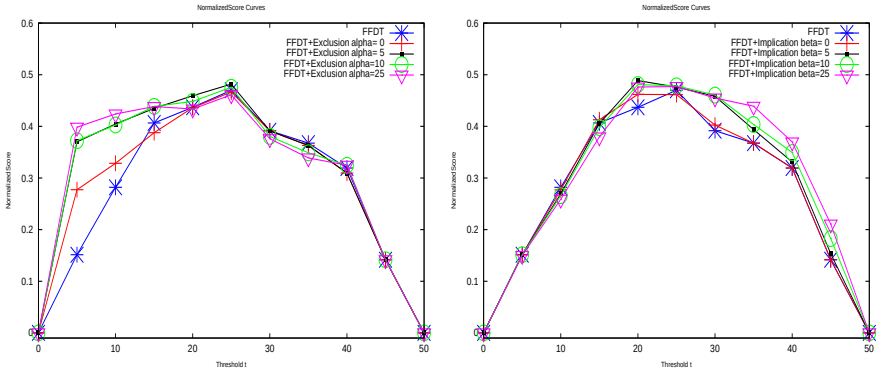
Les descripteurs visuels utilisés sont exclusivement basés sur la couleur. Afin d'obtenir une information liée à la disposition spatiale des objets dans les images, nous segmentons les images en 9 régions qui se chevauchent (voir figure 1). Pour chaque région, nous calculons un histogramme HSV. Le nombre de dimensions de l'histogramme reflète l'importance de la région. La région centrale représente le thème de l'image. La région en haut et celle du bas sont intéressantes pour les concepts visuels généraux, tels que le ciel, le soleil, la végétation, la mer... Les autres régions sont décrites en termes de différences de couleurs entre la gauche et la droite. L'idée est de rendre explicite les symétries. En effet, les objets peuvent apparaître d'un côté ou de l'autre de l'image. Or étant donné que les arbres de décisions ne sont pas capables de découvrir automatiquement ce genre de relations, l'utilisation de ces différences permet de leur donner la possibilité de tenir compte de ces symétries. Au final, chaque image est représentée par un vecteur de valeurs numériques.

2.3.3. *Mesure de performances*

Pour mesurer les performances de notre système de détection de concepts visuels, nous avons choisi d'utiliser le *Normalized Score (NS)* qui a déjà été utilisé par de nombreux modèles d'annotation automatique (Barnard *et al.*, 2003). Pour un concept C donné, le *Normalized Score (NS)* peut être défini ainsi :

$$NS = \frac{r}{w} - \frac{w}{N-n}$$

où N est le nombre d'images de l'ensemble de test, n est le nombre d'images de l'ensemble de test initialement annotées par le concept C, r est le nombre d'images



(a) α (b) β

Figure 2. *Influence des paramètres α et β sur le* Normalized Score (NS) *en fonction du seuil t de décision (les degrés de confiance fournis par les arbres varient entre 0 et 50)*

annotées par le système avec le concept C et qui étaient initialement annotées par ce concept, et w est le nombre d'images annotées par le système avec le concept C et qui n'étaient pas initialement annotées par ce concept C. Ce score correspond donc à la somme de la sensibilité et de la spécificité moins 1. Il varie entre -1 et 1. Le score vaut 1 quand le système ne commet aucune erreur, -1 quand toutes les images sont mal annotées, 0 quand toutes les images sont annotées par le concept C.

2.3.4. *Utilisation des relations d'exclusion et d'implication*

Une étape préliminaire avant d'extraire des concepts visuels est d'étudier les valeurs de cooccurrence entre les concepts pour découvrir les relations d'exclusion et d'implication. Pour les 17 concepts, il y a 136 valeurs de cooccurence. Ces valeurs varient de 0 (non-cooccurrences) à 1443 (sur les 1827 images d'apprentissage). Comme il peut y avoir du bruit, nous avions fixé, lors de notre participation à la tâche VCDT, le seuil α à la valeur 5. Ce seuil avait été déterminé grâce à la distribution des valeurs de cooccurrence de l'ensemble d'apprentissage. Si $\alpha = 5$, cela signifie que deux concepts sont considérés exclusifs si moins de 5 images sont annotées par les deux concepts. La figure 2(a) montre que cette valeur du seuil maximise en effet les résultats pour $t = 25$, mais que pour un seuil de décision $t \leq 15$, il peut être intéressant de prendre une valeur de α plus grande. De même, nous avions fixé $\beta = 5$ (un concept implique un autre concept si au maximum 5 images d'apprentissage ne sont pas annotées par le premier concept, et en même temps annotées par le second concept). La figure 2(b) confirme que les meilleurs scores à $t = 25$ sont obtenus pour $\beta = 5$, mais que pour $t \geq 30$, il peut être intéressant de prendre une valeur de β plus grande. Enfin, ces deux figures montrent que prendre $\alpha = 0$ ou $\beta = 0$ ne donne jamais de meilleurs résultats

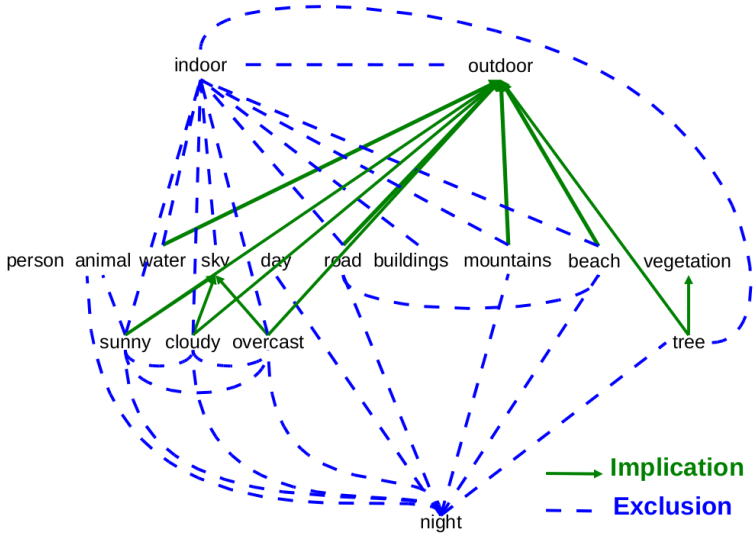


Figure 3. *Schéma montrant les relations d'exclusion et d'implication entre les concepts automatiquement découvertes*

que pour $\alpha = 5$ ou $\beta = 5$, et que donc prendre en compte les erreurs d'annotations permet d'améliorer sensiblement les résultats.

Pour $\alpha = 5$ et $\beta = 5$, notre système a automatiquement découvert 25 relations d'exclusion et 12 relations d'implication (voir figure 3). Nous avons trouvé toutes les relations évidentes (par exemple, les concepts *indoor* et *outdoor* sont exclusifs ; un arbre implique de la végétation), ainsi que quelques relations moins triviales. Par exemple, les concepts *sunny* et *animal* sont trouvés exclusifs. Cette relation peut être expliquée par le fait que, lorsqu'une personne annote une image, son attention peut se focaliser sur un objet (ou un animal) et ne pas préter attention au fait que le ciel soit ensoleillé, cette information étant jugée secondaire ou inintéressante.

Finalement, la figure 4 compare les scores NS obtenus par les FFDTs seuls ou avec les régles d'implication et d'exclusion. Nous notons que les meilleurs résultats sont obtenus pour $t = 20$ par l'application des règles d'implication aux degrés de confiance des FFDT. Pour un seuil de décision $t \leq 15$, il vaut mieux utiliser les exclusions ; pour $t \geq 30$, il vaut mieux utiliser les implications. Finalement, la méthode FFDT+Implication+Exclusion donne globalement les meilleurs résultats. Cependant, nous remarquons que pour $t = 25$, toutes les méthodes donnent quasiment les mêmes résultats. Pour ce seuil que nous utiliserions classiquement pour prendre une décision, l'intérêt d'utiliser les règles d'implication et d'exclusion n'est donc pas totalement concluante.

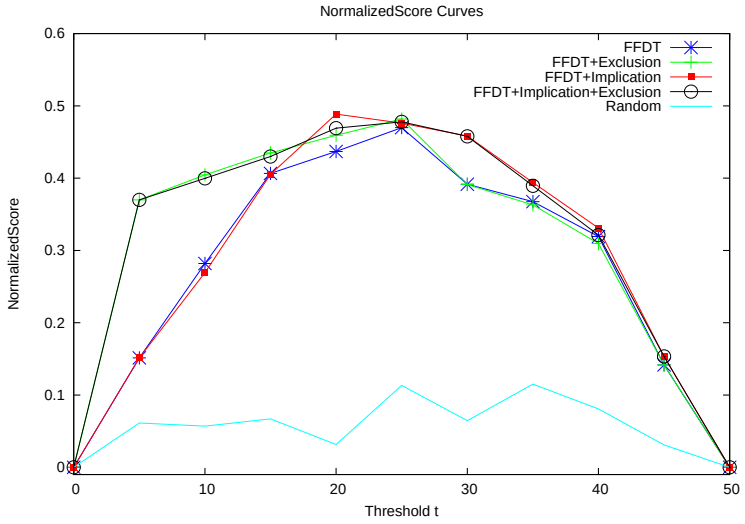


Figure 4. *Courbes de scores NS en fonction du seuil de décision (les degrés de confiance fournis par les arbres varient entre 0 et 50) avec* $\alpha = 5$ *et* $\beta = 5$

	EER	Gain	AUC	Gain
Moyenne des 53 runs	33.92	-	63.64	-
Classifieurs aléatoires	50.17	-48%	49.68	-22%
FFDT	24.55	+28%	82.74	+30%

Tableau 1. *Résultats de la tâche VCDT 2008 (EER : Equal Error Rate - AUC : Area under ROC curve). La moyenne des 53 runs correspond à la moyenne des scores EER et AUC des 53 runs soumis par les 11 équipes participantes à la tâche VCDT en 2008.*

Le tableau 1 montre les résultats obtenus lors de la compagne d'évaluation ImageCLEF en 2008. Notre méthode basée sur les FFDTs est arrivée quatrième sur 53 résultats soumis (troisème équipe sur 11 équipes internationales).

3. Recherche d'images

3.1. *Utilisation de concepts visuels pour améliorer la recherche d'images basée sur le texte*

De nombreux travaux (Barnard *et al.*, 2003, Datta *et al.*, 2008, Tollari *et al.*, 2007) montrent que combiner des informations visuelles et textuelles améliorent la recherche

d'images, mais la plupart de ces travaux se concentrent sur la fusion précoce ou tardive de ces informations ou sur l'annotation des images. Nous proposons d'utiliser les concepts visuels appris par les FFDTs pour améliorer la recherche d'images basée uniquement sur le texte. La difficulté est de déterminer comment utiliser les concepts visuels dans le cas où les seules informations que l'on peut utiliser sont le nom du concept, les descripteurs visuels, et la requête composée de quelques mots-clés.

A l'aide des FFDTs apprises pour chaque concept (voir partie 2) et des descripteurs visuels de chaque image, nous pouvons donner un degré de confiance qu'un certain concept apparaisse dans une nouvelle image. Il reste donc à trouver un moyen de faire la correspondance entre la requête et le (ou les) concept(s) que l'on veut détecter dans les images.

Premièrement, si le nom du concept apparaît directement dans les mots de la requête (méthode DIRECTE), nous proposons de filtrer les images ordonnées par la recherche textuelle en fonction du degré qu'elles obtiennent pour ce concept.

Deuxièmement, si le nom du concept apparaît dans les mots de la requête ou dans une liste de synonymes des mots de la requête donnés par WordNet (Fellbaum, 1998) (méthode WN), nous proposons également de filtrer les images ordonnées par la recherche textuelle en fonction du degré qu'elles obtiennent pour ce concept. Par exemple, la requête 5 d'ImageCLEFphoto 2008 est «*animal swimming*». En utilisant la méthode DIRECTE, le système détermine automatiquement qu'il doit utiliser la FFDT du concept *animal*. Si, de plus nous utilisons WordNet (méthode WN), le système détermine automatiquement qu'il doit utiliser les FFDTs des concepts *animal* et *water* (car d'après WordNet, un synonyme pour *swimming* est «*water sport, aquatics*».

Pour chaque requête, nous déterminons la liste ordonnée des images pertinentes selon le modèle de langue (LM) ou selon le modèle TD-IDF utilisé sur le texte. Puis, à l'aide des FFDTs, nous réordonnons les 50 premières images de chaque requête ainsi : le système parcourt les images retrouvées du rang 1 au rang 50. Si le degré d'une image est inférieur à un seuil t, alors cette image est réordonnée à la fin de la liste des 50 premières images. De cette façon, les images pertinentes se trouvent toujours dans les 50 premiers résultats.

3.2. *Promouvoir la diversité en utilisant le clustering spatial*

Pour une requête donnée, les documents similaires sont naturellement ordonnées à des rangs proches. Quand un utilisateur pose une requête, ce qui l'intéresse c'est d'avoir des documents qui soient certes tous pertinents, mais aussi qui soient les plus dissimilaires les uns des autres.

Les techniques de *clustering* visuel sont étudiées depuis de nombreuses années. Deux approches sont généralement proposées : le partitionnement des données et le partitionnement de l'espace. La première approche nécessite beaucoup de temps de calcul et doit être adaptée à la distribution des premières images résultats d'une re-

quête donnée comme dans (Inoue *et al.*, 2008). La seconde approche, comme elle est faite indépendamment des données, est souvent moins efficace, mais peut-être appliquée de manière très rapide. Nous avons choisi de réaliser un partitionnement de l'espace visuel fondé sur l'histogramme Hue de l'espace HSV. Pour chaque image, nous binarisons l'histogramme Hue. Chaque vecteur binaire correspond à un cluster. En fonction du nombre de dimensions nh de l'histogramme, nous obtiendrons 2^{nh} clusters possibles (les clusters ne seront pas forcement tous instantiés, car certains pourront correspondre à aucune donnée).

Pour chaque requête, les images sont classées en deux listes. Le système parcours les images dans l'ordre : des plus pertinentes vers les moins pertinentes. Si une image appartient à aucun des clusters des images plus pertinentes qu'elle, alors cette image est ajoutée à la fin de la première liste. Si une image appartient au même cluster qu'une image plus pertinente qu'elle, alors cette image est ajoutée à la fin de la deuxième liste. Au final, nous obtenons dans la première liste uniquement des images avec des clusters différents. Nous concaténons ensuite la première liste et la deuxième liste. L'image de rang 1 est toujours au rang 1 ; l'image de rang 2 se retrouve soit au rang 2 si son cluster est différent du cluster de l'image de rang 1, soit à la position $nbcv + 1$ si son cluster est identique (avec $nbcv$ le nombre de clusters visuels), et ainsi de suite. Dans la pratique, comme nous nous intéressons seulement aux 20 premiers documents pertinents, il suffit de s'arrêter de parcourir les images lorsque nous avons trouvé 20 images dans 20 clusters différents. Plus nous aurons de clusters, et moins il y aura de changement dans l'ordre des images. Nous appelons cette méthode : DIVVISU.

Pour avoir un point de comparaison, nous proposons également une méthode de diversification "naïve" qui consiste à permuter aléatoirement les a premiers résultats. Nous appelons cette méthode : DIVALEA.

3.3. *Expérimentations et résultats*

3.3.1. *Corpus*

Nous utilisons le corpus la tâche ImageCLEFphoto (Arni *et al.*, 2008) de la campagne dévaluation CLEF 2008. Ce corpus contient 20k images généralistes et 39 *topics*. Chaque *topic* est composé d'un titre, d'une partie narrative, de 3 images correspondant au *topic*, ainsi que d'un élément indiquant sur quel critère doit être appliqué la diversité (<CLUSTER>). Par exemple, le premier *topic* est :

```
<TITLE>church with more than two towers</TITLE>
<CLUSTER>city</CLUSTER>
<NARR>Relevant images will show a church, cathedral or a mosque with
three or more towers. Churches with only one or two towers are not
relevant. Buildings that are not churches, cathedrals or mosques are
not relevant even if they have more than two towers.</NARR>
```

Les 39 requêtes doivent être dérivées de chacun des *topics*. Il y a 17 critères (parfois appelés sous-thèmes (*subtopics*) (Zhai *et al.*, 2003)) de diversités différents : *animal*,

bird, city, city/nationalpark, composition, country, group composition, landmark location, sport, state, statue, tourist attraction, vehicle type, venue, volcano, weather condition. La plupart de ces critères correspondent à des lieux. Par exemple, pour la première requête, le critère de diversité est *city*. Pour cette requête, ce critère contient 5 clusters ($n_c = 5$) : *Moscow, Saint Petersburg, Melbourne, Sydney, Bolshaya Reka*. Chaque image du corpus est associée à une légende contenant le titre de l'image, sa date de création, sa localisation, le nom du photographe, une description sémantique du contenu de l'image (déterminée par le photographe) ainsi que de notes additionnelles.

3.3.2. *Mesures de performances*

Les mesures classiquement utilisées en recherche d'information sont généralement la précision et le rappel. De plus, afin de combiner ces deux mesures, la F1-mesure est généralement utilisée. Pour ImageCLEFphoto 2008, le but est de retrouver non seulement les documents pertinents, mais aussi de retrouver, dans les premiers résultats, les documents pertinents qui sont les plus différents les uns des autres en fonction du critère de diversité choisi. C'est pourquoi les mesures de performances utilisées sont : la précision à 20 (P20), le *cluster recall* à 20 (CR20) (Zhai *et al.*, 2003) et la F1-mesure appliquée au P20 et au CR20. Soit $nbpr(n)$ le nombre de documents pertinents retrouvés parmi les n premiers documents retrouvés, la précision à 20 peut être définie ainsi :

$$P20 = \frac{nbpr(20)}{20}.$$

Le *cluster recall* à 20 (CR20) (appelé aussi *S-recall*) (Zhai *et al.*, 2003) a pour but de mesurer le nombre de clusters différents présents dans les 20 premiers résultats. Soit n_c le nombre de clusters différents pour une requête donnée, soit $nbcpr(n)$ le nombre de clusters différents couverts par les documents pertinents retrouvés parmi les n premiers documents retrouvés pour cette requête, alors le CR20 est définie ainsi :

$$CR20 = \frac{nbcpr(20)}{n_c}.$$

La dernière mesure utilisée est la F1-mesure définit ainsi dans notre cas :

$$F1 - mesure = 2 \times \frac{P20 \times CR20}{P20 + CR20}.$$

3.3.3. *Correspondance directe et par WordNet*

Nous réalisons d'abord une recherche d'images basée uniquement sur le texte. Pour cela, nous construisons les requêtes en utilisant les éléments du titre ainsi que les phrases de la balise `<NARR>` qui ne contiennent pas le mot *not*. Pour décrire chaque image, nous prenons en compte le texte contenu dans tous les éléments de la légende. Nous appliquons ensuite un modèle classique de langue (LM) ainsi que le modèle TF-IDF.

Pour déterminer si une image contient un concept visuel, nous choisissons de fixer le seuil t à la médianne de tous les degrés obtenus par un concept donné (cette valeur

Tableau 2. *Comparaison des méthodes DIRECTE et WN. Par la méthode DIRECTE, seulement 11 requêtes sont modifiées. Par la méthode WN, 25 requêtes sont modifiées*

		Moyenne sur 39 requêtes		Moyenne des requêtes modifiées		
Texte	Méthode	P20 (gain %)	CR20 (gain %)	Nb topics	P20 (gain %)	CR20 (gain %)
LM	-	0.185(-)	0.247(-)	11	0.041(-)	0.090(-)
				25	0.148 (-)	0.254 (-)
	DIRECTE	0.195(+6)	0.257(+4)	11	0.077(+88)	0.126(+40)
	WN	0.176(-5)	0.248(+1)	25	0.134(-9)	0.257(+1)
TF-IDF	-	0.250(-)	0.300(-)	11	0.155(-)	0.161(-)
				25	0.210(-)	0.305(-)
	DIRECTE	0.269(+8)	0.313(+5)	11	0.223(+44)	0.209(+30)
	WN	0.260(+4)	0.293(-2)	25	0.226(+8)	0.294(-4)

varie de 7.3 (*overcast*) à 28.8 (*outdoor*). Nous n'avons pas utiliser dans cette partie les règles d'exclusion et d'implication.

Le tableau 2 montre que, en moyenne sur tous les topics, la méthode DIRECTE améliore la précision à 20 documents (P20) de +8% par rapport au TF-IDF et de +6% par rapport au LM, tandis que la méthode WN améliore le P20 du TF-IDF de +4%, mais diminue de -5% le P20 du LM. Comme les méthodes DITECTE et WN dépendent de la présence du nom du concept dans la requête textuelle et que certaines requêtes ne contiennent aucun des noms des 17 concepts, les résultats de certaines requêtes ne sont pas modifiés. La méthode DIRECTE modifie seulement 11 requêtes, tandis que la méthode WN modifie 25 requêtes. C'est pourquoi nous séparons, dans le tableau 2, les résultats en 3 groupes. Nous remarquons une amélioration des scores de P20 de +44% par rapport au TF-IDF (+30% par rapport au LM) pour les 11 requêtes modifiés par la méthode DIRECTE, mais une amélioration de seulement +8% et une diminution de -9% en utilisant WordNet. Nous en déduisons que la méthode DIRECTE permet d'améliorer sensiblement les résultats obtenus par le texte seul, mais que par contre l'utilisation de WordNet n'est pas adapté pour ce genre de tâche. Par exemple, d'après WordNet, le concept visuel *person* n'est pas dans la liste des synonymes du mot *people*. Nous ne pouvons donc trouver de lien entre les requêtes recherchant des personnes et le concept *person*. Nous avons également étudié d'autres types de relations (hyponymie, hypernymie...), mais les résultats obtenus étaient globalement inférieurs à ceux obtenus par synonymie.

3.3.4. *Diversification*

Notre méthode de diversification est basée sur un partitionnement de l'espace visuel, et non pas sur des informations sémantiques qui pourraient être utiles comme, par exemple, une liste de villes pour le critère *city*. Nous savons que nos résultats seront sous-optimaux, car la diversité visuelle n'entraîne pas forcement la diversité

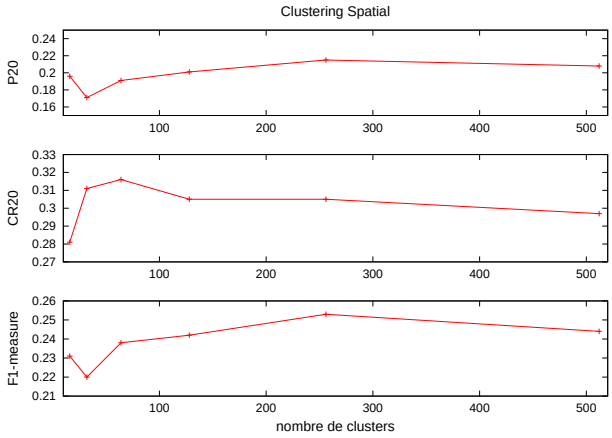


Figure 5. *Influence du nombre de clusters sur les scores P20, CR20 et F1-measure obtenus par diversification DIVVISU des résultats du modèle TF-IDF*

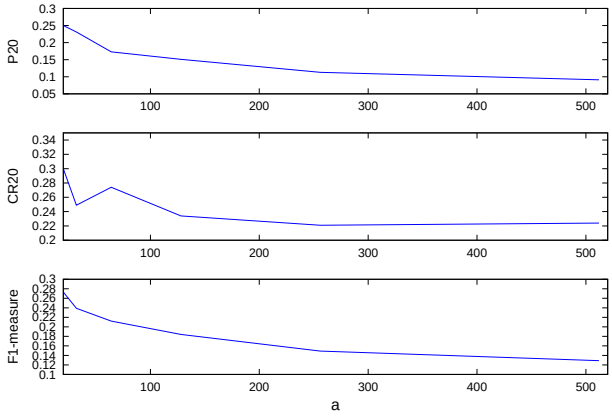


Figure 6. *Permutation aléatoire des a premiers documents obtenus par TF-IDF*

sémantique. Nous supposons cependant que des photographies de cathédrales prises dans une même ville seront assez visuellement similaires. Pour pouvoir étudier l'influence du nombre de clusters visuels, nous avons choisi de faire varier le nombre de dimensions de l'histogramme Hue de 4 à 9 dimensions. La quantité d'information initiale étant ainsi toujours la même. Nous obtenons un nombre théorique de clusters variant de 16 à 512, et un nombre de clusters instanciés légèrement inférieur. La figure 5 montre l'évolution des scores P20, CR20 et F1-measure en fonction du nombre de clusters de l'espace visuel. Pour un nombre de clusters égal à 16, la diversification

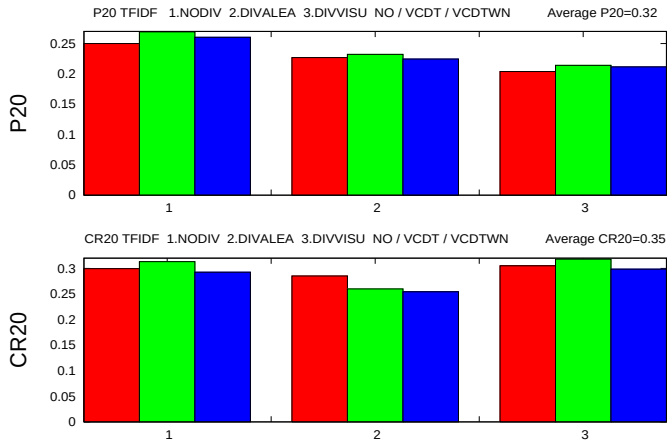


Figure 7. *Comparaison des méthodes de diversification (1. sans diversification, 2. diversification aléatoire (DIVALEA), 3. diversification par clustering spatial (DIVVISU avec $nh = 8$)). Pour chaque méthode de diversification, la première barre correspond au TF-IDF seul, la deuxième à TF-IDF+DIRECTE et la troisième à TF-IDF+WN*

n'est pas complètement effectuée, car certaines images ayant des clusters similaires aux images de rangs plus élevés se retrouvent toujours dans les 20 premiers résultats. Pour un nombre de clusters supérieur à 32, le P20 chute brutalement et puis remonte doucement vers le P20 du TF-IDF sans toutefois l'atteindre. A l'inverse, le CR20 augmente atteignant son maximum pour un nombre de clusters de 64, puis diminue pour atteindre le CR20 du TF-IDF. En moyenne, la meilleur valeur de F1-measure est obtenu pour un nombre de clusters de 256 (soit une dimension de l'espace de 8). Pour étudier la difficulté de garder des scores forts lorsque l'on effectue une diversification des résultats, la figure 6 montre la baisse des scores P20, CR20 et F1-mesure en fonction du nombre de documents permutés aléatoirement. La figure 7 compare les scores de diversification pour les méthodes DIVVISU (à 256 clusters) et DIVALEA (permutation aléatoire des 40 premiers documents) par rapport aux scores TF-IDF sans diversification. Nous remarquons que les deux méthodes proposées donnent des scores P20 largement inférieurs au P20 du TF-IDF, mais DIVALEA diminue le CR20, tandis que DIVVISU l'améliore légèrement (+2%). Nous en concluons que notre méthode DIVVISU améliore légèrement les résultats de diversification, mais que, comme de nombreuses autres méthodes (voir (Tollari *et al.*, 2008)), elle diminue le P20. La figure 8 montre les scores CR20 de TF-IDF versus TF-IDF+DIVVISU. Notre méthode DIVVISU, qui est basée uniquement sur de l'information visuelle, ne semble pas être adaptée spécialement pour un type de critère de diversification. En effet, il n'y a aucun critère de diversification qui donne de bons résultats seulement pour TF-IDF+DIVVISU.

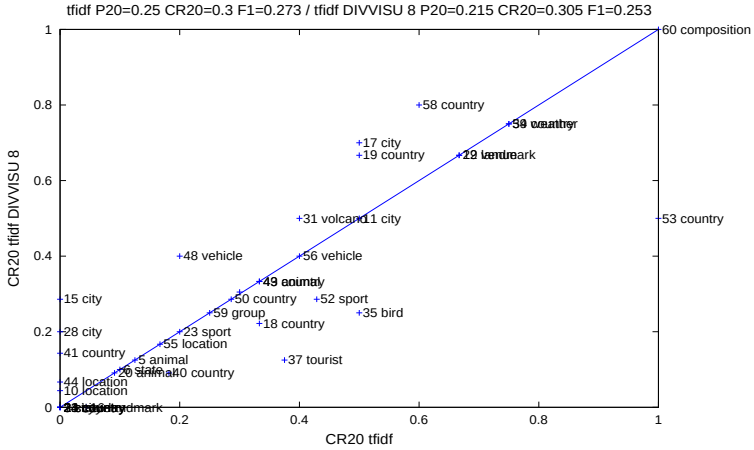


Figure 8. *Comparaison des scores CR20 du TF-IDF et du TF-IDF diversifié par DIV-VISU avec $nh = 8$ (256 clusters visuels). Chaque point correspond à une requête (le nombre correspond au numéro de la requête et le mot associé correspond au critère de diversification demandé dans le topic)*

4. Conclusion

Dans cet article, nous nous intéressons à deux difficultés. La première est l'exploitation de concepts visuels pour améliorer la recherche d'images basée uniquement sur le texte. Pour tenter de résoudre cette difficulté, nous utilisons des forêts d'arbres de décisions flous pour donner un degré de confiance que le concept soit dans une image, puis en fonction des termes de la requête, nous filtrons les images dont le degré de confiance correspond aux concepts visuels de la requête est trop faible. Les résultats montrent une nette amélioration des scores pour les requêtes qui contiennent explicitement le nom d'un concept. Nous en déduisons que la difficulté principale est de déterminer quel concept appliquer pour une requête qui ne contient pas explicitement de concept. La seconde est la diversification des résultats pertinents. Nous proposons d'utiliser le partitionnement de l'espace visuel afin d'obtenir très rapidement le cluster visuel d'une image, puis de garder dans les 20 premiers documents uniquement des images qui ont des clusters différents. Notre méthode augmente légèrement les scores de diversité, et a l'avantage de pouvoir être utilisée très simplement sans lourd calcul.

Dans nos futurs travaux, nous souhaitons améliorer nos règles de résolutions (exclusion et implication) pour obtenir de meilleurs résultats de classification, puis les utiliser dans la tâche de recherche d'images. En effet, nous avons fixé un seuil de décision t à la médiane de tous les degrés obtenus, or cette valeur varie de 7.3 à 28.8, l'utilisation de règles d'exclusion dans la tâche de recherche d'images devrait, d'après la figure 4, améliorer nos résultats.

Remerciements

Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-06-MDCA-002 (projet AVEIR).

5. Bibliographie

Arni T., Clough P., Sanderson M., Grubinger M., « Overview of the ImageCLEFphoto 2008 Photographic Retrieval Task », *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the Cross-Language Evaluation Forum*, Lecture Notes in Computer Science, Aarhus, Denmark, 2008.

Barnard K., Duygulu P., de Freitas N., Forsyth D., Blei D., Jordan M. I., « Matching Words and Pictures », *Journal of Machine Learning Research*, vol. 3, p. 1107-1135, 2003.

Chen H., Karger D. R., « Less is more : probabilistic models for retrieving fewer relevant documents », *ACM SIGIR*, p. 429-436, 2006.

Datta R., Joshi D., Li J., Wang J. Z., « Image retrieval : Ideas, influences, and trends of the new age », *ACM Computing Surveys*, 2008.

Deselaers T., Deserno T. M., « The Visual Concept Detection Task in ImageCLEF 2008 », *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the Cross-Language Evaluation Forum*, Lecture Notes in Computer Science, Aarhus, Denmark, 2008.

Fellbaum C., *WordNet - An Electronic Lexical Database*, Bradford books, 1998.

Inoue M., Grover P., « Effects of Visual Concept-based Post-retrieval Clustering in ImageCLEFphoto 2008 », *Working Notes for the CLEF 2008 workshop*, 2008.

Marsala C., Bouchon-Meunier B., « Forest of fuzzy decision trees », *Proceedings of the Seventh International Fuzzy Systems Association World Congress*, vol. 1, p. 369-374, 1997.

Marsala C., Detyniecki M., « TRECVID 2006 : Forests of fuzzy decision trees for high-level feature extraction », *TREC Video Retrieval Evaluation Online Proceedings*, 2006.

Song K., Tian Y., Gao W., Huang T., « Diversifying the image retrieval results », *ACM Multimedia*, ACM, New York, NY, USA, p. 707-710, 2006.

Tollari S., Glotin H., « Web Image Retrieval on ImagEVAL : Evidences on visualness and textualness concept dependency in fusion model », *ACM Conference on Image and Video Retrieval (CIVR)*, p. 65-72, 2007.

Tollari S., Mulhem P., Ferecatu M., Glotin H., Detyniecki M., Gallinari P., Sahbi H., Zhao Z.-Q., « A comparative study of diversity methods for different text and image retrieval approaches », *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the Cross-Language Evaluation Forum*, Lecture Notes in Computer Science, Aarhus, Denmark, 2008.

Yavlinsky A., Heesch D., Rüger S. M., « A Large Scale System for Searching and Browsing Images from the World Wide Web », *CIVR 2006*, p. 537-540, 2006.

Zhai C. X., Cohen W. W., Lafferty J., « Beyond independent relevance : methods and evaluation metrics for subtopic retrieval », *ACM SIGIR*, p. 10-17, 2003.

Modèle de langue visuel pour la reconnaissance de scènes

Trong-Ton Pham[1,2]**, Loïc Maisonnasse**[3]**, Philippe Mulhem**[1]**, Eric Gaussier**[1]

[1] *Laboratoire d'Informatique de Grenoble UJF-CNRS - 38041 Grenoble Cedex 9, France philippe.mulhem@imag.fr, eric.gaussier@imag.fr*
[2] *IPAL-I2R, Fusionopolis, Singapore 138632 - ttpham@i2r.a-star.edu.sg* [3] *Université de Lyon, INSA-Lyon, LIRIS loic.maisonnasse@imag.fr*

RÉSUMÉ. *Dans cet article, nous décrivons une méthode pour utiliser un modèle de langue sur des graphes pour la recherche et la catégorisation d'images. Nous utilisons des régions d'images (associées automatiquement à des concepts visuels), ainsi que des relations spatiales entre ces régions, lors de la construction de la représentation sous forme de graphe des images. Notre méthode gère différents scénarios, selon que des images isolées ou groupées soient utilisés comme base d'apprentissage ou de tests. Les résultats obtenus sur un problème de catégorisation d'images montre (a) que la procédure automatique qui associe les concepts à une image est efficace, et (b) que l'utilisation des relations spatiales, en plus des concepts, permet d'améliorer la qualité de la classification. Cette approche présente donc une extension du modèle de langue classique en recherche d'information pour traiter le problème de recherche et de catégorisation d'images représentées par des graphes sans se préoccuper des annotations d'images.*

ABSTRACT. *We describe here a method to use a graph language modeling approach for image retrieval and image categorization. Since photographic images are 2D data, we first use image regions (mapped to automatically induced concepts) and then spatial relationships between these regions to build a complete image graph representation. Our method deals with different scenarios, where isolated images or groups of images are used for training or testing. The results obtained on an image categorization problem show (a) that the procedure to automatically induce concepts from an image is effective, and (b) that the use of spatial relationships, in addition to concepts, for representing an image content helps improve the classifier accuracy. This approach extends the language modeling approach to information retrieval to the problem of graph-based image retrieval and categorization, without considering image annotations.*

MOTS-CLÉS : *Représentation de graphes, recherche d'image, catégorisation d'image*

KEYWORDS: *Graph representation, image retrieval, image categorization*

1. Introduction

Après presque 20 ans de recherche dans le domaine de la recherche d'images, ce sujet est toujours considéré comme un défi pour les chercheurs. Les problèmes auxquels ce domaine est confronté ont trait au fossé sémantique ainsi qu'à la manière de représenter le contenu des images. Un autre élément important est qu'il n'est pas rare qu'une image soit reliée à d'autres images. Par exemple, tous les appareils de photographie numériques incorporent l'heure et la date de prise de vue, et cette information peut être utilisée pour les grouper (Platt *et al.,* 2003). De plus, les informations de géolocalisation, qui tendent à se généraliser, peuvent également être utiles (Kennedy *et al.,* 2007). Les groupements d'image peuvent alors profiter à l'indexation et la recherche d'images. Dans ce cas, il faut intégrer des moyens de faire correspondre des groupes. Nous montrons que l'utilisation de modèles de langue peut aisément s'appliquer à des groupes d'images requêtes et documents, et qu'une telle approche est robuste par rapport aux différences entre ces groupements.

Plusieurs travaux ont par le passé proposé l'utilisation de relations spatiales entre régions d'image pour leur indexation et leur recherche. Par exemple, les descriptions par chaînes 2D (2D strings) comme on les trouve dans le système Visualseek (Smith *et al.,* 1996) capturent les séquences d'apparition d'objets suivant une ou plusieurs directions de lecture. Cependant, la recherche de telles chaînes est complexe car elle est basée sur des recherches de sous-chaînes, ce qui est coûteux. Même si des heuristiques ont été proposées, comme dans (Chang *et al.,* 2000), afin d'accélérer (d'un facteur 10) le temps de calcul. D'autres travaux ont considéré l'utilisation de régions d'images dans des modèles probabilistes, en se basant par exemple sur des modèles de Markov cachés 1D (Iyengar *et al.,* 2005) ou 2D, comme dans (Smith *et al.,* 1996) et (Yuan *et al.,* 2007). Ces travaux s'intéressent à l'annotation d'images et n'utilisent pas les relations lors du traitement des requêtes. Les relations entre des éléments d'images peuvent également être exprimés par l'intermédiaire de conventions de nommage, comme dans (Papadopoulos *et al.,* 2007) où les relations sont utilisées pour l'indexation. Enfin des travaux tels que (Mulhem *et al.,* 2006) se sont focalisés sur des graphes conceptuels pour l'indexation et la recherche des images. Les représentations explicites de relations provoquent la génération de graphes complexes, ayant un impact négatif sur la correspondance de graphe qui est déja coûteuse (Ounis *et al.,* 1998). Un aspect de notre travail est de représenter le contenu des images par des graphes, sans souffrir du poids de la complexité de la correspondance de graphes durant l'étape de recherche. Pour cela, nous proposons de nous appuyer sur un ensemble de travaux existants dans le domaine de la recherche d'information.

L'approche à base de modèles de langue pour la recherche d'information existe depuis la fin des années 90 (Ponte *et al.,* 1998). Dans ce cadre, la valeur de pertinence d'un document pour une requête donnée est estimée par la probabilité que la requête soit générée par le document. Même si cette approche a été initialement proposée pour des unigrammes (c'est-à-dire des termes isolés), plusieurs extensions ont été proposées pour traiter des *n-grammes* (i.e. des déquences de termes) (Song *et al.,* 1999, Srikanth *et al.,* 2002), et plus récemment, des relations entre termes et égale-

ment des graphes. Par exemple, (Gao *et al.,* 2004) propose a) d'utiliser un analyseur de dépendance pour représenter les documents et les requêtes, et b) une extension de l'approche à base de modèle de langue pour manipuler ces arbres. Maisonnasse *et al.* (Maisonnasse *et al.,* 2007, Maisonnasse *et al.,* 2008) ont étendu cette approche avec un modèle compatible avec des graphes plus généraux, comme ceux obtenus par une analyse conceptuelle des documents et des requêtes. D'autres approches (comme (Fergus *et al.,* 2005, Gosselin *et al.,* 2007)) ont respectivement utilisé des réseaux probabilistes et des noyaux pour capturer des relations dans les images, ce qui est également notre intention ici. Dans le cas de (Fergus *et al.,* 2005), l'estimation des probabilités des régions repose sur l'algorithme EM, qui est sensible aux probabilités initiales. Dans le modèle que nous proposons, au contraire, la fonction de vraisemblance est convexe et possède un maximum global. Dans le cas de (Gosselin *et al.,* 2007), le noyau utilisé ne considère que les trois plus proches régions d'une région de référence. Nous intégrons dans notre modèle toutes les régions voisines d'une région. Enfin, contrairement à ces travaux, à ceux de (Iyengar *et al.,* 2005) et de (Barnard *et al.,* 2003) basés sur des modèles de langues, nous utilisons explictement des étiquettes de relations spatiales. Nous étendons en fait ici le travail de (Maisonnasse *et al.,* 2007) en appliquant, d'une part, ce travail à des images, et en considérant, d'autre part, que les concepts et les relations peuvent être pondérés.

La suite de cet article est organisé comme suit : la section 2 présente le modèle de langue visuel (VLM) utilisé pour décrire le contenu des images, ainsi que la procédure de correspondance utilisée pour calculer la similarité entre images ; la section 3 décrit ensuite les résultats obtenus par notre approche pour un problème de catégorisation portant sur 101 classes ; nous concluons en section 4.

2. Le modèle de langue pour les graphes d'images

2.1. *Modélisation des images avec des graphes visuels*

Notre objectif ici est de générer automatiquement, à partir d'une image donnée, un graphe qui représente son contenu. Un tel graphe contient les concepts associés à des éléments présents dans l'image, ainsi que les relations qui dénotent comment les concepts sont reliés dans l'image. Pour cela, notre procédure est basée sur quatre étapes :

1) Identifier les régions de l'image qui vont former les blocs de base pour l'identification de concepts.

2) Indexer chaque région avec un ensemble prédéfini de caractéristiques.

3) Regrouper toutes les régions de la collection en K classes, chaque classe représentant un concept. A la fin de cette étape, chaque région de l'image est représentée par un concept, qui est le nom de la classe à laquelle la région appartient. L'ensemble des concepts, que nous notons $\mathcal{C}$, correspond donc à l'ensemble des classes obtenues.

4) Extraire enfin les relations entre les concepts.

La première étape, l'identification de régions, peut être basée sur un découpage arbitraire de blocs non recouvrants de taille égale (par exemple en divisant une image en 25 blocs, soit une division 5x5), ou bien sur des régions définies à partir de points d'intérêts (comme avec des points SIFT)[1]. Le second point vise à représenter les régions par des vecteurs afin de les regrouper. Les caractéristiques que nous avons retenues dans cet article sont des couleurs dans l'espace HSV, qui peuvent être extraites facilement et rapidement. Notre approche se base sur les K-moyennes pour la troisième étape, approche standard, mais d'autres méthodes sont également possibles. Enfin, la quatrième étape génère un ensemble de concepts associés par des relations. Nous nous concentrons ici sur les relations spatiales *au_dessus* et *à_gauche*. A la fin du processus complet, nous obtenons un ensemble de concepts reliés pour représenter une image. Il est à noter qu'un même concept peut apparaître plusieurs fois dans une image (quand différentes régions sont assignées à une même classe, comme cela arrive souvent pour des régions décrivant le ciel par exemple). Chaque concept est donc associé à une pondération qui dénote son nombre d'occurrences dans l'image. De même, chaque relation est associée à un poids dénotant le nombre de fois où elle est observée entre deux concepts donnés d'une image.

Dans la suite, nous désignerons l'ensemble des concepts pondérés qui décrivent une image par W_C. W_C est défini sur $\mathcal{C} \times \mathbb{N}$. Chaque association entre deux concepts c et c' est orientée (comme le sont les relations spatiales dans les travaux autour d'Acemedia (Papadopoulos *et al.*, 2007) par exemple), et est représentée par un triplet de la forme $< (c, c'), l, n(c, c', l) >$, où l est une étiquette de l'ensemble $\mathcal{L}$ des étiquettes possibles et $n(c, c', l)$ un entier. Un tel triplet s'interprète comme le fait qu'il existe dans l'image $n(c, c', l)$ relations portant l'étiquette l entre les deux concepts c et c'. Cette représentation permet de rendre compte de l'absence de relations entre deux concepts par la prise en compte d'une étiquette particulière. Les étiquettes que nous considérons dans la suite sont *au_dessus* et *à_gauche*, les relations inverses (*au_dessous* et *a_droite*) étant implicitement prises en compte dans la mesure où les relations sont orientées.

En résumé, un graphe représentant une image i est défini par $G =< W_C^i, W_E^i >$, avec :

$$\begin{aligned} W_C^i &= \{(c, n(c; i)), c \in \mathcal{C}\} \\ W_E^i &= \{((c, c'), l, n(c, c', l; i)), (c, c') \in \mathcal{C}^2, l \in \mathcal{L}\} \end{aligned}$$

$n(c; i)$ et $n(c, c', l; i)$ sont les poids et correspondent ici à des nombres d'occurrences.

Nous décrivons maintenant sur un exemple un tel graphe par la figure 1. Dans cette figure, considérons une image photographique I découpée en 9 blocs rectangulaires de tailles égales. Chaque bloc est associé à une étiquette élément de $\mathcal{C}$. On remarque que le concept c1 apparaît 3 fois dans l'image, ce qui amène dans la description du contenu par le graphe en bas de la figure à la notation "c1, 3". La même approche est utilisée pour obtenir $W_C^I = \{(c1, 3), (c2, 3), (c3, 2), (c4, 1)\}$. Nous utilisons les relations *à*

1. Dans ce dernier cas, les regions peuvent se recouvrir.

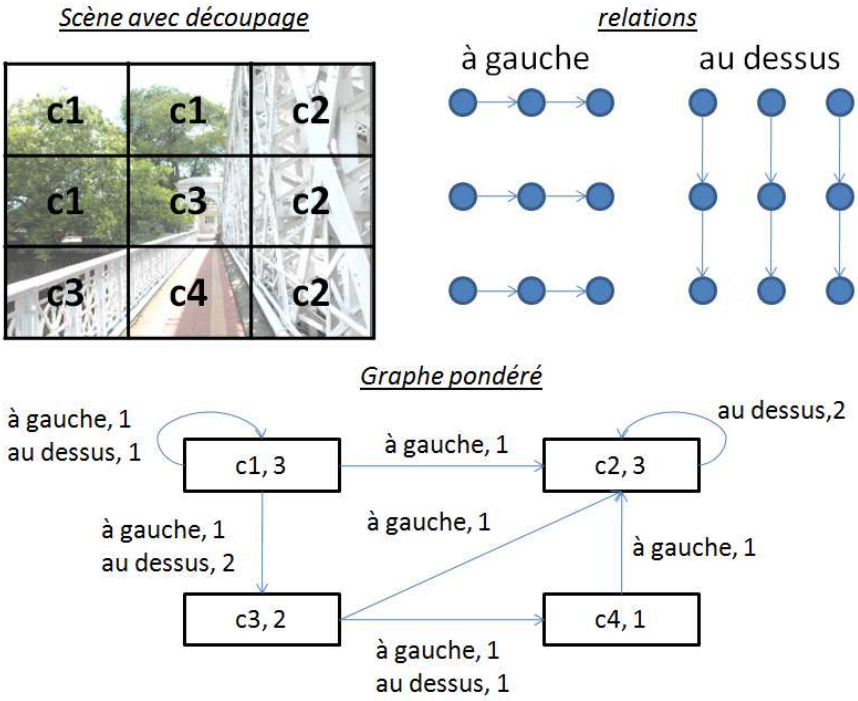


Figure 1. *Exemple de relations spatiales extraites d'une scène, avec le graphe correspondant.*

gauche et *au dessus* entre les régions, comme décrit dans la figure, afin de déterminer par exemple qu'entre les concepts c3 et c4 il existe une occurrence de la relation *à gauche*, donc $(c3, c4, \grave{a}\, gauche, 1) \in W_E^I$ et une occurrence de la relation *au dessus*, donc $(c3, c4, au\, dessus, 1) \in W_E^I$. Ce décompte se retrouve dans la description du graphe.

2.2. *Un modèle de langue pour les graphes visuels*

Notre fonction d'appariement entre images est fondée sur l'approche modèle de langue ((Ponte *et al.*, 1998)), étendue de façon à prendre en compte les éléments définis ci-dessus. De façon à différencier les images que l'on apparie, nous désignerons l'une de ces images par *image requête* et l'autre par *image document*.

La probabilité que le graphe d'une image requête $G_q (=< W_C^q, W_E^q >)$ soit généré à partir du graphe de l'image document G_d est définie par :

$$P(G_q|G_d) \quad = \quad P(W_C^q|G_d) \times P(W_E^q|W_C^q, G_d) \qquad [1]$$

Pour le premier terme du membre droit de l'équation ci-dessus, qui correspond à la probabilité de générer les concepts de l'image requête à partir du graphe de l'image document, nous nous reposons sur une hypothèse d'indépendance conditionnelle entre concepts, hypothèse classique en recherche d'information et en classification. La prise en compte des poids des concepts (c'est-à-dire, ici, des nombres d'occurrences des concepts) conduit naturellement à un modèle multinomial :

$$P(W_C^q|G_d) \quad \propto \quad \prod_{c \in \mathcal{C}} P(c|M_d)^{n(c;q)}$$

où $n(c;q)$ représente le nombre de fois que le concept c apparaît dans le graphe de l'image requête. Les paramètres du modèle $P(c|G_d)$ sont estimés par maximum de vraisemblance, avec un lissage de Jelinek-Mercer :

$$P(c|M_d) \quad = \quad (1-\lambda_u)\frac{F_1^d(c)}{F_1^d(.)} + \lambda_u \frac{F_1^{\mathcal{D}}(c)}{F_1^{\mathcal{D}}(.)}$$

où $F_1^d(c)$ représente le poids de c dans le graphe de l'image document et $F_1^d(.) = \sum_c F_1^d(c)$. Les fonctions $F_1^{\mathcal{D}}$ sont similaires, mais définies sur la collection, c'est-à-dire sur l'union des graphes des images de la collection. Le paramètre λ_u ($0 < \lambda_u < 1$) est le paramètre de lissage. Il joue le rôle d'un IDF (*Inverse Document Frequency*) ((Zhai *et al.*, 2004)) et permet de corriger une information peu fiable au niveau de l'image document par une information plus sûre extraite de la collection. Ce paramètre est en général réglé expérimentalement sur un ensemble d'apprentissage.

En suivant un processus similaire pour les relations, nous obtenons :

$$P(W_E^q|W_C^q, G_d) \quad \propto \quad \prod_{(c,c',l) \in \mathcal{C}^2 \times \mathcal{L}} P(L(c,c') = l|W_C^q, G_d)^{n(c,c',l,q)} \qquad [2]$$

où $L(c,c')$ est une variable à valeurs dans $\mathcal{L}$ qui rend compte des étiquettes possibles entre c et c'. Comme précédemment, les paramètres du modèle $P(L(c,c') = l|W_C^q, G_d)$ sont estimés par maximum de vraisemblance avec un lissage de Jelinek-Mercer, ce qui donne :

$$P(L(c,c') = l|W_C^q, G_d) = (1-\lambda_e)\frac{F_2^d(c,c',l)}{F_2^d(c,c',.)} + \lambda_e \frac{F_2^{\mathcal{D}}(c,c',l)}{F_2^{\mathcal{D}}(c,c',.)}$$

où $F_2^d(c,c',l)$ représente le nombre de fois que les concepts c et c' sont reliés par l'étiquette l dans le graphe de l'image document et $F_2^d(c,c',.) = \sum_{l \in \mathcal{L}} F_2^d(c,c',l)$. Par convention, dans le cas où l'un des deux concepts n'apparaît pas dans le graphe de l'image document :

$$\frac{F_2^d(c,c',l)}{F_2^d(c,c',.)} = 0$$

Les fonctions $F_2^{\mathcal{D}}$ sont similaires mais définies sur toute la collection (i.e., comme précédemment, sur l'union des graphes des images de la collection).

Le modèle que nous venons de présenter est inspiré du modèle défini dans (Maisonnasse *et al.*, 2008). Il en diffère cependant car (a) nous proposons ici une méthodologie complète de représentation d'une image à un niveau que nous qualifions de *conceptuel*, et (b) nous considérons ici des poids sur chaque concept et chaque relation. Dans la mesure où les poids que nous avons considérés sont des entiers, nous nous sommes reposés sur des distributions multinomiales pour modéliser l'appariement entre graphes. Des poids réels conduiraient à la considération d'autres types de distributions (par exemple de type Dirichlet). Nous allons maintenant illustrer le comportement de notre modèle dans le cadre de la classification d'image.

3. Expérimentations

Nous montrons ici la validité de notre approche dans le cadre d'une tâche de classification d'images. Plus précisément, nous voulons vérifier que a) notre proposition d'indexation conceptuelle est bien fondée, et b) que les relations spatiales ont un impact positif dans la caractérisation du contenu des images que nous proposons. Nous mettons également en évidence que notre méthodologie est robuste par rapport aux changements de scénarios présentés.

3.1. *La collection STOIC-101*

La collection *Singapore Tourist Object Identification Collection* est une collection d'images de 101 lieux d'intérêt touristique de Singapour (majoritairement des photographies d'extérieur). Ces localisations peuvent être vues comme des classes pour chacune des 3849 images. Ces images ont été prises dans leur majorité par des appareils photographiqes numériques, de manière similaire à des touristes, de 3 distances et 4 angles différents, avec des occlusions ou des cadrages partiels, de façon à obtenir au minimum 16 images par scène. De plus, les images ont été prises sous différentes conditions météo et différents style photographiques (cf. la figure 2).

L'application initiale de cette collection a été le système Snap2Tell (Lim *et al.*, 2007), dédié à la recherche d'information touristique utilisant des appareil mobiles (par exemple un assistant personnel électronique ou un téléphone portable). Pour les besoins expérimentaux, la collection STOIC-101 est divisée en deux sous-ensembles : l'ensemble d'apprentissage qui contient 3189 images (82.8% de la collection) et l'ensemble de test composé de 660 images (17.15% de la collection). En moyenne, le nombre d'images par classe est de 31,7 pour l'apprentissage et de 6,53 pour le test. Dans l'ensemble de test, le nombre minimum d'image est de 1 et le maximum de 21. Le ratio entre le nombre d'images pour l'apprentissage et le test varie de 12% à 60%. Comme un utilisateur peut prendre une ou plusieurs images de la même scène afin de poser une requête au système de recherche d'information, nous avons considéré plusieurs scénarios d'utilisation :

Figure 2. *Extraits de la base d'images STOIC-101*

1) entraîner le système sur des images isolées et traiter des requêtes d'images isolées ;

2) entraîner le système sur des images isolées, et traiter des requêtes composées d'un groupe d'images de la même scène concaténées ;

3) entraîner le système sur un groupe d'images de la même scène (en les concaténant), et traiter des requêtes d'images isolées ;

4) entraîner le système sur un groupe d'images de la même scène, et traiter des requêtes composées d'un groupe d'images de la même scène.

Le tableau 1 résume ces différents scénarios (une scène correspond à un groupe, toutes les images d'un groupe étant concaténées pour former un seul élément). Notons que

	Entraînement par IMAGE (I)	Entraînement par SCENE (S)
Requête par IMAGE (I)	√	√
Requête par SCENE (S)	√	√

Tableau 1. *Résumé des experimentations sur la collection STOIC-101*

certaines images de la collection ont été remises dans leur orientation correcte (en portrait ou en paysage).

3.2. *Indexation des images avec des concepts et des relations spatiales*

Plusieurs études sur la collection STOIC ont montré que la couleur joue un rôle prédominant, et qu'elle doit être privilégiée par rapport aux autres caractéristiques comme des caractéristiques de bordures ou de texture (Lim *et al.*, 2007). De plus, les caractéristiques de couleurs ont l'avantage d'être extraites facilement et rapidement. Pour ces raisons, nous avons choisi dans ce travail de nous baser sur des caractéristiques de couleurs RGB et HSV. Nous utilisons cependant une méthode de référence sur d'autres caractéristiques visuelles. Cette méthode, appelée *SIFT-couleur*, est similaire aux approches basées sur les points SIFT, largement utilisées et décrites dans (Lowe, 2004). Tout d'abord, les points d'intérêt sont détectés, et pour chaque point un histogramme HSV avec 32 dimensions par canal est utilisé pour indexer le bloc centré sur ce point. Les images de test sont alors comparées avec les images d'entraînement par une distance euclidienne, l'image la plus proche étant utilisée pour catégoriser l'image de test (cette approche revient donc à utiliser une classification de type plus proche voisin avec distance euclidienne).

Pour valider notre méthodologie, nous avons exploré différentes approches pour diviser chaque image en régions, et assigner à chaque région un concept. Pour la division des images en régions, nous avons retenu :

1) Une division à grain fin dans laquelle une région correspond à un pixel (cette approche donne, en moyenne, 86400 régions par image dans la collection). Nous désignons cette division par *gf*, pour grain fin ;

2) Une division "grain moyen", dans laquelle des blocs de 10x10 pixels sont utilisés, le pixel central étant le représentant de la région (cette division donne en moyenne 864 régions par image). Nous appelons cette approche *gm*, pour grain moyen ;

3) Une division grossière, dans laquelle une image est divisée en 5x5 blocs de taille égale. Cette division est appelée *gg*, pour grain grossier.

Pour les divisions *gf* et *gm*, nous avons respectivement quantifié chaque canal RGB et HSV en 8 classes de taille égale (de 0 à 64, etc.). Cela conduit à un vecteur binaire de 512 (8x8x8) dimensions pour une région. Chaque dimension correspond à un concept (défini en fonction des classes des histogrammes), pour lequel chaque dimension correspond à la présence (1) ou l'absence (0) du concept dans la région. L'image globale

est alors indexée par la somme des vecteurs de toutes ses régions. Nous désignerons ces approches par *gf-ConPred* pour “division *gf* avec des concepts préfédinis”, et *gm-ConPred* pour “division *mg* avec des concepts préfédinis”. Ces approches nous servirons de référence pour valider la méthode de regroupement proposée en section 2 pour identifier les concepts de la collection. Dans ce ce que nous venons de décrire, les concepts sont définis arbitrairement au travers des classes des histogrammes, alors qu’en section 2 ils sont définis par regroupement non supervisé.

Pour les divisions *gm* (de nouveau) et *gg*, nous regroupons les vecteurs de caractéristiques HSV de toutes les régions en $K = 500$ classes avec l’algorithme des *K-moyennes*. Cela fournit pour chaque région une affectation stricte de chaque région à un concept. L’ensemble des concepts pondérés W_C est alors obtenu en comptant combien de fois un concept apparaît dans une image. Le choix $K = 500$ est motivé par le fait que nous voulons une certaine granularité pour le nombre de concepts représentant les images. Avec trop peu de concepts, on court le risque de ne pas représenter des différences importances entre les images, alors qu’un nombre trop grand de concepts risque de rendre différentes des images qui sont similaires. Nous appelons les indexations obtenues de cette façon *gm-ConAuto* et *gg-ConAuto*, respectivement pour “division *gm* avec concepts automatiquement générés” et “division *gg* avec concepts automatiquement générés”.

De plus, pour les méthodes *gm-ConAuto* et *gg-ConAuto*, nous avons extrait les relations spatiales entre concepts décrites précédemment (*a_gauche* et *au_dessus*), et nous comptons le nombre d’occurrences de ces relations entre deux concepts donnés afin de les pondérer. La dernière étape fournit un graphe entier pour représenter les images. Nous appelons dans la suite ces deux méthodes *gm-ConAuto-Rel* et *gg-ConAuto-Rel*. Ces deux approches suivent donc le principe décrit en figure 1.

Enfin, pour classifier les images requêtes dans l’une des 101 scènes, nous avons utilisé pour toutes les méthodes d’indexation le modèle de langue pour graphe visuel présenté en section 2. Cela revient à utiliser un classifieur 1-PPV, avec la “similarité” définie par l’équation 1 et ses développements. Quand il n’y a pas de relation, le terme $P(w_E^q|G_d)$ vaut 1 (cf. équation 2), il en résulte que seuls les concepts sont utilisés pour comparer les images.

3.3. *Résultats Expérimentaux*

Les performances des différentes méthodes proposées ont été évaluées d’après le taux de reconnaissance, par image ou par scène. Ce taux est défini comme le ratio d’images (ou de scènes) correctement classifiées :

$$\text{RecoImage} = \frac{TP_i}{N_i}\,,\ \text{RecoScene} = \frac{TP_s}{N_s}$$

où TP_i, resp. TP_s, représente le nombre d’images (resp. scènes) classifié correctement. N_i est le nombre total d’images de test (i.e. 660 images), et N_s est le nombre total de scènes (i.e. 101).

Entraînement	Requête	*gf-ConPred*	*gm-ConPred*	*gm-ConAuto-Rel*	*gg-ConAuto-Rel*
I	I	0.687	0.670	**0.809**	0.551
I	S	0.653	0.650	**0.851**	0.762
S	I	0.409	0.402	0.594	**0.603**
S	S	0.940	0.940	**1.00**	0.920

Tableau 2. *Comparaison globale des differentes méthodes (meilleurs résultats en gras)*

Le tableau 2 présente les résultats que nous avons obtenus en utilisant des concepts prédéfinis et les concepts identifiés automatiquement. Nous constatons que les concepts groupés automatiquement avec un grain moyen fournissent de meilleurs résultats (la différence avec la division à grain grossier pour le scénario S-I étant marginale). Pour le scénario I-I, la méthode *couleur-SIFT* décrite précédemment atteint seulement un taux de 0,425. Ceci montre que pour cette collection notre choix de se focaliser uniquement sur des caractéristiques de couleur semble adéquate[2]. Un autre élément intéressant à souligner est que la division à grain grossier n'aide pas à généraliser par rapport à l'approche à grain moyen. En particulier, les scénarios S-I et I-S correspondent en fait à une utilisation dégénérée du système car les ensembles d'entraînement et de test sont de nature différente. Dans ces cas, il est préférable de s'abstraire d'une description très fidèle des images, afin de généraliser correctement à de nouvelles données de test. L'évolution du taux de reconnaissance des méthodes *gf-ConPred* and *gg-ConAuto-Rel* illustre ce point : les taux de reconnaissance pour les scénarios I-S et S-I sont meilleurs que ceux du scénario I-I pour *gg-ConAuto-Rel*, alors qu'il est plus mauvais pour *fg-ConPred* (ce dernier point se vérifiant également pour la méthode *mg-ConPred*, même si la différence est moins marquée, comme l'on s'y attendait). La méthore *fg-ConPred*, fondée sur une indexation qui est très fidèle à l'image originale, n'est capable de bien généraliser pour aucun des usages.

Ceci étant dit, il y a une différence importante entre les scénarios I-S et S-I : le système traite des requêtes avec davantage d'information dans le scénario I-S que dans S-I. Cette différence a un impact sur les performances pour chaque méthode : les résultats sont moins bons pour le scénario S-I que pour tous les autres, et cela pour toutes les méthodes utilisées. Nous pensons que c'est là l'explication du fait que les résultats obtenus pour la méthode *gm-ConAuto-Rel* pour S-I sont moins bons que pour I-I. Il semble qu'il y ait un plateau pour le scénario S-I autour de 0,6. Nous comptons à l'avenir explorer ce phénomène plus avant.

Nous avons aussi évalué l'utilité des relations spatiales, en comparant les résultats entre les méthodes avec et sans ces relations. Les résultats sont présentés dans le ta-

2. Sur STOIC-101, en utilisant le même découpage entrainement/test, D.-D. Le et S. Satoh, du National Institute of Informatics au Japon, ont obtenu un taux de reconnaissance de 0,744 en utilisant un système à vecteurs de support avec des caractéristiques fondées sur des moments de couleur, des motifs binaires locaux et d'orientation de bordures (communication personnelle).

Entraînement	Requête	*gm-ConAuto*	*gm-ConAuto-Rel*	*gg-ConAuto*	*gg-ConAuto-Rel*
I	I	0.789	**0.809** (+2.5%)	0.484	0.551 (+13.8%)
I	S	0.822	**0.851** (+3.6%)	0.465	0.762 (+63.8%)
S	I	0.529	0.594 (+12.3%)	0.478	**0.603** (+26.1%)
S	S	**1.00**	**1.00**	0.891	0.920 (+3.2%)

Tableau 3. *Impact des relations spatiales sur les performances (meilleurs resultats en gras ; amélioration relative par rapport à la méthode sans relation entre parenthèse)*

bleau 3. Comme nous le constatons, l'utilisation de relations spatiales améliore dans tous les cas les résultats, sauf dans le scénario S-S avec la division *gm*. Ce résultat justifie l'approche modèle de langue sur graphes, avec détection automatique de concepts et prise en compte de relations spatiales, développée dans la section 2.

4. Conclusion

Nous avons introduit dans cet article une nouvelle méthodologie pour indexer des images par des graphes de concepts reliés entre eux. Nous avons de plus proposé un modèle fondé sur le modèle de langue utilisé en recherche d'information pour apparier de tels graphes. Les graphes que nous utilisons capturent les relations spatiales entre les concepts associés à des régions dans des images. Du point de vue formel, notre modèle s'inscrit dans les approches à base de modèle de langue, et étend un certain nombre de travaux antérieurs. A un niveau pratique, l'utilisation de régions et de concepts associés permet un gain en généralité lors de la description des images, une généralité qui est bénéfique lorsque l'usage du système diffère de son environnement d'entraînement. Ceci a de grandes chances de se produire en pratique dès lors que l'on considère des collections où une ou plusieurs images peuvent être utilisées pour représenter une scène.En fonction de l'entraînement réalisé (fondé sur une image ou un ensemble d'images pour une catégorie), les résultats du système varieront.

Les expérimentations menées ont visé à estimer la validité de notre approche par rapport à ces éléments. Nous avons en particulier montré que l'utilisation de relations spatiales conduit à une amélioration significative des résultats. Le modèle proposé est capable de rechercher avec qualité des images et des ensembles d'images représentés par des graphes. De plus, nous avons montré la qualité de notre procédure pour extraire automatiquement les concepts des images, par l'utilisation d'une approche de partionnement classique (K-moyennes). Ces résultats, qui sont les meilleurs présentés sur cette collection, suggèrent qu'une division à grain moyen des images, combiné avec l'utilisation de relations spatiales, constitue une bonne stratégie pour décrire et rechercher des images.

Dans le futur, nous allons utiliser le modèle de graphe décrit ici avec différentes mesures de divergences. Le cadre que nous avons étudié ici est relié à la divergence de Kullback-Leibler. Cependant, la divergence de Jeffrey, utilisée avec succès sur des

collections d'images, pourrait avantageusement remplacer celle de Kullback-Leibler. Nous voulons également étudier les différents couplages entre grain fin, moyen et grossier, avec l'idée d'obtenir une unique représentation utilisable dans tous les cas.

Acknowledgement

Ce travail a été partiellement financé par l'Agence Nationale de la Recherche (ANR-06-MDCA-002).

5. Bibliographie

Barnard K., Duygulu P., Forsyth D., de Freitas D., Blei D., Jordan M. J., « Matching Words and Pictures », *Journal of Machine Learning Research*, vol. 2003, n° 3, p. 1107-1135, 2003.

Chang Y., Ann H., Yeh W., « A unique-ID-based matrix strategy for efficient iconic indexing of symbolic pictures », *Pattern Recognition*, vol. 33, n° 8, p. 1263-1276, 2000.

Fergus R., Perona P., Zisserman A., « A sparse object category model for efficient learning and exhaustive recognition », *Conference on Computer Vision and Pattern Recognition*, 2005.

Gao J., Nie J.-Y., Wu G., Cao G., « Dependence language model for information retrieval », *In SIGIR '04 : Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, p. 170-177, 2004.

Gosselin P., Cord M., Philipp-Foliguet S., « Kernels on bags of fuzzy regions for fast object retrieval », *International conference on Image Processing*, 2007.

Iyengar G., Duygulu P., Feng S., Ircing P., Khudanpur S. P., Klakow D., Krouse M. R., Manmatha R., Nock H. J., Petkova D., Pytlik B., Virga P., « Joint Visual-Text Modeling for automatic Retrieval of Multimedia Documents », *In ACM Multimedia, Singapore*, p. 21-30, 2005.

Kennedy L., Naaman M., Ahern S., Nair R., Rattenbury T., « How flickr helps us make sense of the world : context and content in community-contributed media collections », *In Proceedings of the 15th international Conference on Multimedia*, p. 631-640, 2007.

Lim J., Li Y., You Y., Chevallet J., « Scene Recognition with Camera Phones for Tourist Information Access », *In ICME 2007, International Conference on Multimedia & Expo*, 2007.

Lowe D. G., « Distinctive image features from scale-invariant keypoints », *Journal of Computer Vision*, vol. 60, n° 2, p. 91-110, 2004.

Maisonnasse L., Gaussier E., Chevallet J., « Revisiting the Dependence Language Model for Information Retrieval », *poster SIGIR 2007*, 2007.

Maisonnasse L., Gaussier E., Chevallet J., « Multiplying Concept Sources for Graph Modeling », *In C. Peters, V. Jijkoun, T. Mandl, H. Muller, D.W. Oard, A. Peñas, V. Petras, D. Santos, (Eds.) : Advances in Multilingual and Multimodal Information Retrieval. LNCS #5152. Springer-Verlag.*, 2008.

Mulhem P., Debanne E., « A framework for Mixed Symbolic-based and Feature-based Query by Example Image Retrieval », *International Journal for Information Technology*, vol. 12, n° 1, p. 74-98, 2006.

Ounis I., Pasca M., « RELIEF : Combining Expressiveness and Rapidity into a Single System », *In SIGIR '98 : Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, p. 266-274, 1998.

Papadopoulos T., Mezaris V., Kompatsiaris I., Strintzis M. G., « Combining Global and Local Information for Knowledge-Assisted Image Analysis and Classification », *EURASIP Journal on Advances in Signal Processing*, 2007.

Platt J. C., Czerwinski M., Field B. A., « PhotoTOC : Automatic Clustering for Browsing Personal Photographs », *Proc. Fourth IEEE Pacific Rim Conference on Multimedia*, 2003.

Ponte J. M., Croft W. B., « A language modeling approach to information retrieval », *In SIGIR '98 : Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, p. 275-281, 1998.

Smith J. R., Chang S. F., « VisualSEEk : a fully automated content-based image query system », *In Proceedings of the Fourth ACM international Conference on Multimedia*, p. 87-98, 1996.

Song F., Croft W. B., « general language model for information retrieval », *CIKM'99*, p. 316-321, 1999.

Srikanth M., Srikanth R., « Biterm language models for document retrieval », *Research and Development in Information Retrieval*, p. 425-426, 2002.

Yuan J., Li J., Zhang B., « Exploiting spatial context constraints for automatic image region annotation », *In Proceedings of the 15th international Conference on Multimedia*, p. 25-29, 2007.

Zhai C., Lafferty J., « A study of smoothing methods for language models applied to information retrieval », *ACM Trans. Inf. Syst.*, vol. 22, n° 2, p. 179-214, 2004.

Chapitre 3

Recherche et Distribution de l'Information

Clustering en recherche d'information : Concentration vs. Distribution de l'information pertinente

Sylvain Lamprier, Tassadit Amghar, Bernard Levrat et Frédéric Saubion

LERIA - Université d'Angers
2, Bd Lavoisier 49000 Angers
{lamprier,amghar,levrat,saubion}@info.univ-angers.fr

RÉSUMÉ. *S'appuyant sur la Cluster Hypothesis, qui stipule que les documents pertinents à une requête tendent à être plus proches les uns des autres que des documents non pertinents, la plupart des systèmes de recherche d'information réalisant une catégorisation de leurs résultats visent à regrouper l'ensemble des documents pertinents dans un même groupe. Nous proposons ici, par la mise en place de nouvelles mesures d'évaluation, de reconsidérer les bénéfices résultant d'une telle concentration de l'information pertinente. Contrairement à ce qui est habituellement admis, nous montrons finalement que des systèmes réalisant une distribution de l'information pertinente peuvent s'avérer au moins aussi intéressants pour l'utilisateur que des systèmes regroupant l'ensemble des documents pertinents dans un cluster unique.*

ABSTRACT. *Relying on the Cluster Hypothesis, which states that relevant documents tend to be more similar one to each other than to non-relevant ones, most of information retrieval systems producing search results as a set of clusters seek to gather all relevant documents in the same cluster. We propose here, by the settlement of new evaluation measures, to reconsider the benefits of the entailed concentration of the relevant information. Contrary to what is commonly admitted, we finally show that systems realizing a distribution of the relevant information may be at least as useful for the user as systems gathering all relevant documents in a single group.*

MOTS-CLÉS : *Recherche d'information, Clustering, Évaluation*

KEYWORDS: *Information Retrieval, Clustering, Evaluation*

1. Introduction

Généralement, un système de recherche d'information retourne, en réponse à la requête d'un utilisateur, une liste de documents ordonnée selon une estimation de leur potentiel de pertinence (Manning *et al.*, 2008). Néanmoins, dans le but de réduire l'effort à fournir pour localiser les informations pertinentes, de nombreuses approches ont proposé des présentations alternatives des résultats (Hearst *et al.*, 1996, Tombros, 2002), la plupart utilisant les relations existant entre les documents retournés pour orienter l'utilisateur dans sa recherche (Croft, 1980). Dans ce contexte, les techniques de clustering ont été largement employées afin de faciliter l'accès à l'information en regroupant les résultats aux thématiques similaires[1]. L'utilisation de ce genre de processus en recherche d'information est principalement supportée par la *Cluster Hypothesis* (Jardine *et al.*, 1971), qui stipule que les documents pertinents à une requête tendent à être plus proches les uns des autres que des documents non pertinents et que donc, ces documents ont de grandes chances de figurer dans un même cluster (Manning *et al.*, 2008). Lorsque cette hypothèse se vérifie sur un corpus de textes donné, il suffit alors à l'utilisateur d'identifier le groupe le plus en rapport avec ses besoins pour localiser l'ensemble des documents pertinents.

Alors que la plupart des systèmes réalisant une catégorisation des résultats considèrent la *Cluster Hypothesis* comme un phénomène largement bénéfique, et s'y appuient pour tenter de créer le cluster le plus informatif possible, nous aurons plutôt tendance à la considérer comme un obstacle majeur à la production de groupes permettant à l'utilisateur de localiser rapidement les informations qu'il recherche. Tout d'abord, les documents pertinents tendant à se regrouper dans un seul et même cluster, la majorité des informations initialement présentées à l'écran risquent de ne pas correspondre aux besoins de l'utilisateur. Il suffit alors que le représentant du cluster contenant l'ensemble des documents pertinents n'ait pas été judicieusement choisi pour que l'utilisateur soit incapable de localiser les informations qu'il recherche. De plus, quand bien même le représentant s'avère intéressant, l'impression première que l'utilisateur peut avoir est que peu d'informations correspondent à son sujet (ou que le système de recherche est mauvais), ce qui peut le conduire à reformuler sa requête (ou changer de système) jugeant alors que la piste suivie ne lui permettra pas de satisfaire ses besoins informationnels.

Par ailleurs, bien que la validité de la *Cluster Hypothesis* ait été vérifiée à maintes reprises (Hearst *et al.*, 1996), elle ne fait qu'énoncer des tendances générales et il est probable que certains documents pertinents ne soient pas regroupés avec les autres. Or, le fait qu'une majorité de documents pertinents appartiennent à un même groupe conduit les documents pertinents malencontreusement contenus dans les autres clusters à se trouver isolés parmi des documents déconnectés du besoin de l'utilisateur. Ces documents ont alors peu de chances de trouver dans le représentant de leur cluster (document ou liste de termes) un "porte-parole" efficace. Lorsque la majorité des documents pertinents sont contenus dans un même cluster, l'utilisateur est amené à ne

1. Voir par exemple le meta-moteur de recherche *Vivisimo* (Koshman *et al.*, 2006).

s'intéresser qu'à ce cluster en particulier, pouvant alors passer à côté de documents qui auraient pu compléter sa recherche, en apportant des informations complémentaires, en faisant part d'un point de vue différent ou même, en traitant d'un aspect différent de la question. Le paradoxe est alors considérable : alors que l'on cherche à aider un utilisateur dans sa collecte d'informations, on risque de restreindre sa perception du sujet en l'incitant à ne visiter qu'un seul cluster susceptible de ne contenir que des documents abordant la question sous un même angle.

Enfin, le fait de rassembler l'ensemble des documents pertinents dans un même groupe ne permet pas de faire émerger la structure de l'information pertinente. Or, une même requête peut comprendre un certain nombre d'aspects bien distincts. L'utilisateur, face à un jeu de clusters dans lequel un unique groupe lui est présenté comme étant susceptible de lui être utile, n'a alors aucune idée de la multitude d'aspects que son sujet peut présenter. Lorsqu'il entre dans le cluster des documents pertinents, il est alors face à une liste ordonnée de documents, certes "filtrée" mais qu'il faut tout de même parcourir linéairement jusqu'à avoir le sentiment d'avoir collecté suffisamment d'informations. Un problème majeur se pose alors : l'utilisateur doit prendre la décision d'arrêter la collecte d'informations alors qu'il ne connait pas la diversité des textes en relation avec sa requête[2].

Pour ces différentes raisons, nous pensons que le fait de chercher à regrouper l'ensemble des documents pertinents dans un même groupe n'est pas nécessairement le meilleur choix. Une distribution de l'information pertinente est alors certainement préférable à sa concentration dans un unique cluster. La *Cluster Hypothesis* se pose alors bien comme un obstacle à surmonter, la majorité des documents pertinents présentant une tendance naturelle à se regrouper. Un certain nombre d'approches, les approches de clustering orienté requête (Tombros, 2002), proposent de considérer la requête de l'utilisateur dans le processus de clustering, affirmant qu'une prise en compte du contexte dans lequel la catégorisation des documents est effectuée constitue un facteur déterminant pour l'obtention d'un partitionnement adapté à l'utilisation que l'on souhaite en faire. Bien que ces approches n'aient pas été nécessairement conçues dans un tel but, elles peuvent, selon nous, permettre de produire des clusters mieux organisés autour de la requête de l'utilisateur et ainsi, lorsque celle-ci comporte un certain nombre d'aspects bien distincts, de distribuer l'information pertinente dans des clusters différents. Cependant, les mesures d'évaluations existantes, qui présentent une forte tendance à favoriser les systèmes rassemblant la majorité des documents pertinents dans un même cluster, ne permettent pas de mettre en évidence les bénéfices retirés d'une telle distribution de l'information pertinente. Après avoir présenté ces

2. Notons néanmoins qu'un second processus de clustering peut être appliqué à ce cluster particulier, permettant ainsi de faire émerger une certaine structure de l'information pertinente. Dans ce cas cependant, le problème précédent risque d'être amplifié par le fait que l'on risque de donner à l'utilisateur l'impression qu'il se trouve face à un système lui permettant de percevoir l'ensemble des aspects de son sujet, alors que les documents abordant réellement la requête sous un angle différent sont susceptibles d'être contenus par les autres clusters et donc de n'être pas représentés par les groupes qui lui sont présentés.

mesures d'évaluation existantes, nous proposons, en section 2, la mise en place de nouvelles mesures permettant une comparaison plus équitable des différents types de systèmes. En section 3, nous réalisons un certain nombre d'expérimentations visant à vérifier la validité des mesures proposées. Différents types de systèmes sont finalement comparés en utilisant nos mesures d'évaluation.

2. Évaluer l'accès à l'information

L'objectif d'un système réalisant une catégorisation de ses résultats est de proposer un partitionnement d'un ensemble ordonné $\mathcal{D} = \{D_1, \ldots, D_n\}$, des n premiers documents retournés par une recherche initiale, en k clusters $\mathcal{C} = \{C_1, \ldots, C_k\}$ tels que[3] :

$$\begin{cases} \forall i \in \{1, \ldots, k\}, C_i = \{C_i^1, \ldots, C_i^{|C_i|}\} \neq \emptyset, \\ \forall i, j \in \{1, \ldots, k\}^2, i \neq j \Rightarrow C_i \cap C_j = \emptyset \\ \bigcup_{i=1}^{k} C_i = \mathcal{D} \end{cases} \quad [1]$$

S'appuyant pour la plupart sur la *Cluster Hypothesis* (Manning *et al.*, 2008), ces systèmes sont généralement évalués sur leur capacité à regrouper l'ensemble des documents pertinents dans un même cluster. La mesure la plus fréquemment utilisée est alors la mesure *MK1*, proposée dans (Jardine *et al.*, 1971), qui consiste à juger la qualité d'un ensemble de groupes de documents en considérant uniquement le meilleur cluster qu'il contient. Le score attribué à un système par cette mesure dépend du nombre et de la proportion de documents pertinents dans le meilleur cluster que l'on puisse trouver parmi l'ensemble $\mathcal{C}$ des clusters produits par le système :

$$MK1(\mathcal{C}) = \min_{C_i \in \mathcal{C}} \Big(1 - \frac{(1+\beta^2) \times Precision(C_i) \times Rappel(C_i)}{\beta^2 \times Precision(C_i) + Rappel(C_i)}\Big)$$
$$\text{Où} \quad Precision(C_i) = \frac{|\mathcal{P}ert \cap C_i|}{|C_i|} \quad \text{et} \quad Rappel(C_i) = \frac{|\mathcal{P}ert \cap C_i|}{|\mathcal{P}ert|} \quad [2]$$

avec $\mathcal{P}ert \subseteq \mathcal{D}$ correspondant à l'ensemble des documents pertinents. Selon (Jardine *et al.*, 1971), la mesure *MK1* présente l'avantage d'isoler la qualité du clustering des biais induits par l'utilisation d'une stratégie de recherche particulière. Néanmoins, ne permettant pas l'évaluation de systèmes proposant une distribution de l'information pertinente, cette mesure ne peut pas être utilisée dans notre étude.

Afin d'évaluer les systèmes réalisant un clustering des résultats de la même façon que les systèmes classiques (en utilisant par exemple la mesure de précision moyenne), certaines approches ont proposé de reconstruire des listes ordonnées de documents à partir des clusters formés (Hearst *et al.*, 1996, Bellot *et al.*, 1999). Pour ce faire, les

3. Dans ce qui suit, $|A|$ correspond au cardinal de l'ensemble A.

approches commencent généralement par définir un ordre entre les clusters (couramment selon la proximité à la requête de leur document qui en est le plus proche ou bien la similarité moyenne de leurs documents à la requête) et entre les documents de chaque groupe (selon leur similarité avec la requête ou avec le représentant du groupe) pour obtenir un ensemble ordonné $\mathcal{C} = \{C_1, \ldots, C_k\}$ de k sous-ensembles ordonnés $C_i = \{C_i^1, \ldots, C_i^{|C_i|}\}$ (C_i^j correspond alors au j-ième document du i-ième cluster C_i). Les approches d'évaluation par reconstruction de listes ordonnées construisent leur liste finale $L = \{L_1, \ldots, L_n\}$ selon les chemins que l'utilisateur peut emprunter dans cette liste de listes définissant l'ensemble des parcours possibles. La seule contrainte sur la construction d'une telle liste concerne l'ordre dans lequel les documents d'un cluster sont examinés : le document C_i^{j+1} ne peut être d'indice inférieur à C_i^j dans la liste finale L (puisque sauf exceptions, l'utilisateur examine les documents dans l'ordre où ils sont listés).

Une fois la liste L produite, il est alors possible d'y appliquer une mesure d'évaluation classique, telle que la populaire mesure de précision moyenne (*Average Precision Ap*), qui correspond à la moyenne des précisions (proportion de documents pertinents dans un ensemble de documents) calculées après chaque document pertinent de la liste ordonnée. Si l'on dispose d'une fonction $Pert : \mathcal{D} \rightarrow \{0, 1\}$ retournant 1 si le document considéré est pertinent (0 sinon), la précision moyenne d'une liste $Ap : \mathcal{D} \rightarrow [0, 1]$ s'obtient par :

$$Ap(L) = \frac{1}{\sum_{i=1}^{n} Pert(L_i)} \times \sum_{i=1}^{n} \sum_{j=1}^{i} \frac{Pert(L_i) \times Pert(L_j)}{i} \qquad [3]$$

Cette mesure, qui considère le rang des éléments pertinents dans la liste produite, permet d'évaluer un parcours réalisé à travers les groupes proposés, et ainsi, de rendre compte de la capacité d'accès à l'information pertinente.

Les différences entre les approches d'évaluation par reconstruction de listes ordonnées résident alors dans les parcours réalisés à travers les clusters. Les deux approches de reconstruction de listes les plus répandues s'inspirent des parcours souvent réalisés dans les arbres de recherche : alors que le parcours en profondeur examine les clusters les uns après les autres en commençant par celui possédant le plus fort potentiel de pertinence, le parcours en largeur considère séquentiellement les premiers documents non encore examinés de chaque liste. Ces deux parcours, qui représentent les deux extrêmes de l'ensemble des chemins qu'un utilisateur peut emprunter, observent une corrélation inverse : alors que le parcours en profondeur favorise les systèmes regroupant l'ensemble des documents pertinents dans un même cluster, le parcours en largeur favorise les systèmes réalisant une distribution des documents pertinents sur l'ensemble des groupes. Bien que leur utilisation conjointe permet d'obtenir certaines indications sur les performances d'un système, ces deux parcours présentent, selon nous, un certain nombre de limites :

– L'amélioration du score d'une de ces deux évaluations tend à faire diminuer celui de l'autre. Cela rend difficile l'interprétation des résultats obtenus, d'autant plus que

l'équilibre de ces deux mesures n'est pas évident.

– La considération conjointe de ces deux critères pour comparer des clusterings produits par différentes méthodes tend à pénaliser ceux qui réalisent une distribution de l'information pertinente puisqu'il est plus difficile d'obtenir un score élevé selon un parcours en largeur que selon un parcours en profondeur.

– Les caractéristiques des partitionnements produits (tailles des clusters, degré de répartition de l'information pertinente) ont un fort impact sur les scores obtenus.

– Les scores obtenus dépendent fortement de l'ordre dans lequel sont présentés les clusters. Cela biaise les résultats puisque cet ordre n'a pas d'impact direct sur la route qu'un utilisateur emprunte à travers les clusters : à partir des descriptions présentées, l'utilisateur identifie les aspects qui semblent répondre au mieux à ses besoins et parcourt les clusters correspondants en priorité.

2.1. *Parcours moyen*

Au vu de ces différentes observations, nous cherchons à définir un compromis efficace entre ces deux parcours extrêmes, qui puisse permettre d'évaluer la capacité d'accès à l'information pertinente à partir d'un ensemble de groupes proposés, sans favoriser un type de système par rapport à un autre. Nous proposons alors de réaliser l'évaluation d'un partitionnement par considération du parcours "moyen" qu'il est possible d'effectuer à travers les groupes proposés. Cela revient à considérer l'espérance mathématique du score de précision moyenne pour un parcours effectué par un utilisateur "aveugle" (*i.e.,* qui n'oriente pas sa recherche selon les informations qu'il a déjà collectées dans les différents groupes). Étant donnée une fonction $exam : \{0, \ldots, n-1\} \times \{0, \ldots, |\mathcal{P}ert|\} \rightarrow \mathcal{D}$, définie telle que $exam(t, p)$ retourne le prochain document examiné après avoir déjà rencontré t documents dont p sont pertinents, cette espérance $ExpAP(\mathcal{C})$ peut être calculée par[4] :

$$\frac{\sum_{t=0}^{n-1} \sum_{i=1}^{k} \sum_{j=1}^{|C_i|} \sum_{p=0}^{|\mathcal{P}ert|} Pert(C_i^j) \times P(exam(t,p) = C_i^j) \times \frac{p+1}{t+1}}{|\mathcal{P}ert|} \quad [4]$$

Toute la difficulté réside alors dans le calcul de la probabilité $P(exam(t, p) = C_i^j)$. Il n'est en effet pas raisonnable de chercher à l'obtenir en testant, pour tous les documents et toutes les positions t et p envisageables, toutes les possibilités de parcours partiels permettant d'examiner C_i^j après t documents dont p pertinents. Afin de réduire le nombre de calculs à effectuer, nous choisissons alors de travailler avec des configurations $\vec{x}$, dont les composantes x_i (avec $i \in \{1, \ldots, k\}$) représentent les nombres de documents déjà examinés dans chaque cluster C_i (et respectent donc l'axiome suivant : $\forall i \in \{1, \ldots, k\}, x_i \leq |C_i|$). Avec $\mathcal{X}$ l'ensemble de toutes les configurations possibles, nous considérons alors une fonction $sel : \mathcal{X} \rightarrow \mathcal{C}$, de sélection du prochain

4. $P(A)$ dénote, de manière classique, la probabilité de l'évènement A.

cluster à parcourir à partir d'une configuration donnée, qui nous permet de définir la probabilité de sélectionner un cluster C_i (avec $i \in \{1, \ldots, k\}$) selon une configuration $\vec{x}$ de documents déjà examinés[5] :

$$P(sel(\vec{x}) = C_i) = \frac{1}{|\{j \in \{1, \ldots, k\}, x_j < |C_j|\}|} \quad [5]$$

De manière à évaluer la probabilité de rencontrer une configuration $\vec{x}$ donnée, nous définissons alors une notion de voisinage entre configurations par le biais d'une fonction $V : \mathcal{X} \times \mathcal{C} \rightarrow \mathcal{X}$ telle que $V(\vec{x}, C_i)$ représente la configuration permettant d'atteindre $\vec{x}$ en examinant le premier document non encore examiné dans le cluster C_i :

$$V(\vec{x}, C_i) = (y_1, \ldots, y_k) \mid y_i = x_i - 1 \ \wedge \ \forall j \in \{1, \ldots, k\}, j \neq i \Rightarrow y_j = x_j \quad [6]$$

Ainsi, la probabilité de rencontrer une configuration donnée $\vec{x}$ peut être évaluée par[6] :

$$P(\vec{x}) = \sum_{i \in \{1,\ldots,k\}, x_i > 0} P(V(\vec{x}, C_i)) \times P(sel(V(\vec{x}, C_i)) = C_i) \quad [7]$$

Il est alors possible de calculer la probabilité $P(exam(t, p) = C_i^j)$ qui nous intéresse :

$$P(exam(t, p) = C_i^j) = \sum_{\vec{x} \in config(C_i^j, t, p)} P(\vec{x}) \times P(sel(\vec{x}) = C_i) \quad [8]$$

où $config(C_i^j, t, p)$ correspond à l'ensemble des configurations permettant d'examiner C_i^j après avoir déjà rencontré t documents dont p pertinents, qui est définie par une fonction $config : \mathcal{D} \times \{0, \ldots, n-1\} \times \{0, \ldots, |\mathcal{P}ert|\} \rightarrow 2^{\mathcal{X}}$ telle que :

$$\begin{aligned} config(C_i^j, t, p) = \quad & \{(x_1, \ldots, x_k) \in \mathcal{X} \mid x_i = (j-1) \wedge \\ & \textstyle\sum_{l=1}^{k} x_l = t \ \wedge \ \sum_{a=1}^{k} \sum_{b=1}^{x_a} Pert(C_a^b) = p\} \end{aligned} \quad [9]$$

De tels calculs restent relativement complexes, mais leur emploi pour l'évaluation de partitionnements de 50 documents paraît tout à fait envisageable (autour de 10 secondes sur un *Pentium 4, 3GHz PC*). Par ailleurs, le score $ExpAP$ peut être estimé statistiquement, en effectuant des parcours aléatoires à travers les clusters (respectant les probabilités de sélection de cluster) et en calculant la moyenne des précisions moyennes obtenues sur les listes résultant de ces différents parcours. Des expérimentations[7] ont montré que cette estimation se révélait très proche du score réel que l'on aurait pu obtenir avec la formule 4, et très robuste pour des nombres de documents considérés supérieurs à 50, quels que soient le corpus utilisé et le nombre de clusters produits. Cette estimation peut alors être utilisée lorsque les calculs de l'espérance paraissent trop complexes.

---

5. Où $|\{j \in \{1, \ldots, k\}, x_j < |C_j|\}|$ correspond au nombre de groupes n'ayant pas encore été entièrement examinés dans la configuration $\vec{x}$.
6. Il est à noter que $P((0, \ldots, 0)) = 1$.
7. Les corpus utilisés dans ces expérimentations sont ceux décrits en section 3.

2.2. *Parcours orienté par la pertinence des documents*

Dans (Leuski, 2001), Leuski propose un compromis entre les deux extrêmes que représentent le parcours en profondeur et le parcours en largeur en considérant les jugements de pertinence établis lors de l'annotation du corpus. L'idée est de définir une stratégie de parcours qui tente de simuler le comportement qu'aurait pu avoir un utilisateur réel face aux différents clusters présentés par le système à évaluer : lorsque le nombre de documents non pertinents examinés dans un cluster dépasse le nombre de documents pertinents examinés dans ce même cluster, le processus stoppe le parcours de la liste correspondant au cluster courant pour s'intéresser à un autre groupe de documents. Le nouveau groupe choisi correspond alors au cluster dans lequel la meilleure proportion de documents pertinents a été observée sur les documents examinés. Cette stratégie simule le parcours qu'un utilisateur aurait pu emprunter en ce sens qu'elle se sert des éléments examinés pour orienter son parcours vers les clusters les plus susceptibles de contenir les informations pertinentes. La liste de documents finalement obtenue par le parcours réalisé est alors supposée mieux refléter la réalité que les parcours de clusters en profondeur ou en largeur communément employés, ce qui laisse augurer d'une meilleure évaluation du système. Néanmoins, si cette évaluation limite quelque peu les biais présentés par les parcours en profondeur et en largeur, elle favorise elle aussi les méthodes regroupant la plupart des documents pertinents dans un même groupe. En effet, l'intervalle existant entre le dernier examen d'un document pertinent dans un cluster et la prise de décision de changer de cluster implique des prises en compte de documents non pertinents. Or, le nombre de changements de clusters requis est plus important lorsque l'on considère un partitionnement où l'information pertinente est distribuée sur l'ensemble des groupes.

Les jugements de pertinence, utilisés pour orienter la stratégie de parcours proposée dans (Leuski, 2001), peuvent être inclus dans le calcul du score d'évaluation du parcours moyen afin de simuler un parcours réalisé par un utilisateur plus expérimenté. Ainsi, afin d'orienter plus probablement la recherche vers des clusters qui, selon leurs documents déjà examinés, présentent un fort ratio de documents pertinents, la probabilité $P(sel(\vec{x}) = C_i)$ peut être remplacée dans les formules 7 et 8 par :

$$\frac{0.5 + \sum_{j=1}^{x_i} Pert(C_i^j)}{x_i + 1} \Big/ \sum_{l \in \{1,\ldots,k\}, x_l < |C_l|} \frac{0.5 + \sum_{j=1}^{x_l} Pert(C_l^j)}{x_l + 1} \qquad [10]$$

Au début du processus, tous les clusters possèdent la même probabilité de sélection. Selon le nombre et les positions des documents pertinents qu'ils contiennent, la probabilité de sélection de chaque cluster évolue au fur et à mesure que de nouveaux documents sont examinés. Contrairement à la stratégie de recherche proposée dans (Leuski, 2001), ce parcours ne requiert pas de déterminer un ordre entre les clusters. Les ratios de documents pertinents dans chaque cluster sont considérés en parallèle, ce qui permet d'écarter le biais, énoncé plus haut pour la stratégie de recherche proposée dans (Leuski, 2001), concernant l'intervalle existant entre le dernier examen d'un document pertinent dans un cluster donné et la prise de décision de changer de cluster.

2.3. *Parcours orienté par la proximité des documents pertinents*

Tel que c'est le cas pour la prise en compte des retours de pertinence, il est possible d'inclure une considération du contenu des documents examinés dans le calcul du score d'évaluation du parcours moyen. Il s'agit alors d'orienter plus probablement la recherche vers des clusters dont le contenu des documents examinés semble correspondre aux informations portées par les documents pertinents. La probabilité $P(sel(\vec{x}) = C_i)$ peut alors être remplacée, dans les formules 7 et 8, par :

$$\frac{\dfrac{0.5 + \sum_{j=1}^{x_i} \sum_{D \in \mathcal{P}ne(\vec{x})} \dfrac{Sim(D, C_i^j)}{|\mathcal{P}ne(\vec{x})|}}{x_i + 1}}{\sum_{l \in \{1,\ldots,k\}, x_l < |C_l|} \dfrac{0.5 + \sum_{j=1}^{x_l} \sum_{D \in \mathcal{P}ne(\vec{x})} \dfrac{Sim(D, C_l^j)}{|\mathcal{P}ne(\vec{x})|}}{x_l + 1}} \qquad [11]$$

où $\mathcal{P}ne(\vec{x})$ correspond à l'ensemble des documents pertinents n'ayant pas encore été examinés dans la configuration $\vec{x}$. Les probabilités de sélection des clusters dépendent alors des similarités moyennes des documents examinés dans chacun d'entre eux avec les documents pertinents n'ayant pas encore été rencontrés. Afin de travailler avec des valeurs de similarité suffisamment dispersées pour que les différences puissent influer sur la recherche, le processus d'évaluation commence par identifier les similarités minimales et maximales de chaque document pertinent et utilise ces valeurs pour répartir ses similarités sur $[0, 1]$. De la même façon qu'avec le parcours moyen utilisant des jugements de pertinence, tous les clusters possèdent initialement la même probabilité d'être sélectionnés. Ces probabilités évoluent ensuite selon le contenu des documents rencontrés dans chaque cluster. Le fait de ne considérer que les documents pertinents non rencontrés permet d'orienter la recherche vers d'autres clusters lorsque tous les documents pertinents connectés à la thématique d'un cluster donné ont déjà été examinés. Ce critère d'évaluation, fondé sur la proximité des pertinents plutôt que sur des retours binaires de pertinence, prend alors en compte les informations contenues par l'ensemble des documents, qu'ils soient pertinents ou non, pour orienter la recherche vers les clusters les plus pertinents, ce qui nous semble mieux correspondre aux comportements réels des utilisateurs.

Certains systèmes présentent les clusters de documents par des labels décrivant leur contenu. L'utilisateur peut alors s'appuyer sur ces descriptions pour choisir les clusters qui lui semblent le mieux correspondre à ses besoins. Pour effectuer une évaluation réaliste, il est envisageable, dans le cas de tels systèmes, d'inclure une prise en compte de la similarité de ces labels avec les documents pertinents dans les calculs des probabilités de sélection donnés par la formule 11. Par ailleurs, avec des systèmes décrivant le contenu des clusters en sélectionnant un document représentatif dans chacun d'entre eux, il est probable que les utilisateurs examinent l'ensemble des représentants de cluster avant de se lancer dans le parcours des groupes qui leur sont proposés. Les k représentants de cluster sont alors susceptibles d'être les k premiers documents à

être rencontrés (probablement dans l'ordre où ils sont présentés). Pour être plus réaliste, notre mesure d'évaluation peut alors inclure cette observation dans ses calculs, en fixant $P((1,\ldots,1)) = 1$ et en considérant $P(exam(i-1, \sum_{j=1}^{i-1} Pert(C_j^1)) = C_i^1) = 1$ pour chacun des représentants de cluster[8] (les autres probabilités d'examen sont fixées à 0 pour ces documents). Néanmoins, étant donné le faible impact qu'elle a sur les résultats finaux[9], une telle orientation de la recherche peut être omise.

3. Expérimentations

Quatre collections de documents tirées du corpus de la conférence TREC-1 ont été utilisées dans nos expérimentations. Le corpus ZIFF contient des articles informatiques, FR des extraits du registre fédéral des États Unis et AP et WSJ des articles de presse édités par l'Associated Press et le Wall Street Journal respectivement. Le même jeu de requêtes, les topics 1-50 de TREC, a été utilisé pour chaque corpus.

Les expérimentations présentées concernent le partitionnement des premiers documents retournés, en réponse à chaque requête, par le très populaire système Smart (Salton *et al.*, 1988). Deux algorithmes de clustering sont utilisés : les classiques méthodes K-Means (Tou *et al.*, 1974) et Group-Average (Tombros, 2002). Alors que la méthode K-Means produit une partition de l'ensemble de documents en optimisant la similarité de chaque élément avec le centre du cluster auquel il appartient, la méthode hiérarchique Group-Average construit un clustering en fusionnant, à chaque itération, les deux plus proches clusters jusqu'à ce que le nombre de clusters désiré soit atteint. Deux mesures de similarité inter-documents, qui s'appuient sur le modèle vectoriel (Baeza-Yates *et al.*, 1999) où les textes sont représentés par des vecteurs des poids de leurs termes significatifs[10], sont utilisées dans nos expérimentations : la mesure *Cosine* et la *Query-Sensitive Similarity Measure*. Selon la mesure *Cosine*, la similarité $Sim(D_i, D_j)$ de deux documents D_i et D_j est donnée par :

$$\frac{\sum_{l=1}^{t} w_{D_i,l} \times w_{D_j,l}}{\sqrt{\sum_{l=1}^{t} w_{D_i,l}^2 \times \sum_{l=1}^{t} w_{D_j,l}^2}} \text{ où } w_{D_i,l} = (1 + \ln(tf_{D_i,l})) \times \ln \frac{N}{n_l} \qquad [12]$$

où t représente le nombre de termes uniques du corpus, $tf_{D_i,l}$ le nombre d'occurrences du terme l dans le document D_i et n_l le nombre de documents, parmi les N documents du corpus, dans lesquels le terme l apparaît.

La *Query Sensitive Similarity Measure* (*QSSM*), présentée dans (Tombros *et al.*, 2004), semble être l'approche de clustering orienté requête la plus performante. Cette mesure réalise, pour le calcul de la similarité entre deux documents, un produit entre

8. En supposant d'avoir placé les représentants en début de cluster : $\forall_{i \in \{1,\ldots,k\}, Rep(C_i)=C_i^1}$
9. Les premiers documents des clusters ont, dans tous les cas, une forte probabilité d'être examinés en début de recherche.
10. Ici, les termes sont les stemmes des mots porteurs de sens, obtenus après avoir supprimé les mots trop fréquents (stop-list) et avoir appliqué un processus de stemmatisation (Porter, 1980).

leur score de similarité thématique classique (en terme de *Cosine*) et un score de proximité à la requête du vecteur correspondant à l'intersection de leurs deux représentations vectorielles (où le poids de chaque terme correspond à la moyenne de ses poids dans les vecteurs individuels des deux documents). De cette manière, les documents contenant des termes de la requête différents, ce qui peut être considéré comme un indicateur de thématiques différentes, obtiennent un score de similarité minoré. À l'inverse, deux documents partageant un grand nombre de termes de la requête, et qui donc ont des chances de correspondre à un même aspect du sujet, voient leur similarité renforcée par cette prise en compte de leur proximité commune à la requête.

Quatre systèmes sont étudiés dans nos expérimentations :

– G, C : la méthode hiérarchique *Group-Average* utilisant la mesure *Cosine* ;

– K, C : la méthode *K-Means* utilisant la mesure *Cosine* ;

– G, Q : la méthode hiérarchique *Group-Average* utilisant la mesure *QSSM* ;

– K, Q : la méthode *K-Means* utilisant la mesure *QSSM*.

Dans les expérimentations réalisées, nous avons choisi d'ordonner les documents de chaque cluster selon leur similarité avec leur représentant (en terme de *Cosine*). Chaque représentant correspond au document le plus proche de la requête parmi les documents contenus dans le cluster concerné. Lorsque nécessaire, un ordre entre les groupes de documents est déterminé selon la similarité du représentant de chaque groupe avec la requête (en terme de *Cosine*). Dans l'ensemble de nos expérimentations, nous utiliserons les notations données par la table 1.

NbP	Nombre de représentants pertinents
$MK1$	Qualité du groupe optimal
PRO	Parcours en profondeur
LAR	Parcours en largeur
LEU	Parcours proposé dans (Leuski, 2001)
$PM1$	Parcours moyen
$PM2$	Parcours orienté par la pertinence des documents
$PM3$	Parcours orienté par la proximité des documents pertinents

Tableau 1. *Notations des mesures d'évaluation*

3.1. *Étude des mesures proposées*

3.1.1. *Degré de corrélation avec le comportement des utilisateurs*

L'objectif des mesures est de rendre compte de la capacité qu'aura un utilisateur à atteindre les documents qu'il recherche à partir de la liste de clusters qui lui est présentée. Pour être de bons indicateurs de cette capacité, les parcours de clusters doivent se rapprocher au maximum des routes qu'un utilisateur réel aurait pu suivre. Nous proposons alors ici d'évaluer la corrélation entre les scores obtenus par nos mesures

et ceux résultant de parcours d'utilisateurs réels. Pour ce faire, nous avons demandé à dix personnes volontaires de chercher à localiser le plus rapidement possible, pour 20 différentes requêtes, les documents pertinents en utilisant les partitionnements produits par nos quatre systèmes. Afin d'obtenir des résultats représentatifs de recherches réelles, nous leur avons expliqué que les différents groupes présentés étaient supposés représenter différents aspects de l'ensemble des documents considérés et qu'ils devaient alors lire chacun des documents qui leur seraient présentés pour orienter leur recherche vers les groupes qui leur semblent le plus probablement contenir les informations pertinentes. Ainsi, avant chaque examen de document, l'utilisateur doit choisir un index de groupe à explorer. Le premier document non encore examiné du cluster sélectionné lui est alors présenté et, selon le contenu de celui-ci, doit prendre la décision de continuer dans l'examen de ce cluster (le documents sont ordonnés dans les clusters selon leur similarité avec leur représentant), ou bien de passer à un autre groupe, qui lui semble mieux correspondre à la description de la requête qui lui a été fournie. En fin de processus, la moyenne des scores de précision moyenne obtenus sur les parcours suivis par les différents sujets est calculée pour chacune des requêtes. La corrélation entre les variations des scores obtenus par des utilisateurs réels et celles des scores calculés par les mesures d'évaluation[11] est estimée au moyen d'un coefficient $R^2_{o,m}$ dont les valeurs sont données, pour les différents systèmes, par la table 2.

$R^2_{o,m}$	PRO	LAR	LEU	PM1	PM2	PM3
G, C	0.85	0.80	0.89	0.83	0.92	0.92
K, C	0.51	0.76	0.62	0.81	0.86	0.90
G, Q	0.72	0.72	0.83	0.78	0.89	0.91
K, Q	0.47	0.71	0.55	0.74	0.81	0.88

Tableau 2. *Corrélation entre variations observées*

La méthode *Group-Average* conduit bien souvent à l'obtention d'un cluster contenant un très grand nombre de documents. Les choix proposés aux utilisateurs sont alors bien moins nombreux puisque les autres clusters sont souvent très vite épuisés et qu'il ne reste alors plus qu'un seul cluster contenant des documents non examinés. Il est alors bien plus aisé de suivre les variations observées. Dans ce cas, le fait que le parcours en profondeur obtienne un meilleur coefficient de corrélation que le parcours en largeur s'explique par le fait que les documents contenus par les plus petits clusters sont souvent bien marginaux et n'ont alors pas grand chose à voir avec le sujet de la recherche. Avec l'algorithme *K-Means* qui produit des clusters de tailles

11. Puisque l'objectif est d'évaluer la capacité des mesures à suivre les variations de scores des utilisateurs réels, et pas nécessairement d'obtenir les mêmes valeurs, chaque distribution de scores s (observés ou obtenus par une mesure) est normalisée selon la moyenne $\bar{s}$ et l'écart-type σ_s qu'elle observe sur l'ensemble des requêtes i : $\forall i \in \{1, \ldots, 20\}, s_i = (s_i - \bar{s})/\sigma_s$.

plus homogènes, il est plus difficile de suivre une route optimale puisque les documents pertinents sont mieux répartis dans les clusters. Cette difficulté est par ailleurs accrue lors de l'utilisation de la mesure *QSSM*[12]. Les coefficients de corrélation reportés montrent que notre parcours orienté par la pertinence des documents $PM2$, et *a fortiori* orienté par la proximité des documents pertinents $PM3$, parait bien mieux suivre les variations de score observées avec les individus impliqués dans l'étude que les autres approches d'évaluation quel que soit le système utilisé. Ils sont donc susceptibles d'être de meilleurs indicateurs de la capacité à atteindre les documents pertinents à partir d'une liste de clusters présentée à l'utilisateur.

3.1.2. *Influence du degré de répartition des documents pertinents*

Afin d'être capable de comparer les différents types de systèmes de manière équitable, le degré de concentration / distribution de l'information pertinente ne doit pas avoir un impact significatif sur les scores obtenus par les mesures d'évaluation utilisées. Dans le but de vérifier l'impartialité de nos mesures, nous proposons de réaliser des expérimentations sur des clusterings aléatoires présentant différents degrés de distribution des documents pertinents dans les clusters : après avoir aléatoirement distribué dans 5 clusters l'ensemble des documents non pertinents considérés, nous affectons chacun des documents pertinents au premier des cinq clusters avec une probabilité de 0.2, 0.33, 0.5, 0.66 ou 1 selon le type de clustering souhaité. Alors qu'une probabilité de 1 conduit à la production d'un cluster contenant la totalité des documents pertinents, une probabilité de 0.2 favorise la distribution de l'information pertinente. La figure 1 présente les différentes distributions de scores obtenus[13], par les mesures sur ces différents types de clustering.

Tel qu'attendu, l'espérance du parcours en profondeur augmente avec la probabilité d'affecter tous les documents pertinents dans un même cluster. L'espérance du parcours en largeur diminue dans le même temps. Conformément à notre intuition, le parcours proposé par (Leuski, 2001) semble largement favoriser les clusterings rassemblant tous les documents pertinents dans le même groupe. Selon les courbes présentées par la figure 1, nos parcours $PM2$ et $PM3$ paraissent constituer les mesures les plus équitables puisque leur espérance ne semble que très légèrement affectée par les variations du degré de répartition des documents pertinents dans les clusters. L'espérance du parcours $PM3$ semble néanmoins diminuer légèrement lorsque la concentration des documents pertinents augmente. Cela peut s'expliquer par le fait que les clusters considérés ne représentent pas de réelles thématiques puisqu'ils ont été produits de manière aléatoire. Ce parcours a alors des difficultés à déterminer les clusters les plus potentiellement intéressants à partir du contenu des premiers documents des cluster. Cette diminution d'espérance ne s'observe pas avec des partitionnements

12. Il est à noter que cette difficulté plus élevée ne traduit en rien une moins bonne efficacité du système, il ne s'agit ici que de la difficulté à trouver la meilleure route possible, et non de la difficulté accrue à atteindre les documents pertinents.

13. Les valeurs représentent des moyennes obtenues, pour chaque requête et chaque type de partitionnement, sur 100 clusterings des 50 premiers documents (du corpus AP) retournés. Les représentants de clusters sont ici sélectionnés aléatoirement.

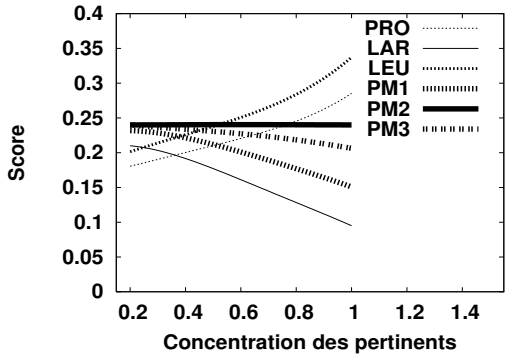


Figure 1. *Influence de la répartition des documents pertinents dans les clusters*

produits par des méthodes de clustering considérant les similarités existant entre les documents. Selon les résultats obtenus, on peut légitimement considérer les parcours $PM2$ et $PM3$ comme les plus à même de comparer des systèmes produisant des partitionnements de types différents[14].

3.2. *Comparaison des systèmes*

	ZIFF				AP				WSJ				FR			
	G,C	K,C	G,Q	K,Q	G,C	K,C	G,Q	K,Q	G,C	K,C	G,Q	K,Q	G,C	K,C	G,Q	K,Q
NbP	2.25	2.61	2.32	**2.69**	1.84	**2.38**	1.93	2.36	1.85	2.58	1.97	**2.59**	2.49	3.02	2.71	**3.21**
$MK1$	**0.62**	0.68	**0.62**	0.68	0.59	0.64	**0.58**	0.64	**0.66**	0.72	0.67	0.71	**0.55**	0.62	0.56	0.62
PRO	**0.54**	0.51	0.53	0.50	**0.55**	0.53	**0.55**	0.52	0.62	0.61	**0.63**	0.60	**0.54**	0.52	0.52	0.52
LAR	0.50	**0.54**	0.51	**0.54**	0.46	0.51	0.46	**0.52**	0.49	0.54	0.49	**0.56**	0.48	0.56	0.49	**0.58**
LEU	**0.55**	0.53	**0.55**	0.52	0.56	0.55	**0.57**	0.55	0.65	0.64	**0.66**	0.63	0.56	0.56	0.56	**0.57**
$PM1$	0.52	**0.55**	0.52	**0.55**	0.49	0.52	0.49	**0.53**	0.55	0.58	0.55	**0.61**	0.52	0.63	0.54	**0.65**
$PM2$	0.55	**0.57**	**0.57**	**0.57**	0.55	0.58	0.56	**0.59**	0.65	0.69	0.67	**0.70**	0.57	0.65	0.57	**0.67**
$PM3$	0.55	0.59	0.56	**0.60**	0.54	0.58	0.56	**0.61**	0.66	0.68	0.67	**0.70**	0.56	0.65	0.58	**0.71**

Tableau 3. *Résultats des systèmes*

La table 3 présente les résultats moyens obtenus par nos quatre systèmes réalisant une catégorisation, en cinq clusters, des 50 premiers documents retournés par la recherche initiale en réponse à chacune des requêtes. Alors que les résultats montrent que le système utilisant la méthode *Group-Average* obtient généralement le meilleur score selon la mesure *MK1*, ce système semble produire un clustering des documents à

14. Des expériences additionnelles ont par ailleurs montré que le nombre et la taille des clusters considérés n'ont pas non plus d'impact significatif sur les scores obtenus par nos mesures.

partir duquel l'accès aux documents pertinents est plus difficile qu'avec celui proposé par le système utilisant la méthode *K-Means*. En effet, alors que ce système obtient des résultats légèrement supérieurs selon un parcours en profondeur, nous observons une forte dominance de celui utilisant la méthode *K-Means* lorsque le parcours en largeur est considéré. Or, la stratégie de parcours LEU donne, bien souvent, de meilleurs scores au système utilisant la méthode Group Average : cette mesure, qui favorise les approches regroupant la plupart des documents pertinents, ne peut pas être utilisée pour comparer des systèmes produisant des clusterings de types différents. Nos mesures, qui comparent les systèmes de manière plus équitable, permettent de mettre en évidence, dans tous les cas, la supériorité des systèmes utilisant la méthode *K-Means*.

Par la considération des mesures d'évaluation existantes, l'utilisation de la mesure QSSM n'apporte pas d'amélioration significative des résultats. Les meilleurs scores obtenus par le parcours en largeur sont généralement minorés par une diminution du score obtenu par la recherche en profondeur. Il semble alors que l'augmentation de la capacité à regrouper l'ensemble des documents pertinents dans un même cluster, lorsque ceux-ci abordent le sujet de l'utilisateur sous un même angle, est contrebalancée par les pénalités que les évaluations attribuent à la mesure *QSSM*, pour avoir conduit à la distribution de l'information pertinente dans plusieurs groupes lorsque la requête comporte plusieurs aspects bien distincts. Les approches d'évaluation que nous proposons, et tout particulièrement le parcours orienté par la proximité des pertinents $PM3$, permettent de de mettre en valeur les bénéfices tirés de l'utilisation de mesures telle que la mesure *QSSM* : une prise en compte de la requête dans le processus de clustering peut effectivement permettre d'améliorer l'accès à l'information en mettant en évidence, lorsque le sujet de la requête est suffisamment large, différents aspects de l'information recherchée par l'utilisateur.

4. Conclusion

L'application de techniques de clustering sur les résultats d'une recherche d'information a pour but d'en faire émerger les thématiques principales. Cependant, le niveau de diversité des textes considérés implique bien souvent un faible degré de finesse du clustering réalisé et certaines thématiques émergentes peuvent se trouver en forte déconnection avec la requête formulée. La plupart des systèmes réalisant une catégorisation de leurs résultats y voient un effet bénéfique puisque cela permet de regrouper la plupart des textes pertinents dans un même cluster, donnant ainsi la possibilité à un utilisateur de filtrer les résultats retournés en ne parcourant que le cluster contenant les informations qui l'intéressent. Adoptant un point de vue différent, nous considérons ce phénomène comme largement négatif pour de multiples raisons, un tel regroupement de l'information pertinente ne permettant notamment pas de présenter à un utilisateur les différents aspects de sa requête et risquant alors de lui en restreindre la perception à un unique point de vue. Selon nous, une distribution de l'information pertinente sur l'ensemble des groupes formés peut s'avérer bien plus intéressante que sa concentration dans un unique cluster. Tout dépend du niveau d'accessibilité des textes pertinents

à partir de la liste de descriptions de clusters présentée. Or, nous nous sommes aperçus que la plupart des mesures d'évaluation présentaient une forte tendance à favoriser les systèmes regroupant l'ensemble des textes pertinents dans un même cluster. Nous avons alors cherché à établir des mesures réalisant une estimation plus équitable de la capacité d'accès à l'information pertinente. À la lumière des approches proposées, qui semblent refléter efficacement le comportement d'un utilisateur réel face à une liste de clusters, nous avons mis en évidence les bénéfices potentiels résultant d'une prise en compte de la requête dans le processus de clustering pour distribuer l'information pertinente dans des clusters distincts. Pour la première fois, la concentration de l'information pertinente n'est alors pas perçue comme le meilleur moyen d'aider l'utilisateur dans sa recherche. À ce titre, nous pensons que les mesures proposées ici sont susceptibles de remettre en cause nombre d'hypothèses concernant l'utilisation de techniques de clustering en recherche d'information.

5. Bibliographie

Baeza-Yates R. A., Ribeiro-Neto B. A., *Modern Information Retrieval*, ACM Press / Addison-Wesley, 1999.

Bellot P., El-Bèze M., « Query Length, Number of Classes and Routes through Clusters : Experiments with a Clustering Method for Information Retrieval », *ICSC '99*, Springer-Verlag, London, UK, p. 196-205, 1999.

Croft W. B., « A model of cluster searching bases on classification. », *Information Systems*, vol. 5, n° 3, p. 189-195, 1980.

Hearst M. A., Pedersen J. O., « Reexamining the cluster hypothesis : Scatter/gather on retrieval results », *SIGIR'96*, Zürich, CH, p. 76-84, 1996.

Jardine N., Van Rijsbergen C. J., « The use of hierarchic clustering in information retrieval. », *Information Storage and Retrieval*, vol. 7, n° 5, p. 217-240, 1971.

Koshman S., Spink A., Jansen B. J., « Web searching on the vivisimo search engine », *Journal of the American Society for Information Science and Technology*, vol. 57, n° 14, p. 1875-1887, 2006.

Leuski A., « Evaluating document clustering for interactive information retrieval », *CIKM '01*, ACM, New York, NY, USA, p. 33-40, 2001.

Manning C. D., Raghavan P., Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Porter M.F., « An algorithm for suffix stripping », *Program*, vol. 14, n° 3, p. 130-137, 1980.

Salton G., Buckley C., « Term-weighting approaches in automatic text retrieval », *Information Processing & Management*, vol. 24, n° 5, p. 513-523, 1988.

Tombros A., The Effectiveness of Query-Based Hierarchic Clustering of Documents for Information Retrieval, PhD thesis, University of Glasgow, UK, 2002.

Tombros A., Van Rijsbergen C. J., « Query-Sensitive Similarity Measures for Information Retrieval », *Knowledge Information Systems*, vol. 6, n° 5, p. 617-642, 2004.

Tou J. T., Gonzalez R. C., *Pattern recognition principles*, Applied Mathematics and Computation, Reading, Mass. : Addison-Wesley, 1974, 1974.

Routage sémantique des requêtes dans les systèmes pair-à-pair

Taoufik YEFERNY[*] – Khedija AROUR[**] – Yahya SLIMANI[*]

[*] *Faculté des Sciences de Tunis*
Campus Universitaire, Tunis 1060, Tunisie
yeferny.taoufik@gmail.com – yahya.slimani@fst.rnu.tn
[**] *Institut National des Sciences Appliquées et de Technologie de Tunis, 1080 Tunisie*
Khedija.arour@issatm.rnu.tn

RÉSUMÉ. *Les systèmes pair-à-pair (P2P) se sont imposés ces dernières années comme la technologie majeure d'accès à différentes ressources sur Internet. De nombreuses recherche concernant la sélection des meilleurs pairs contenant les données appropriées à une requête,ont émergé et constituent un axe de recherche très actif. L'efficacité de la recherche dans ces systèmes, et surtout le cas non structuré, peut être améliorée en introduisant de la sémantique dans le processus de routage des requêtes. Cette sémantique est généralement construite à partir du contenu des pairs, mais peut également faire intervenir le comportement explicite des utilisateurs. Nous présentons dans cet article un nouvel algorithme de routage des requêtes par apprentissage basé sur le comportement implicite des utilisateurs qui est déduit à partir d'un historique de requêtes. Pour tester l'algorithme proposé, nous avons défini une couche de routage sur le simulateur PeerSim qui nous a permis d'évaluer l'efficacité de notre algorithme.*

ABSTRACT. *The Peer-to-Peer systems (P2P) were led these last years as the major technology of access upon various resources on Internet. The efficiency of the researches in these systems, and especially the not structured case, can be improved by introducing of semantics into the process of routing of the requests. This semantics is generally built from the contents of the peers, but can also bring in the explicit behavior of the users. We present in this paper a new algorithm of routing of the requests by learning on the implicit behavior of the users which is deducted from a history of requests. To test the proposed algorithm, we defined a layer of routing on the simulator PeerSim.*

MOTS-CLÉS : *Systèmes P2P, Routage sémantique, Recherche d'information, Apprentissage, Simulateur PeerSim.*

KEYWORDS: *P2P, Semantic routing, Information retrieval, Learning, PeerSim Simulator.*

1. Introduction

Ces dernières années les systèmes P2P sont devenus très populaires car ils offrent la possibilité aux utilisateurs de partager et d'accéder à des ressources diverses, distribuées à large échelle. Il existe de nombreuses architectures des systèmes P2P, qui utilisent différentes techniques de localisation des données, qui se traduisent par différentes méthodes de routage des requêtes (Defude, 2007). Nous pouvons classifier ces systèmes selon leur modèle de recherche sous-jacent, qui peut être soit non structuré (propagation aléatoire des requêtes dans le graphe des pairs), soit structuré (propagation des requêtes selon une structure d'organisation des pairs basée en général sur des fonctions de hachage). Chaque type de système possède des avantages et des inconvenants. Parmi les avantages des systèmes non structurés, nous pouvons signaler le fait qu'ils respectent au mieux l'autonomie des pairs et ils supportent des langages de requêtes plus expressifs. Cependant, ils ne sont pas efficaces sur le plan du routage des requêtes. Plusieurs travaux de recherche ont tenté d'améliorer la méthode de routage des requêtes dans ces systèmes en introduisant la sémantique dans le processus de propagation de requête (Christoph *et al.,* 2004, Defude, 2007). Dans cet article, nous allons présenter un algorithme de routage sémantique de requêtes dans les systèmes P2P qui :

– Sélectionne les pairs les "plus pertinents" pour répondre à une requête en se basant sur le profil utilisateur.

– Propage la requête à ces pairs.

– Génère périodiquement les profils, en mettant à jour une base de connaissances.

Le reste de l'article est organisé comme suit. Dans la section 2, nous présentons un état de l'art sur le routage des requêtes dans les systèmes P2P. Une synthèse sur les méthodes de routage actuelles fera l'objet de la section 3 . La section 4 définit notre algorithme de routage sémantique de requêtes. Les résultats des différentes experimentations de notre algorithme seront discutés dans la section 5. Enfin, nous concluons cet article par une conclusion et quelques perspectives intéressantes de prolongement de notre travail.

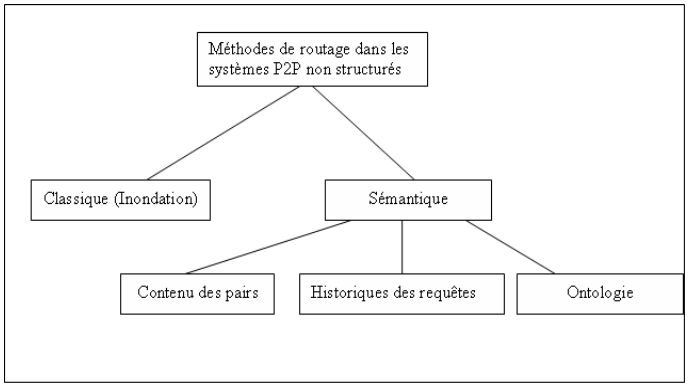


Figure 1. *Classification des méthodes de routage*

2. Etat de l'art sur le routage des requêtes dans les systèmes P2P

La solution idéale pour le routage des requêtes dans les systèmes P2P, consiste à ce qu'une requête, formulée par l'utilisateur, soit automatiquement envoyée vers l'ensemble des pairs susceptibles de fournir une réponse. Dans la littérature, plusieurs travaux de recherche ont essayé d'améliorer la méthode classique de routage des requêtes, qui propage une requête à un ensemble de pairs choisit d'une manière aléatoire, en introduisant la sémantique dans le processus de propagation de requêtes (Christoph *et al.,* 2004, Defude, 2007). Une première approche de cette méthode est appelée système de routage par contenu. Le contenu de chaque pair est résumé dans une étiquette de contenu et un routeur fait suivre les requêtes de recherche selon les meta-informations des pairs qui se sont enregistrés chez lui . Une autre approche consiste à utiliser la sémantique associée aux pairs pour router les requêtes. La figure 1 présente une classification des différentes méthodes de routage dans les systèmes P2P non structurés. Parmi celles qui sont basées sur le contenu de pairs, nous pouvons citer les approches CORI (Collection Retrievel Inference Network) (Callan *et al.,* 1995) et gGLOSS (generalized Glossary of Server's Server) (Luis *et al.,* 1994, Luis *et al.,* 1995) qui représentent la collection de chaque pair voisin, par un document de taille assez grande (superdocument). L'ensemble de tous les superdocuments forme une collection spéciale, utilisée pour calculer un score de classement de collections de chaque pair en fonction des termes de la requête. Par la suite, la requête est propagée aux voisins ayant le plus haut score. Le système de recherche d'information, PlanetP (Raja *et al.,* 2006), représente le contenu de chaque pair de manière compacte à l'aide d'un filtre de Bloom décrivant les termes des documents stockés par celui-ci. Ces filtres de Bloom sont distribués dans le réseau en utilisant un algorithme de propagation. Chaque pair possède un index global constitué d'une liste d'autres pairs, associés chacun à leur filtre de Bloom (Raja *et al.,* 2006), permettant de donner au pair une vision partielle et approchée du contenu global du réseau. Un noeud qui reçoit une requête commence d'abord par faire une recherche dans son index local. S'il ne peut pas hono-

rer la requête, il calcule les rangs des pairs de son index global et il propage la requête aux pairs de plus grand rang.
Peu de méthodes utilisent l'information sur l'historiques des requêtes. Néanmoins, la méthode REMINDIN (Christoph *et al.,* 2004) exploite les métaphores sociales pour définir une stratégie de routage des requêtes basée sur l'historique des requêtes. Pour cela, elle utilise une ontologie (un dépôt local). Dans l'implémentation de REMINDIN, ce sont des déclarations RDF construites à partir des réponses aux requêtes passées. Le choix des pairs est basé sur des observations de la connaissance des autres pairs. En effet, l'évaluation de requête, en utilisant le dépôt local associé à un noeud renvoie un ensemble de pairs triés selon les valeurs de confiance dans des ressources spécifiques pour chaque pair et des valeurs de confiance globales pour chaque pair. Si le nombre de pairs choisis est insuffisant (inférieur à un seuil), la méthode REMINDIN rafraîchie la requête et répéterait le même processus plusieurs fois.

3. Synthèse sur les méthodes de routage

La difficulté de fournir l'accès au contenu des pairs réside essentiellement dans le problème d'échelle. En effet, pour pouvoir trouver les pairs pertinents à une requête, il faut disposer de l'information sur le contenu de tous les pairs. Dans ce cas, il est nécessaire de construire un index global qui serait consulté pour le routage des requêtes. Cette solution peut être viable à petite échelle, mais elle ne s'adapte pas à la taille du réseau pair à pair. L'espace de stockage nécessaire pour l'index est également trop important et sa mise à jour est difficile. Une autre solution consisterait à envoyer chaque requête à tous les pairs (par inondation). Même si cette méthode assure que tous les pairs pertinents seront trouvés, elle implique trop de trafic sur le réseau et une surcharge de travail pour les pairs. En fait, il faut une solution intermédiaire qui permettrait d'acheminer une requête à un ensemble réduit de pairs et ceci sans accéder à un index global. Un autre problème est celui du nombre de réponses qui peuvent être obtenues à la suite d'une recherche sur un nombre important de pairs. Si l'utilisateur reçoit trop de réponses à sa question, il ne pourra pas les bien exploiter. Pour éviter cela, il faudrait imaginer, un moyen de la découverte progressive de l'espace d'information et la spécialisation des requêtes. Un problème important se pose pour les approches sociales (comme REMINDIN par exemple) qui suppose avoir une ontologie partagée par l'ensemble des pairs, ce qui n'est pas toujours disponible pour la communauté des systèmes P2P. De même, l'autre problème qui se pose est que le processus de routage est itératif : plusieurs formulation de la requête peuvent être effectuées (Chernov *et al.,* 2005).
Le tableau 1 compare les différentes méthodes étudiées selon différents critères, à savoir :

– *Information utilisée* : informations utilisées pour définir la sémantique.

– *Représentation de l'information sémantique* : structures de données utilisées pour stocker les informations.

– *Passage à l'échelle* : par passage à l'échelle, nous voulons savoir si la méthode permet de supporter un grand nombre de pairs (i.e + +), un nombre moyen (i.e + -) ou peu (i.e - -).

– *Maintenance* : fréquence des mises à jour des informations sémantiques. La fréquence peut être importante (i.e + +) ou plus au moins importante (i.e + -).

– *Espace de stockage utilisé* : espace mémoire utilisé pour stocker les informations sémantiques.

– *Partage de connaissances* : coopération des pairs pour construire des connaissances. La coopération peut être forte (i.e + +) ou sans coopération (i.e - -).

Critères / Méthodes	gGLOSS, CORI	PlanetP	REMINDIN
Information utilisée	Contenu des pairs voisins	Contenu des pairs	Historique des requêtes+ Ontologie
Représentation de l'information sémantique	Superdocument	Filtre de Bloom	Ensemble de déclarations RDF
Passage à l'échelle	- -	+ -	+ +
Espace de stockage utilisé	+ +	+ +	+ +
Partage de connaissances	- -	+ +	- -
Maintenance	+ +	+ -	+ +

Tableau 1. *Tableau comparatif des différentes méthodes de routage*

En se qui nous concerne, nous proposons une solution basée sur le modèle de routage sémantique. Le système de routage sémantique est une amélioration des méthodes classiques de routage en utilisant l'historique des requêtes. En plus des documents, notre méthode de recherche supporte les nouvelles connaissances que nous avons désigné par le terme profil. Un profil est une corrélation entre les requêtes passées et les pairs positifs ou les requêtes passées et les termes associés. Par pair positif, nous entendons les pairs qui ont agi positivement vis-à-vis d'une requête (il y a eu téléchargements des documents). Ainsi, nous pouvons dire dire que notre approche est dérivée des approches sociales, dans le sens où une requête sera routée vers les pairs partageant des connaissances communes.

4. Modèle proposé

4.1. *Introduction*

L'idée sous-jacente à notre proposition est de substituer le routage classique, basé sur la propagation par inondation itulisé dans Gnutella (Gnutella, 2007), par un routage sémantique du meilleur contenu basé sur un ensemble de profils. La solution que nous proposons supporte le facteur d'échelle en organisant les routeurs par pair et en préparant les profils d'une manière hors ligne.

4.2. *Architecture globale*

Dans notre proposition nous avons repris l'architecture du système Gnutella (Gnutella, 2007) qui est le système le plus représentatif des systèmes de recherche d'information P2P non structurés. Gnutella est un système de partage de fichiers sur Internet qui représente aujourd'hui l'archétype des systèmes P2P non structurés. Dans ce système, une requête est d'abord évaluée sur le pair d'émission puis propagée récursivement sur un sous-ensemble aléatoire des pairs voisins (principe d'inondation). La terminaison de la recherche est assurée par une détection de cycles ainsi qu'une profondeur maximale de recherche (appelée TTL ou Time To Live).
Dans le but d'améliorer l'efficacité de cette méthode, nous avons remplacé le module de propagation des requêtes par un autre module de propagation " guidée " qui utilise une base de connaissances afin de router les requêtes vers les pairs "pertinents". De plus nous avons ajouté deux autres modules, à savoir :

– Un module de gestion de profils qui s'exécute périodiquement. Son rôle consiste à construire la base de connaissances à partir des informations extraites de l'historique des requêtes. Cette historique se trouve dans un fichier log.

– Un module gestion du fichier log qui s'exécute à la réception d'une réponse.

La figure 2 illustre l'architecture globale de notre approche.

4.3. *Module gestion de fichiers logs*

A la réception d'une réponse, ce module met à jour le fichier log en ajoutant les informations relatives à cette requête, à savoir l'identifiant de la requête, l'ensemble de ses termes, les documents téléchargés et les pairs associés.

4.4. *Module de gestion de profils*

L'objectif de ce module est de générer un ensemble de profils qui représentent des corrélations sémantiques entre les requêtes passées et les pairs positifs. La génération de cet ensemble de profils se fait hors ligne au niveau de chaque pair. Pour cela, nous

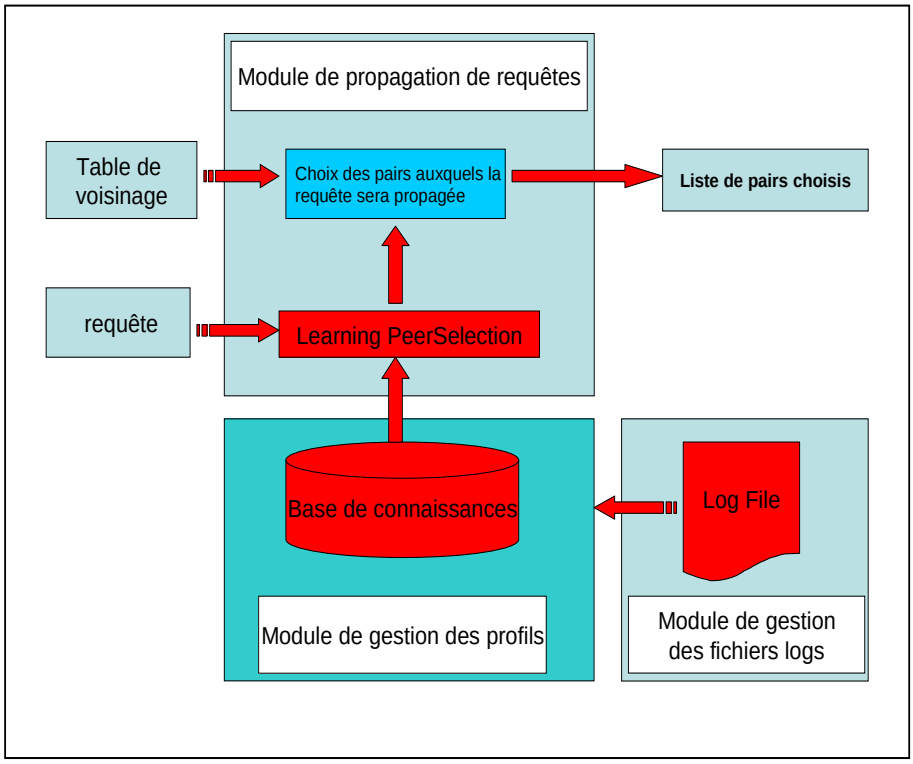


Figure 2. *Architecture globale de notre approche*

avons utilisé une approche formelle de construction de ces profils. Cette approche est basée sur l'Analyse des Concepts Formels.

– *l'Analyse Formelle de Concepts* utilise comme point de départ un « contexte formel » qui définit une relation binaire reliant les objets à leurs attributs (Ganter *et al.,* 1997) .

– *Un contexte formel* : est un triplet $K = (O, A, R)$, où O et A sont, respectivement, l'ensemble des objets et des attributs (caractéristiques), et $R \subseteq O \times A$ est une relation exprimant que $\forall(o, a) \in R$, a est un attribut de l'objet o.
Nous notons par f et g deux applications caractéristiques de R définies comme suit :

$f : P(O) \rightarrow P(A)$
$X \rightarrow \{y \in A | \forall x \in X, (x, y) \in R\}$
$g : P(A) \rightarrow P(O)$
$Y \rightarrow \{x \in O | \forall y \in Y, (x, y) \in R\}$

Un concept associe un ensemble maximal d'objets à l'ensemble d'attributs que ces objets partagent.

– Un concept est une paire $C = (X, Y)$ où $X \subseteq O, Y \subseteq A$

et

$X = \{o \in O | \forall y \in Y, (o, y) \in I\}$ est l'extension d C (objets couverts), notée $Ext(C)$

$Y = \{a \in A | \forall x \in X, (x, a) \in I\}$ est l'intension C (attributs partagés), notée $Int(C)$

De plus, nous avons la propriété suivante, pour tout concept $C = (X, Y), Y = f(X)$ et $X = g(Y)$.

Dans notre cas, nous nous sommes basés sur deux contextes qui sont des projections sur le fichier log. La première projection représente le lien entre les requêtes passées et les termes associés, appelé contexte $C1$. La seconde représente le lien entre les requêtes passées et les pairs positifs (i.e les pairs à partir desquels y a eu des téléchargements), appelé contexte $C2$. En effet, les attributs du contexte $C1$, respectivement de $C2$, sont les termes des requêtes et les pairs ayant répondu à ces requêtes. Un algorithme de génération de concepts formels est par la suite appliqué pour générer deux ensembles de concepts, notés $E1$ et $E2$. Pour ce faire, nous avons utilisé l'algorithme de Godin (Godin *et al.,* 1995) implémenté dans la plate-forme Galicia V3 (Valtchev *et al.,* 2003). Les concepts de l'ensemble E_1, respectivement de E_2, seront sous la forme suivante $(\{R_1, .., R_i\}, \{T_1, .., T_k\})$, respectivement, $(\{R_1, .., R_i\}, \{P_1, .., P_j\})$ avec $\{R_1, .., R_i\}$ un ensemble d'identifiants de requêtes, $\{T_1, .., T_k\}$ un ensemble de termes et $\{P_1, .., P_j\}$ un ensemble de pairs. Ces ensembles constituent une base $B(E_1, E_2)$ qui servira par la suite comme une base de connaissances pour l'algorithme de sélection des pairs. L'exemple de la figure 3représente les étapes de construction des concepts formels à partir du fichier log.

4.5. *Module de propagation*

A la réception d'une requête, ce module fait appel à un algorithme de sélection des pairs (LearningPeerSelection ou LPS) qui retourne un ensemble de pairs "pertinents" pour répondre à la requête. Si le nombre de pairs choisi par l'algorithme LPS est inférieur à un certain seuil nous rajoutons aux pairs sélectionnés un ensemble de pairs choisis de manière aléatoire parmi les pairs voisins (procédure $AjouterAleatoire()$ dans l'algorithme 1).

4.5.1. *Modélisation*

Le diagramme d'activité de la figure 4 modélise le processus de propagation des requêtes.

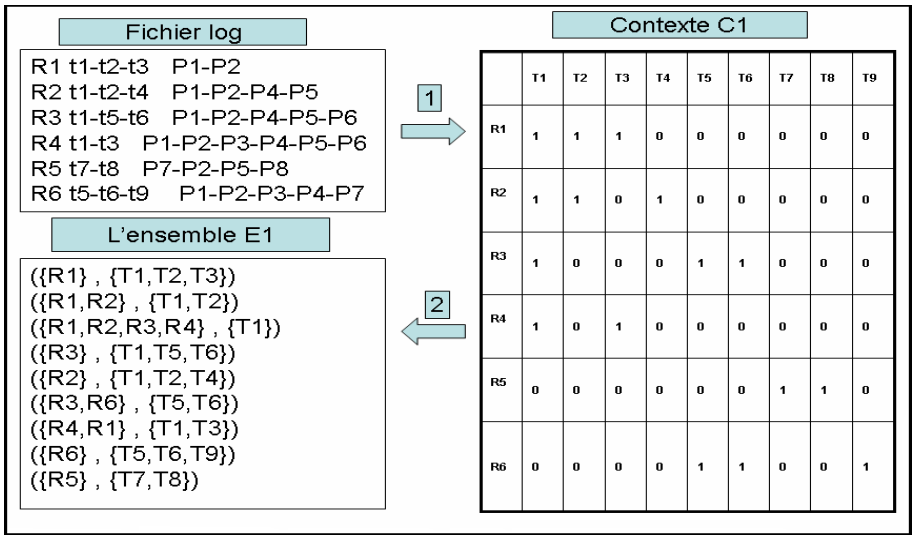


Figure 3. *Les étapes de construction des concepts formels à partir du fichier log*

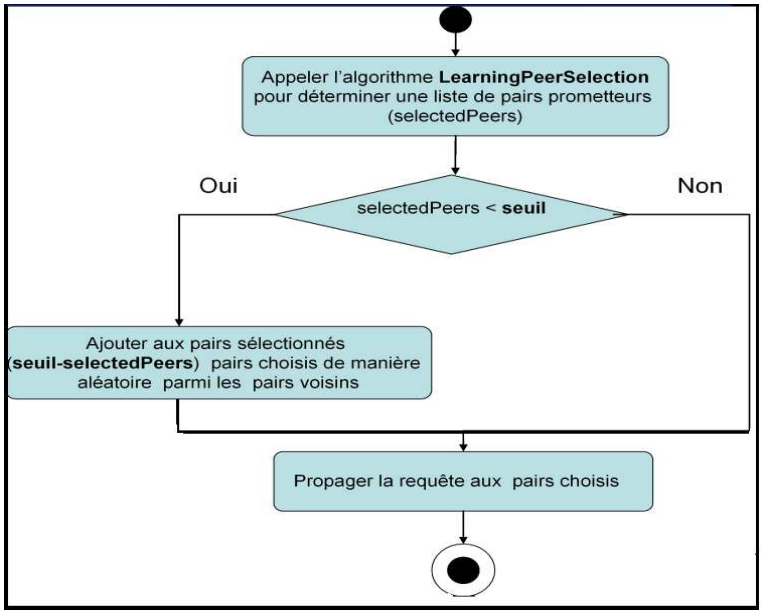


Figure 4. *Diagramme d'activité de l'algorithme de propagation*

4.5.2. *Algorithme LPS : LearningPeersSelection*

L'objectif de cet algorithme est de générer l'ensemble de concepts similaires pour une requête donnée. A partir de ces concepts, l'ensemble des pairs vers qui la requête sera routée est généré.

4.5.3. *Calcul de la similarité*

La génération des concepts similaires au concept de notre requête peut être considérée selon deux points de vue :

– Du point de vue système par exemple en utilisant le modèle vectoriel (Salton, 1989).

– Du point de vue sémantique en utilisant les propriétés communes entre deux objets (Salton, 1989).

Dans notre cas, nous avons opté pour la similarité sémantique.
Soit $k = (O, P, R)$ un contexte formel. La similarité entre deux objets a et b de O se caractérise par les propriétés communes. D'une manière formelle, cette similarité s'exprime comme suit :
Soit P_a et P_b les ensembles des propriétés respectives des objets a et b. Les propriétés communes de ces objets est l'ensemble $P_a \bigcap P_b$. La similarité se définit comme suit :

$$Sim(P_a, P_b) = \frac{|P_a \bigcap P_b|}{|P_a \bigcup P_b|}$$

4.5.4. *Présentation de l'algorithme LSP*

L'algorithme de sélection des pairs utilise la base $B(E_1, E_2)$, générée par le module de génération de profils pour choisir les pairs "pertinents". Pour une requête Q, l'algorithme détermine à partir de l'ensemble E_1 un ensemble de concepts similaires à Q (noté par $SQTC$: SimilarQueriesTermsConcepts), qui seront triés selon la valeur de similarité (fonction getSimilarConcepts() de l'algorithme 2). La similarité entre un concept $C \in E_1$ et une requête Q est calculée comme suit :

$$Sim(Q, Int(C)) = \frac{|Q \cap Int(C)|}{|Q \cup Int(C)|}$$

Où :
$Int(C)$: représente l'intention du concept C.

Par la suite, pour chaque concept C_i de l'ensemble $SQTC$, nous déterminons, à partir de l'ensemble E_2, un ensemble de concepts similaires ($SQPC$: SimilarQueriesPeersConcepts) contenant les meilleurs pairs pour la requête Q. Dans ce cas, la

similarité entre un concept $C_i \in SQTC$ et un concept $C_j \in E_2$ est calculée comme suit :

$$Sim(Ext(C_i), Ext(Cj)) = \frac{|Ext(C_i) \cap Ext(C_j)|}{|Ext(C_i) \cup Ext(C_j)|}$$

Où :
$Ext(C)$ représente l'extension du concept C.

Finalement, l'ensemble des pairs choisis est égal à l'ensemble des pairs figurants dans l'intension de chaque concept de l'ensemble $SQPC$ (fonction getSelectedPeers() de l'algorithme 2.

Algorithme : PROPAGATION $(B, Q, Lv, Pmax)$
Entrées :
B : Base de connaissances.
Q : Requête à propager.
Lv : Liste des voisins.
$Pmax$: Nombre de pairs vers lesquels la requête sera propagée.
début
    $listeFinal := \emptyset$// : Liste de pairs vers lesquels la requête sera propagée.
    $selectedPeers$:=LearningPeerSelection(B, Q) ;
    $listeFinal := selectedPeers$;
    si $selectedPeers.size()$< $Pmax$ **alors**
        $N := Pmax - selectedPeers.size()$;
        AjouterAleatoire($Lv, ListeFinal, N$) ;
    Propager($ListeFinal, Pmax$) ;
fin

Algorithme 1 : ALGORITHME DE PROPAGATION

5. Expérimentations

5.1. *Environnement*

Pour tester notre approche, nous avons choisi le simulateur PeerSim (Jelasity *et al.,* 2007), qui est un outil Opensource écrit en Java qui présente l'avantage d'être déjà spécialisé pour l'étude des systèmes P2P et qui dispose d'une architecture ouverte et modulaire qui permet de l'adapter et de lui intégrer de nouvelles couches.

Plusieurs types de simulations sont possibles, mais celle qui nous intéresse ici est la simulation par cycles : pour cela, nous laisserons évoluer le réseau pendant un nombre prédéfini de cycles. Il nous suffira alors d'analyser l'état du réseau à la fin de la

Algorithme : LEARNINGPEERSELECTION(B, Q)
Entrées :
B : Base de connaissances ;
Q : Requête ;
Sortie :
$selectedPeers$: liste des pairs sélectionnés ;//
début
  $SQPC = \emptyset$;
  $SQTC$=getSimilarConcept($B.E1, Q$) ;
  for $C \in SQTC$ **do**
    $SQPC := SQPC \cup$ getSimilarConcept($B.E2, Extent(C)$) ;
  $selectedPeers$:=getSelectedPeers($SQPC$) ;
  Return ($selectedPeers$) ;
fin

Algorithme 2 : ALGORITHME DE SÉLECTION DE PAIRS

simulation pour en tirer les conclusions. De façon à simuler et à observer tout type de réseau P2P, tous les éléments de la simulation sont paramétrables. Les données utiles au simulateur sont regroupées dans un fichier de configuration, qui définit quatre types d'éléments :

– *Les Protocoles* : Un protocole qui est une fonctionnalité associée à un noeud. En effet, dans un réseau P2P que l'on veut simuler avec PeerSim, chaque noeud dispose d'une pile de protocoles qui vont être exécutés à chaque cycle de simulation, et ceci dans leur ordre de déclaration dans le fichier de configuration.

– *Les Initializers* : Ce sont des modules lancés en début de simulation (1er cycle) pour initialiser les protocoles. Par exemple, il servent à définir la topologie initiale du réseau (associer à chaque noeud ses voisins), à distribuer le jeu de données (requêtes et documents).

– *Les Dynamics* : Leur but est d'introduire un dynamisme dans le réseau, comme par exemple des suppressions de liens entre noeuds, ou des suppressions de noeuds.

– *Les Observers* : Comme leur nom l'indique, ils ont pour but de surveiller le réseau.

5.2. *Intégration de la solution*

Comme nous l'avons déjà mentionné, le simulateur PeerSim est un environnement de simulation pour les réseaux P2P qui n'est pas adapté à un système de recherche d'information P2P. Pour cela et afin de tester notre algorithme de routage, nous devons implémenter tout un système de recherche d'information P2P qui utilise la méthode de routage proposée dans cet article. Pour contourner cette difficulté,

nous avons décidé d'utiliser le simulateur générique de systèmes P2P de recherche d'informations (Sécard *et al.,* 2006) réalisé dans le cadre du projet de recherche RARE au sein de l'Institut National de Télécommunication de Paris (RARE, 2008). Ce Simulateur est construit au dessus de PeerSim et peut être vu comme une spécialisation de PeerSim pour la recherche d'information. L'utilisation de ce simulateur nous a permis de développer plus facilement notre système puis de le comparer avec le système Gnutella. Pour mettre en oeuvre la méthode proposée, nous avons implanté un nouveau protocol, un nouveau Initializer afin d'initialiser les bases de connaissances ainsi qu'un observateur pour afficher les résultats de simulation et mettre à jour les fichiers logs.

5.3. *Les données source*

Pour tester notre algorithme LPS, nous avons utilisé le jeu de données " BigDataSet ", développé dans le cadre du projet RARE. Ce jeu de données a été obtenu à partir d'une analyse statistique sur des données collectées d'un système pair-à-pair Gnutella (Sécard *et al.,* 2006) et des données de la collection TREC (TREC, 2008), ce qui nous permet de réaliser des simulations en conditions réelles.
BigDataSet est composé de 25 000 documents et 4999 requêtes qui est réparti avec duplication sur 499 pairs. En effet, quelques requêtes et documents sont répliqués. Ce jeu de données fournit des fichiers XML décrivant les noeuds du système et les documents qu'ils possèdent, ainsi que les requêtes qui seront lancées sur le réseau.

5.4. *Mesures d'évaluation*

Pour tester les performances de notre approche, nous avons utilisé les métriques rappel (R) et le nombre de messages, définies comme suit pour une requête Q et sachant que DPR est le nombre de documents pertinents retournés et DP est le nombre de documents pertinents

– *R(Q)* = $\frac{DPR}{DP}$

– *Nombre de messages* = nombre de messages échangés pour répondre à la requête Q

5.5. *Initialisation des paramètres de simulation*

La simulation de notre algorithme ainsi que celui utilisé par Gnutella s'est basée sur les paramètres suivants :

– *TTL* : Profondeur maximale de recherche, initialisée à 5.

– *Pmax* : Nombre de pairs auxquels la requête doit être propagée.

– *Overlay size* : Nombre de pairs dans le réseau, initialisé à 500 (nombre de pairs dans le jeu de données).

De plus notre algorithme a besoin d'une base de connaissances pour chaque pair. Pour cela, nous avons lancé les 20600 premières requêtes, en utilisant l'algorithme de Gnutella, afin de construire un fichier log initial pour chaque pair. Par la suite, nous avons lancé l'exécution du module de gestion de profils pour construire une base de connaissances pour chaque pair à partir de son fichier log, noté $B1$. Pour prendre en compte les nouvelles informations sur les requêtes passées, les bases des connaissances sont mises à jour après l'émission d'un nombre de requêtes. Le tableau 2 décrit les bases générées selon les caractéristiques suivantes :

– *Nombre de requêtes* : Nombre de requêtes émises par les différents pairs du système. Ce nombre, pour une base B_i est égal au nombre de requêtes relatives à une base B_{i-1} plus le nombre de nouvelles requêtes.

– *Taille moyenne* : C'est la somme des tailles des bases divisée par le nombre de pairs.

Nom de la base	# requêtes	Taille moyenne de la base (en Mo)
B1	20652	0.046
B2	28927	0.071
B3	34732	0.092

Tableau 2. *Caractéristiques des bases de connaissances de test*

5.6. *Résultats d'expérimentation*

Pour comparer les performances de notre algorithme par rapport à celui de Gnutulla nous avons calculé le rappel moyen et la moyenne de nombre de messages par intervalle de 2000 requêtes envoyées par les différents pairs du système. La figure 5, montre que le rappel moyen pour notre algorithme, varie entre 0.82 et 0.86 alors que le rappel moyen pour Gnutella varie entre 0.43 et 0.44. De plus, le rappel pour notre algorithme augmente à chaque enrichissement des bases de connaissances. La figure 6 montre que la moyenne de nombre de messages pour notre algorithme a diminué de 283 en utilisant la base initiale $B1$ à 262 en utilisant la base $B2$ et à 259 en utilisant $B3$. Par contre il reste stable pour Gnutella avec une valeur égale à 313. A partir de ces deux figures, nous pouvons déduire que la taille de la base de connaissances a un impact sur la qualité de routage de notre algorithme.

6. Conclusion

De nombreux travaux de recherche proposent d'exploiter la sémantique sur le contenu des pairs pour router plus efficacement les requêtes, mais peu de travaux

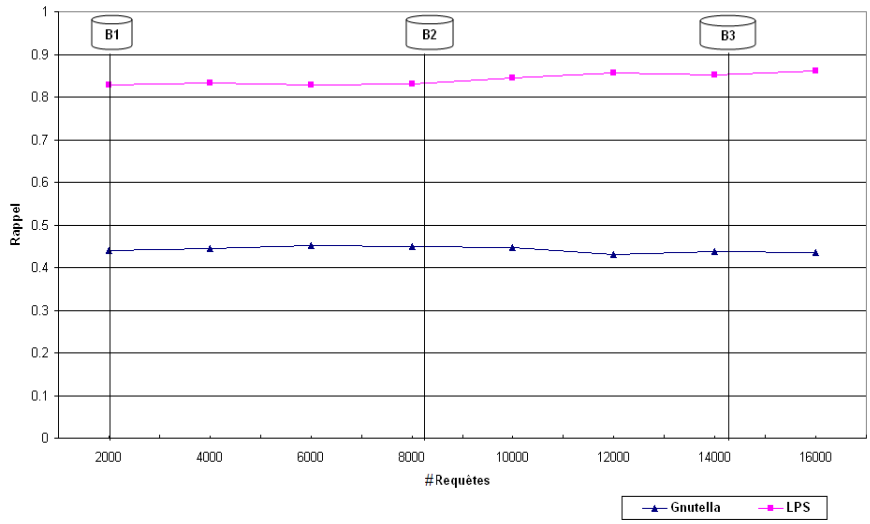


Figure 5. *Rappel en fonction du nombre de requêtes*

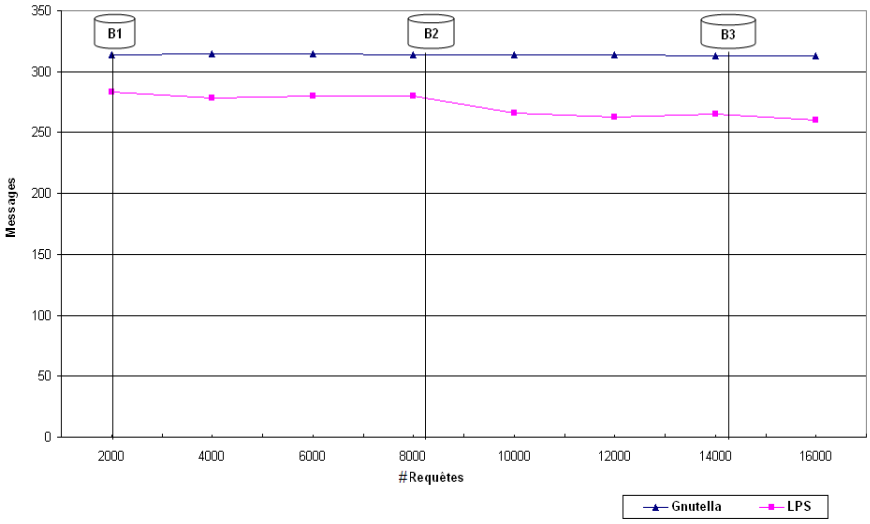


Figure 6. *Nombre de messages en fonction du nombre de requêtes*

explorent les informations relatives à l'historique des requêtes. Ainsi, nous avons proposé un algorithme de routage des requêtes par apprentissage basé sur les profils des utilisateurs qui sont déduits à partir des historiques de requêtes. Pour construire ces profils, nous avons utilisé une approche formelle basée sur l'Analyse Formelle de Concepts. Des tests, réalisés en utilisant le simulateur PeerSim, ont montré que notre algorithme de routage est plus performant qu'un algorithme de routage classique, en terme de rappel et de nombre de messages, et que la qualité de routage de notre algorithme dépend de la taille de la base de connaissances. Il nous reste encore à définir une stratégie de maintenance des bases de connaissances et à étudier plus finement son impact sur la qualité de routage.

7. Bibliographie

Callan J. P., Lu Z., Croft W. B., « Searching Distributed Collections with Inference Networks », *In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM Press, p. 21-28, 1995.

Chernov S., Serdyukov P., Bender M., Michel S., Weikum G., Zimmer C., « Database selection and result merging in P2P web search », *In DBISP2P 2005*, 2005.

Christoph T., Steffen S., Adrian W., « Semantic Query Routing in Peer-to-Peer Networks based on Social Metaphors », *13th International World Wide Web Conference, May 17-22, New York City, NY*, 2004.

Defude B., « Organisation et routage sémantiques dans les systèmes pair-à-pair », *Actes du XXVème Congrès INFORSID, May 22-25, Perros-Guirec, France*, p. 12-8, 2007.

Ganter B., Wille R., *Formal Concept Analysis : Mathematical Foundations*, Springer-Verlag New York, Inc, 1997.

Gnutella, « Gnutella website », `http ://www.gnutella.com/`, January, 2007.

Godin R., Missaoui R., Alaoui H., « Incremental concept formation algorithms based on galois (concept) lattices », *Computational Intelligence*, vol. 11(2), p. 246-267, 1995.

Jelasity M., Montresor A., Jesi G. P., Voulgaris S., « The Peersim Simulator », `http ://peersim.sf.net`, 2007.

Luis G., Hector G., Anthony T., « The effectiveness of gloss for the text database discovery problem », *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, Minneapolis, Minnesota, May 24-27*, ACM Press, p. 24- 27, 1994.

Luis G., Hector G.-M., « Generalizing gloss to vector-space databases and broker hierarchies », *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, Morgan Kaufmann, p. 78-89, 1995.

Raja C., Bruno D., Georges H., « Définition et diffusion de signatures sémantiques dans les systèmes pair-à-pair », *Extraction et gestion des connaissances (EGC'2006), Actes des sixièmes journées Extraction et Gestion des Connaissances, Lille, France, 17-20 janvier 2006, 2 Volumes*, vol. RNTI-E-6 of *Revue des Nouvelles Technologies de l'Information*, p. 463-468, 2006.

RARE, « Le projet RARE (Routage optimisé par Apprentissage de REquêtes) », `http ://www-inf.int-evry.fr/defude/RARE/`, 2008.

Salton G., « Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer », *Addison Wesley*, 1989.

Sécard J., Bruno D., Chiky R., Georges H., Sorin M., « Un simulateur générique pour les systèmes de recherche d'informations en pair à pair », *3ème conférence en Recherche d'Informations et ses Applications, Lyon March 15-17*, 2006.

TREC, « Text REtrival Conference », `http ://trec.nist.gov/`, 2008.

Valtchev P., Grosser D., Roume C., Hacene M. R., « Galicia : an open platform for lattices », *In Using Conceptual Structures : Contributions to the 11th Intl. Conference on Conceptual Structures (ICCS'03*, Shaker Verlag, p. 241-254, 2003.

Chapitre 4

Recherche d'Information Sémantique

Indexation semi-automatique de textes : thésaurus et transducteurs

Laurent Kevers

Centre de traitement automatique du langage (CENTAL)
Université catholique de Louvain (UCL)
Collège Erasme - Place Blaise Pascal, 1
1348 Louvain-la-Neuve - Belgium
laurent.kevers@uclouvain.be

RÉSUMÉ. *Cet article présente une méthode de classification ne nécessitant pas de phase d'apprentissage. Son but est d'améliorer l'indexation manuelle des documents textuels, une opération souvent menée au sein de certains systèmes d'information requérant un niveau de précision élevé. Le système, qui apporte une aide à l'indexeur humain, est semi-automatique. Par analogie à la terminologie utilisée en apprentissage automatique, la méthode est dite supervisée car elle exploite une définition préalable des catégories d'indexation. Un vocabulaire contrôlé, par exemple un thésaurus, est utilisé comme la ressource de base servant à la génération automatique de transducteurs (ou automates). L'application de ceux-ci à un texte permet d'extraire un nombre limité d'expressions pertinentes, chacune accompagnée d'au moins un code de catégorie dont l'analyse finale permet la classification du document. Nos tests sur un corpus de textes en français ont permis d'obtenir une f-measure située entre 0,51 et 0,64.*

ABSTRACT. *This article presents a classification method without any learning stage. It can help to improve the manual indexation process of textual documents traditionally conducted in some high precision information systems. The described system is defined as semi-automatic as it will help the human indexing. By analogy with machine learning terminology, this method can be qualified as supervised as it uses a priori defined indexing categories. A controlled vocabulary, e.g. a thesaurus, is used as the main resource to automatically generate a set of transducers (or automata). The extraction of a document's significant phrases, each one comming with at least one corresponding class code, is obtained when using these transducers on the text. The final classification is obtained after analysis of phrases and codes. Testing results on a french text corpus are comprised between 0.51 and 0.64 for f-measure.*

MOTS-CLÉS : *catégorisation, classification supervisée, indexation, thésaurus, transducteur*

KEYWORDS: *categorization, supervised classification, indexing, thesaurus, transducer*

1. Introduction

Traditionnellement, l'indexation de textes au sein de grandes bases de données documentaires est réalisée à l'aide de mots clés, souvent issus d'un vocabulaire contrôlé. Cette indexation s'effectue généralement manuellement, ce qui apporte une grande précision au système. La cohérence de l'indexation peut cependant être diminuée par la variabilité des décisions au cours du temps et en fonction des différents indexeurs. D'autre part, la quantité de documents à traiter augmente rapidement et rend le processus manuel inabordable pour beaucoup d'organisations. Il est dès lors souvent remplacé par des méthodes d'indexation automatiques, moins précises mais beaucoup plus rapides et cohérentes. Si ces techniques sont adéquates pour des applications *temps réel* ou lorsqu'un très grand nombre de documents doit être analysé, elles ne constituent pas la solution idéale dans le contexte de systèmes d'information nécessitant une précision élevée.

Nous décrivons une méthode ayant une visée applicative réelle et concrète. Le but est d'améliorer l'efficacité et la cohérence de l'indexation manuelle en suggérant à l'indexeur une liste de catégories ou de mots clés potentiels automatiquement construite. L'approche adoptée est centrée sur l'utilisation d'une ressource de base - un thésaurus définissant les catégories - pour générer automatiquement des automates de reconnaissance, ou plus précisément des transducteurs (*cf.* section 4.3). Ceux-ci permettent d'exprimer des patrons contraints dépassant de loin les possibilités de la simple expression régulière (*cf.* section 6.1). L'application des transducteurs à un texte permet d'extraire un nombre limité d'expressions *pertinentes*, chacune accompagnée d'au moins un code de catégorie dont l'analyse finale permet la classification du document (*cf.* sections 6.2 à 6.4). L'indexation humaine se résume alors à la consultation sommaire du document - titre, résumé et éventuellement premier paragraphe - et à la sélection d'un ou plusieurs éléments dans la liste proposée et non plus dans le thésaurus complet. Afin de conserver ce gain de temps, il est évidemment très important que l'indexeur n'ait pas à consulter le texte en entier pour faire son choix. Par conséquent, la liste doit contenir un maximum de mots clés *plausibles* quitte à y inclure certaines propositions non pertinentes. Cette méthode semi-automatique à l'avantage d'améliorer la rapidité et la cohérence de l'indexation grâce à l'analyse automatique tout en conservant une précision élevée garantie par la validation humaine. L'approche, sans apprentissage, ne nécessite pas de données annotées manuellement et est fonctionnelle dès le premier document. Cette caractéristique ouvre des perspectives en terme d'interaction avec d'autres méthodes requérant un apprentissage.

Cet article introduit quelques concepts sur la classification et les thésaurus. Les hypothèses, les prérequis et les choix d'implémentation sont ensuite présentés. Ils sont suivis par un aperçu des travaux précédemment menés sur cette thématique. L'analyse de texte et la classification sont ensuite détaillées, avant de clôturer avec les résultats, perspectives et conclusions.

2. Classification

L'attribution d'index issus d'un vocabulaire contrôlé à un document est comparable au processus de classification de textes, domaine largement couvert par les techniques d'apprentissage automatique. Bien que notre méthode n'en fasse pas partie, nous allons brièvement en présenter la terminologie afin de pouvoir nous situer par rapport à ces travaux.

La classification de textes peut être divisée en deux activités distinctes : le clustering et la catégorisation. Le *clustering* regroupe les documents en ensembles de textes similaires sur la base du seul contenu de ces documents. La *catégorisation* exploite une ressource extérieure, les catégories à attribuer. Le clustering et la catégorisation sont aussi qualifiées respectivement de classification *non supervisée* et *supervisée*. Notre système s'apparente à la classification *supervisée* que nous appelerons désormais indifféremment classification ou catégorisation. Dans notre cas, le résultat attendu est un liste de catégories accompagnées de poids, ce qui serait qualifié de classification *soft* et *multi labels* en apprentissage automatique.

3. Thésaurus

Nous avons mentionné dans la section précédente que la classification s'appuie sur une ressource qui définit l'ensemble des catégories attribuables. Cette ressource peut être aussi simple qu'une liste de termes ou prendre la forme plus élaborée d'une ontologie ou d'un thésaurus. C'est souvent ce dernier qui est utilisé car il représente un bon compromis entre puissance descriptive et complexité acceptable du point de vue du développement et de la maintenance.

Un thésaurus est un vocabulaire contrôlé qui regroupe un ensemble de concepts relatifs à un certain domaine. Il constitue un moyen de décrire ce domaine, d'en définir les concepts et de fixer la terminologie utilisée par un groupe de personnes. Un concept est représenté par un terme principal appelé *descripteur* qui peut être relié à plusieurs *non descripteurs* ou *synonymes* par une relation *used-for* (UF). Les concepts sont organisés hiérarchiquement à l'aide des relations *broader-than* (BT) et *narrower-than* (NT). La relation *related-term* (RT) permet de définir un lien de similarité entre deux concepts. Les grands thésaurus peuvent être fragmentés en *microthésaurus* couvrant chacun un sous thème particulier. Plusieurs normes internationales dont (ISO, 1986) et (AFNOR, 1981) définissent plus précisément les thésaurus.

La portée d'un thésaurus peut être très large, par exemple Eurovoc[1], ou au contraire très spécialisée, tel que Agrovoc[2]. Ces deux thésaurus comptent un grand nombre de

1. Thésaurus du Parlement de la Communauté européenne, couvre une grande diversité de domaines, mais toujours en rapport avec le travail parlementaire : http ://europa.eu/eurovoc/
2. Thésaurus de l'Organisationn des Nations Unies pour l'alimentation et l'agriculture, se concentre sur l'agriculture : http ://www.fao.org/aims/ag_intro.htm

niveaux hiérarchiques et de descripteurs[3]. De nombreuses organisations se contentent cependant de vocabulaires de tailles plus modestes. Van Slype (Van Slype, 1987) préconise l'usage de 500 à 1500 descripteurs pour des bases de données ayant un accroissement de 10.000 documents par an et de 3000 à 6000 descripteurs si la base s'étend jusqu'à 100.000 documents par an.

De nombreuses applications en traitement automatique du langage peuvent tirer parti d'une ressource telle qu'un thésaurus (Da Sylva, 2006). Par exemple, l'expansion de requêtes pour les moteurs de recherche, la désambiguisation lexicale ou encore la traduction automatique. En ce qui nous concerne, nous allons utiliser le thésaurus comme base du processus de classification : chaque descripteur, identifié par son code et accompagné par ses synonymes, constituera une catégorie.

4. Hypothèses, prérequis et choix d'implémentation

4.1. *Hypothèses*

Nous pensons que l'appartenance d'un texte à une catégorie thématique se matérialise dans le document par l'utilisation d'un certain nombre de mots. Dès lors, si les catégories sont correctement définies (elles doivent posséder un terme descripteur principal et de préférence autant de non-descripteurs qu'il existe de synonymes), il est possible de trouver une intersection suffisante entre le vocabulaire du document et la définition des catégories à sélectionner pour décider de manière automatique de leur assignation.

Nous adhérons également au principe qu'une expression composée a généralement un sens très précis et constitue souvent un bon candidat en tant que concept descripteur du document. L'observation du lexique d'une langue telle que le français nous montre que les concepts complexes sont souvent exprimés à l'aide d'expressions composées. On utilise par exemple rarement le terme *allocations* seul, mais plutôt dans une forme composée telle que *allocations de chômage* ou *allocations familliale*. De plus, les expressions composées sont souvent moins polysémiques (Yarowsky, 1993). Par conséquent, notre système ne doit pas se limiter aux mots simples et doit prendre en compte les unités polylexicales.

A partir de ces hypothèses, nous voulons montrer qu'il est possible de mettre en œuvre une analyse performante ne requérant pas de phase d'apprentissage pour la classification/indexation de textes. Cette analyse est basée sur des principes simples et utilise une ressource lexicale et sémantique telle qu'un thésaurus. Celui-ci doit être de qualité, c'est-à-dire que les descripteurs soient correctement organisés, qu'ils soient au moins accompagnés de leurs synonymes les plus courants et qu'ils ne soient pas trop abstraits.

3. Eurovoc : 6,645 pour chaque langue ; Agrovoc : 28,718 en anglais uniquement

4.2. *Prérequis*

L'implémentation fait appel à diverses techniques de traitement automatique du langage exploitant des ressources linguistiques. Le principal prérequis pour l'utilisation de cette méthode est l'existence d'une description des catégories utilisées pour l'indexation, tel qu'un thésaurus. Cette limitation est peu contraignante en pratique puisque de nombreuses organisations utilisent ce genre de ressource.

4.3. *Choix d'implémentation : extraction ou assignation de mots clés*

La première approche, l'*extraction de mots clés*, part du texte. Elle consiste à extraire des mots simples ou composés porteurs de sens. L'extraction s'appuie sur des critères lexicaux et grammaticaux. L'appartenance au thésaurus des termes extraits d'un texte permet de dériver les catégories potentiellement attribuables à celui-ci. Beaucoup de ces termes sont cependant comparés sans succès au thésaurus.

La seconde approche, l'*assignation de mots clés*, démarre du thésaurus et consiste à créer une ressource d'extraction à l'aide des termes descripteurs, accompagnés de leur synonymes. Chaque descripteur correspond à une catégorie. Cette ressource est appliquée aux textes afin de retrouver un maximum d'expressions dites *pertinentes*, c'est-à-dire ayant *a priori* un intérêt pour la classification car dérivées du thésaurus, en délaissant les expressions ne possédant *a priori* pas de pouvoir classifiant.

Pour extraire les candidats, ne retenir que ceux qui sont potentiellement intéressants et enfin les confronter au thésaurus, l'extraction de mots clés nécessite un grand nombre d'heuristiques et de règles, difficiles à développer et à maintenir. Au contraire, l'assignation réalisée à partir d'une ressource dérivée du thésaurus constitue une méthode plus simple. Elle se limite à analyser un ensemble restreint d'expressions *pertinentes*, ce qui est plus efficace. Nous avons donc choisi d'utiliser la seconde approche.

La ressource d'extraction est constituée par des transducteurs et est capable de reconnaître des unités formées d'un nombre variable et non limité de tokens. Le formalisme des automates et transducteurs a été choisi car il dépasse de loin la simple recherche *mot à mot*. Il est utilisé pour un grand nombre d'analyses effectuées en traitement automatique du langage (Crochemore *et al.*, 1994). Il nous permet d'exprimer le contenu du thésaurus au moyen d'expressions régulières complexes, de codes grammaticaux et sémantiques[4], d'y insérer certaines contraintes sur le contexte lexico-syntaxique ou encore d'autoriser des insertions facultatives à certains endroits. La manière dont ces possibilités sont exploitées pour adapter le thésaurus en une ressource d'extraction la plus performante possible est exposée plus en détail à la section 6.1 et un exemple de graphe[5] est présenté à la figure 1. L'utilisation des transducteurs

4. Après application des dictionnaires électroniques.
5. Les transducteurs sont représentés graphiquement sous la forme de graphes. Un graphe contient toujours un état initial et un état final et peut être accompagné d'un ensemble de chemins concurents constitués de transitions étiquettées ainsi que des sorties (ou transductions).

sur un texte résulte en une liste d'expressions, chacune accompagnée d'une indication d'appartenance à une ou plusieurs catégories du thésaurus. La pondération de ces expressions permet ensuite de trier les résultats présentés à l'indexeur humain.

5. Travaux apparentés

Comme nous l'avons déjà mentionné, la classification (semi-) automatique est un domaine dans lequel les techniques d'apprentissage automatique sont souvent appliquées. Ces techniques ne nous intéressent pas directement dans le cadre de ce travail. Le lecteur intéressé pourra en trouver une introduction en consultant une référence telle que (Sebastiani, 2002), (Moens, 2006), ou encore (Baeza-Yates *et al.*, 1999).

Bien que l'usage d'un thésaurus pour améliorer ou guider la classification ne soit pas l'approche la plus répandue, certains travaux y sont cependant apparentés. KEA++ (Medelyan *et al.*, 2006) est un système agissant en deux phases : l'extraction de mots clés et leur filtrage à l'aide d'un thésaurus sont suivis par une étape d'apprentissage capable de mettre en œuvre différents types d'algorithmes. Pouliquen *et al.* (Pouliquen *et al.*, 2006) présentent une méthode statistique et associative de classification de documents dans Eurovoc qui se base sur différentes mesures de similarité. Cette étude a été menée sur diverses langues dont l'anglais, l'espagnol et le français. Névéol *et al.* (Névéol *et al.*, 2005) évaluent deux systèmes hybrides - MTI pour l'anglais, MAIF pour le français - d'indexation de documents médicaux dans MeSH[6]. Ces systèmes combinent à la fois une approche de traitement automatique du langage de type *sac de mots* et une approche plus statistique (*PubMed Related Citations* pour l'anglais, une mesure de similarité exploitant la méthode *k-Nearest Neighbour* pour le français). Toujours dans le domaine médical et en français, Pereira *et al.* (Pereira *et al.*, 2008) étudient avec F-MTI les possibilités d'assignation de descripteurs MeSH à l'aide de plusieurs terminologies. La technique d'analyse repose à nouveau sur un algorithme de *sac de mots*. Enfin, c'est Névéol *et al.* (Névéol *et al.*, 2006) qui s'approche le plus de notre méthode. Ce système d'indexation de documents en français repose sur MeSH et exploite une série de transducteurs. Il existe cependant une différence de taille puisque ceux-ci sont construits manuellement en collaboration avec des experts alors que nous proposons de les générer automatiquement.

6. Analyse du texte et classification

Notre système de classification nécessite la production d'une version du thésaurus sous la forme de transducteurs, mieux adaptés au traitement automatique des textes (*cf.* section 6.1). Cette opération est unique et ne fait pas à proprement parler du processus de classification répété pour chaque document. Elle se doit d'être complètement automatique en raison du nombre élevé d'éléments que peut contenir un thésaurus.

6. Medical Subject Headings : http ://www.nlm.nih.gov/mesh/

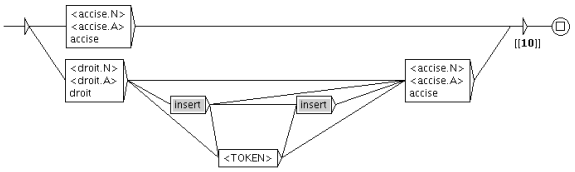


Figure 1. *Transducteur lemmatisé (descripteur : accise ; synonyme : droit d'accise ; catégorie de code 10).*

Une fois cette ressource à disposition, son application aux textes (*cf.* section 6.2) donne une liste d'expressions et de termes *pertinents*, accompagnés des catégories auxquels ils sont reliés. Pour chaque expression, une pondération est calculée selon plusieurs critères (*cf.* section 6.3). Les poids sont ensuite additionnés de manière à obtenir une valeur globale pour chaque catégorie représentée dans le texte. Cette liste pondérée permet finalement de sélectionner les catégories les plus significatives pour le document (*cf.* section 6.4).

6.1. *Du thésaurus aux transducteurs*

Les transducteurs sont automatiquement générés à partir du thésaurus dans un format compatible avec le logiciel de traitement de corpus Unitex[7] (Paumier, 2008). Chaque catégorie est représentée par un automate contenant une transduction qui renseigne le code de la catégorie. Les transducteurs générés sont rassemblés en un transducteur principal. La figure 1 illustre un transducteur généré automatiquement.

Nous avons mené deux types d'expériences différentes. Pour la première, les catégories sont réduites aux grandes divisions matérialisées par les microthésaurus. Un transducteur contient l'ensemble des descripteurs et synonymes reliés à un microthésaurus particulier. Ce regroupement constitue une sorte de généralisation ou simplification qui permet de réduire le nombre de catégories. La seconde expérience, conserve toutes les catégories. Un transducteur est donc généré par descripteur. Dans ce cas nous obtenons de petits transducteurs en très grand nombre alors que le premier cas débouche sur des transducteurs plus volumineux mais en nombre plus restreint. Les paragraphes suivants détaillent les quatre principaux traitements nécessaires à la production des patrons qui seront ensuite retranscrits sous la forme de transducteurs.

Le premier traitement est une étape de généralisation dont le but est d'étendre la couverture aux variations possibles d'une expression, telle que le passage du singulier au pluriel. Par exemple, à partir de l'expression *taux d'intérêt légal* issue du thésaurus, nous désirons aussi retrouver les formes *taux d'intérêt**s** légal*, *taux d'intérêt lég**aux***, ou

7. http ://www-igm.univ-mlv.fr/ unitex/

encore *taux d'intérêt**s*** *lég**aux***[8]. Deux techniques permettent d'atteindre ce résultat : le stemming et la lemmatisation. Le *stemming* consiste en l'extraction d'un préfixe correspondant à la racine d'un mot. Nous avons utilisé l'implémentation Snowball[9] de l'algorithme de Porter (Porter, 1997). Cette approche produit des transducteurs principalement composés d'expressions régulières. La *lemmatisation* permet de relier une forme fléchie à sa forme canonique (ou lemme). Les patrons générés font alors directement référence aux lemmes et non plus à des formes fléchies, ce qui permet de tirer parti de la puissance des dictionnaires électroniques disponibles dans Unitex. Pour obtenir le lemme, nous avons utilisé Treetagger[10], un étiqueteur morpho-syntaxique multilingue (Schmid, 1994). C'est cette dernière approche qui a finalement été choisie en raison des meilleurs résultats préliminaires obtenus. De plus, l'utilisation d'Unitex nous a permis de constater que le temps d'exécution des transducteurs constitués d'expressions régulières est de loin plus élevé que celui obtenu avec les transducteurs lemmatisés.

Le deuxième traitement consiste, comme dans de nombreux travaux en recherche d'information, à éliminer les *stopwords* (*mots vides*). Ils ont en réalité été remplacés par une méta-étiquette (`<TOKEN>`). Cela permet d'améliorer la reconnaissance d'expressions dans lesquelles un mot peut être remplacé par un autre. C'est par exemple le cas dans *contrôle **de** chômeurs*, *contrôle **du** chômeur* et *contrôle **des** chômeurs*[11].

Le troisième traitement apporté est la possibilité d'insertion, entre chaque mot, d'un terme facultatif. Cette extension permet d'aller plus loin dans la reconnaissance d'expressions similaires. En effet, il est assez courant qu'une expression possède une forme complète et une forme simplifiée ou qu'elle soit modulée par des adjectifs. C'est par exemple le cas pour *agence de protection de l'environnement* qui peut être retrouvé sous la forme d'*agence **<ADJ>** de proctection de l'environnement*, avec <ADJ> tel que *fédérale*, *régionale*, *belge*...

Enfin, le quatrième traitement concerne certaines exceptions qui doivent être prises en compte. Par exemple l'acronyme *CAS* qui correspond à *Caisse d'allocations sociales* est ambigu avec le nom *cas*. La casse ne peut être prise en compte car certains thésaurus sont encodés complètement en majuscules, ce qui empêche l'exploitation de ce critère. Insérer cet acronyme tel quel conduirait à reconnaitre sa présence à chaque occurence du nom commun, ce qui fausserait fortement l'analyse. Dans ce cas, nous nous limitons donc à utiliser l'acronyme dans sa version avec points (*C.A.S.*). Autre exception, suite au traitement des *stopwords*, certains patrons se résumeraient normalement à une seul méta-étiquette telle que `<TOKEN>`. Comme cela conduirait à la reconnaissance de toutes les unités du texte, nous gardons alors la forme d'origine lemmatisée avec la restriction supplémentaire de la présence d'un déterminant.

---

8. De telles ressources ont une tendance à la surgénération. Dans un contexte de reconnaissance, cela ne représente pas un problème. Au contraire, cela permet de reconnaître les occurences mal orthographiées ou s'éloignant de la norme.
9. http ://snowball.tartarus.org/
10. http ://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
11. Ce cas n'est pas couvert par la lemmatisation car le lemme de *des* est *un*

Forme	Thésaurus	Substitut
art.	ART (arts plastiques)	*article*
livre (section de texte)	LIVRE (ouvrage)	*livreSection*
titre (section de texte)	TITRE (finance)	*titreSection*
au titre de ...	TITRE (finance)	
à titre ...	TITRE (finance)	
à juste titre	TITRE (finance)	
j'ai (nous avons) l'honneur de ...	HONNEUR	*je*

Tableau 1. *Normalisation des mots ambigus d'un texte.*

6.2. *Application des transducteurs aux documents et catégorisation*

Avant l'application des transducteurs générés à partir du thésaurus aux documents, une étape de prétraitement des textes est nécessaire. Lors de la création des transducteurs et du traitement des *stopwords*, les formes élidées telles que *l'* ont été remplacée par une méta-étiquette, par exemple `<TOKEN>`. Or Unitex crée, pour cette forme *l'*, deux tokens : *l* et *'*. Cette forme n'est donc plus reconnaissable par une seule étiquette `<TOKEN>` mais bien par deux. Afin d'éviter ce problème, un transducteur de prétraitement remplace toutes les formes élidées par une forme complète correspondante, par exemple *le* pour *l'*.

Une procédure de désambiguïsation ciblée est également souhaitable afin d'éviter certaines erreurs récurentes. L'expression *art. 2* (article 2) est par exemple interprétée comme reliée à la catégorie `ART` (arts plastiques, etc.) du thésaurus. D'autres cas sont repris au tableau 1. L'idéal est de réaliser une étude exhaustive des termes posant un problème d'ambiguïté dans le thésaurus. Evidemment, cette tâche n'est pas complètement automatisable et est spécifique à un thésaurus et à une langue en particulier. Afin de minimiser l'effort nécessaire, on peut cependant envisager de mener cette étude lors de la construction même du thésaurus, qui mobilise de toutes façons les compétences de spécialistes en terminologie. Pour les thésaurus existants, il est nécessaire de mettre au point une méthode de détection de la polysémie permettant de repérer les cas problématiques et requérant une intervention. Ce point fait partie des développements futurs que nous envisageons.

Finalement, d'autres tâches de prétraitement plus classiques sont réalisées. Le texte est désaccentué, car certains thésaurus sont complètement capitalisés et non accentués, ce qui est le cas de celui utilisé pour nos expériences. La suite du processus est réalisé au moyen d'Unitex : tokenisation, application des dictionnaires et application des transducteurs issus du thésaurus. Le résultat obtenu se présente sous la forme d'un index de mots ou d'expressions tel qu'illustré à la figure 2[12].

12. Les deux premières colonnes indiquent les numéros des tokens délimitant l'expression

0 12 @000101024.xml@	193 193 batiments[[MT191]]
14 16 <title>	235 235 controlees[[MT992]]
53 53 aeroport[[MT111]]	264 270 personnel de le aeroport[[MT111]]
57 57 bruxelles[[MT991]]	274 274 bruxelles[[MT991]]
60 63 </title>	295 295 ministre[[MT124]]
77 77 president[[MT157]]	299 299 transports[[MT111]]
113 113 ministre[[MT124]]	348 348 aeroport[[MT111]]
117 117 transports[[MT111]]	356 356 livre[[MT133]]
124 124 armee[[MT122]]	360 360 marchandises[[MT192]]
124 124 armee[[MT102]]	385 385 ministre[[MT124]]
140 140 aeroport[[MT111]]	420 420 president[[MT157]]
144 144 bruxelles[[MT991]]	446 446 deputee[[MT124]]

Figure 2. *Liste de mots ou d'expressions retrouvées à l'aide des transducteurs pour un texte. Le code de catégorie (ici des microthésaurus) est inclus entre crochets.*

6.3. *Pondération*

Sur la base de la liste construite après application des transducteurs au texte (*cf.* Figure 2), un poids est calculé pour chaque expression et ensuite globalement pour chaque catégorie. Cette pondération est basée sur une mesure de fréquence mais d'autres critères sont aussi pris en compte. Ils sont implémentés par des multiplicateurs appliqués au poids initial. La recherche de leurs valeurs optimales a été effectuée empiriquement avec un nombre limité de valeurs[13].

La valeur de base pour la pondération est TF.IDF (*term frequency-inverse document frequency*). Cette mesure est couramment utilisée pour évaluer le poids d'un terme par rapport à un corpus donné. Ce score de base sera éventuellement modifié pour déterminer le score final d'une expression. Les formules appliquées sont :

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \qquad idf_i = log\frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

où n_{ij} est la fréquence d'un terme i dans le document d_j, $|D|$ étant le nombre de documents dans le corpus et $|\{d_j : t_i \in d_j\}|$ le nombre de documents dans lesquels le terme i est présent. La valeur finale du TF.IDF est obtenue par : $tf.idf_{ij} = tf_{ij} * idf_i$ Le but de cette mesure est de donner plus d'importance aux mots très fréquents dans un document, mais rares à l'échelle du corpus. Chaque expression de la liste obtient donc un poids TF.IDF. Les valeurs IDF sont précalculées sur le corpus en appliquant les transducteurs de reconnaissance issus du thésaurus. Cette méthode peut être perçue comme un biais, mais il s'agit d'une approximation raisonnable des scores IDF qui seraient graduellement construits lors du traitement des mêmes documents en situation réelle.

Le deuxième critère est basé sur le fait que les informations importantes apparaissent souvent au début du document, c'est-à-dire principalement dans le titre et le résumé s'il existe. Nous avons donc introduit un multiplicateur qui est appliqué au

13. Pour chaque multiplicateur, les valeurs testées sont 1, 2, 5, 10, 20, 50 et 100. Toutes les combinaisons ont été testées.

score de base (TF.IDF) si l'expression se situe dans le titre. Pour nos expériences, ce multiplicateur a été fixé empiriquement à 100.

Le troisième critère exploite l'intérêt particulier soulevé pour les expressions composées. Bien que cette caractéristique soit déjà indirectement prise en compte dans la mesure du TF.IDF, nous avons prévu un multiplicateur supplémentaire pour en augmenter le score. Sa valeur, 2, a été fixée de manière empirique.

Le quatrième et dernier critère envisagé concerne les entités nommées. Celles-ci sont détectées à l'aide de transducteurs spécifiques lors de la phase de prétraitement du texte. Le multiplicateur relié à ce critère a été fixé empiriquement à 2.

Après l'application de ces multiplicateurs, un score final est calculé pour chaque catégorie représentée dans le document afin d'obtenir une liste ordonnée des meilleurs propositions.

6.4. *Définition d'un seuil de sélection*

La liste pondérée obtenue peut être assez longue et les différences de poids importantes. Nous désirons donc réduire cette liste afin de ne garder que les candidats les plus probables. Cette sélection est opérée au moyen d'un seuil.

La première méthode (*k-first*) consiste simplement à conserver les k premières catégories correspondant aux meilleurs scores. Deux autres méthodes consistent à calculer la moyenne arithmétique des poids de catégories (*averaged pivot*) ou la moitié du score maximal (*middle pivot*). Ces valeurs centrales constituent un premier seuil. A partir de celui-ci, d'autres valeurs plus grandes ou plus petites sont ensuite obtenues par sauts de taille fixe. Pour obtenir x seuils plus élevés, on augmente ainsi x fois la valeur centrale de $\frac{val.max. - val.centrale}{x}$. De même, on diminue à x reprises la valeur centrale de $\frac{val.centrale}{x}$ pour obtenir x valeurs de seuil inférieures. Nous avons fixé x à 10, ce qui donne 21 valeurs de seuil au total. Notons que la première méthode produit toujours k propositions par point alors que les deux autres en retournent un nombre variable. Le but final est de déterminer quel type de seuil serait le plus approprié dans un environnement applicatif réel.

7. Expériences et résultats

Nos tests ont été réalisés sur un corpus de textes en français. La méthode peut être adaptée assez aisément pour une autre langue à partir du moment où les ressources suivantes sont disponibles : un dictionnaire électronique général, une liste de *stopwords*, un étiqueteur morpho-syntaxique ainsi que le graphe de prétraitement de l'ambiguïté (non indispensable, mais préférable). Bien entendu, un thésaurus approprié au corpus est toujours nécessaire. Nous n'avons pas à ce jour mené nos tests sur une autre langue que le français. Il s'agit cependant d'une de nos priorités pour le futur.

7.1. *Description du corpus et du thésaurus*

Les documents et le thésaurus proviennent de la base documentaire d'une organisation actuellement en activité. Les textes sont des documents relevant du domaine législatif et parlementaire. Le thésaurus a été spécialement conçu pour l'indexation de ces documents au sein de cette organisation.

Le thésaurus contient 2514 descripteurs et 2362 synonymes. Les decripteurs sont répartis en 47 microthésaurus. Le nombre de niveaux hiérarchiques monte jusqu'à 6, mais s'établit plus fréquemment entre 2 et 4. Les expressions composées sont bien représentées : 66,59% des descripteurs (1674 sur 2514) et 61.85% des synonymes (1461 sur 2362).

Notre corpus de test compte 12.734 fichiers XML contenant 32.953.724 mots[14]). La taille moyenne d'un document se situe par conséquent à 2588 mots. Le titre du document est délimité à l'aide de balises particulières. Pour chaque document, on dispose des catégories assignées manuellement par des indexeurs professionnels en situation réelle. Ces informations nous serviront de référence pour l'évaluation. Le nombre de descripteurs attribués varie entre 1 et 37, la valeur moyenne étant de 1,92. Dans ce corpus, certaines catégories du thésaurus ne sont représentées par aucun document et d'autres, au contraire, sont utilisées de manière très soutenue. 669 catégories ne sont jamais utilisées et le descripteur le plus fréquent est lié à 412 documents. En moyenne, une catégorie est utilisées par 9,71 documents.

7.2. *Description des expériences*

Nous avons utilisé le système décrit dans cet article sur l'ensemble des documents à notre disposition. Un test préliminaire a permis d'évaluer les performances de classification à l'aide de transducteurs ne contenant que les formes telles qu'elles apparaissent dans le thésaurus, sans aucune transformation. Le poids de chaque mot est uniquement fonction de sa fréquence d'apparition, sans intervention d'un facteur IDF ou de multiplicateurs. Deux expériences ont ensuite été menées : l'une avec les transducteurs générés au niveau hiérarchique des microthésaurus (47 catégories possibles), et l'autre à l'aide des transducteurs générés pour tous les descripteurs (2514 catégories possibles). Quelques catégories ont été interdites de sélection. C'est par exemple le cas pour le microthésaurus - et pour toutes les catégories qu'il contient - concernant les expressions temporelles. Celles-ci sont en effet souvent seulement constituées d'une indication d'année, ce qui en fait un élément très ambigu. Nous proposons plutôt d'exploiter ces informations lors d'une analyse séparée. Cette tâche, dont nous ne nous sommes pas occupé, pourra constituer une extension de ce travail.

---

14. Cette mesure approximative du texte brut (pas de balises XML) a été obtenue à l'aide de la commande *wc*

7.3. *Mesures*

Pour évaluer nos résultats, nous avons employé les mesures classiques de précision (P), de rappel (R) et de f-measure (F).

$$P = \frac{Syst_{OK}}{Syst_{TOT}} \quad R = \frac{Syst_{OK}}{Man_{OK}} \quad F = \frac{2 * P * R}{P + R}$$

où $Syst_{OK}$ est le nombre de catégories correctement proposées par le système, Man_{OK} est le nombre de catégories attribuées manuellement par l'indexeur humain et $Syst_{TOT}$ est le nombre total de catégories proposées par le système. La f-measure est une combinaison à proportion égale de la précision et du rappel.

Nous avons choisi de calculer les résultats globaux du système selon une approche microscopique. La précision, le rappel et la f-measure sont donc calculés pour chaque document et les mesures finales sont obtenues au moyen d'une moyenne arithmétique de ces valeurs. Ce choix est motivé par l'application visée qui consiste à traiter les documents un à un et non globalement.

7.4. *Résultats et perspectives*

Les résultats que nous rapportons doivent être interprétés avec précautions, car il s'agit d'une évaluation effectuée par rapport à une indexation manuelle réalisée, pour chaque document, par une seule personne. Van Slype (Van Slype, 1987) montre que la cohérence de l'indexation d'un même document par deux indexeurs se situe entre 50% et 80%. De même, Pouliquen *et al.* (Pouliquen *et al.*, 2006) rapportent un accord inter-annotateur allant de 78% à 87%. Etant donné ce désaccord, on peut considérer que notre système peut difficilement atteindre les 100% s'il est évalué par rapport à l'annotation d'une seule personne. Une évaluation plus correcte devrait être effectuée en comparant nos résultats avec les indexations de plusieurs personnes. Une autre possibilité consisterait à faire vérifier à la main les propositions de notre système afin de déterminer parmi les *mauvais* descripteurs proposés lesquels auraient pu être sélectionnés par une autre personne et lesquels auraient été jugés totalement inappropriés. Malheureusement, ce type d'évaluation nécessite une mobilisation de spécialistes qu'il est souvent difficile d'obtenir et nous ne pouvons garantir que nous pourrons mener ce type d'évaluation dans le futur.

Pour les différents tests, les trois méthodes de sélection par seuil ont été calculées. Le test préliminaire a été effectué sur l'ensemble des 2514 catégories. La recherche des expressions d'origine non modifiées et dont la fréquence n'a pas été pondérée a abouti à une f-measure maximale de 23,83% (rappel=31,65% et précision=19,11%). Le meilleur rappel atteint se situe à 52,80% mais il est accompagné par une précision très faible (6,91%). En ce qui concerne les deux expériences principales, trois points ont été mis en évidence et sont repris dans le tableau 2 : celui qui obtient la meilleure f-measure, celui qui obtient le meilleur rappel pour une précision *acceptable* d'environ 30% et enfin celui qui obtient le rappel maximal.

	47 catégories			2514 catégories		
	k-first	averaged pivot	middle pivot	k-first	averaged pivot	middle pivot
Meilleure f-measure						
Nbr. de cat.	2	1,8	**1,9**	2	1,9	**2,3**
F-measure	0,5743	0,6362	**0,6431**	0,4427	0,5066	**0,5117**
Rappel	0,6789	0,6555	**0,6785**	0,4990	0,5009	**0,5296**
Précision	0,4976	0,6180	**0,6113**	0,3978	0,5123	**0,4949**
Meilleur rappel avec précision à +/- 30%						
Nbr. de cat.	4	6,4	**5,6**	3	**10,1**	4
F-measure	0,4523	0,4516	**0,4714**	0,4004	**0,4141**	0,4799
Rappel	0,8119	0,8630	**0,8587**	0,5610	**0,6291**	0,5876
Précision	0,3135	0,3058	**0,3248**	0,3113	**0,3086**	0,4056
Meilleur rappel						
Nbr. de cat.	21	15,1	**15,1**	21	38,8	**38,8**
F-measure	0,2424	0,2354	**0.2354**	0,1694	0,1450	**0,1450**
Rappel	0,9077	0,9101	**0,9101**	0,6890	0,7086	**0,7086**
Précision	0,1399	0,1352	**0,1352**	0,0966	0,0807	**0,0807**

Tableau 2. *Résultats des test de classification.*

On remarque que la méthode *k-first* est significativement moins bonne que pour les deux autres. L'inconvénient présenté par cette méthode est de toujours proposer le même nombre de descripteurs, quel que soit le texte. Les autres méthodes, basées sur une valeur moyenne, adaptent automatiquement le nombre de propositions retournées en fonction du nombre total de descripteurs dans la liste complète, et surtout en fonction de leurs scores. Ces méthodes dynamiques sont bien entendu plus adaptées étant donné le nombre variable de catégories attribuées par les indexeurs humains. Comme ces deux méthodes donnent des résultats relativement similaires et sauf indication contraire, nous n'allons détailler les résultats que par rapport à la méthode *middle pivot*.

Dans le cas de la classification sur les 47 microthésaurus, les meilleurs résultats en terme de f-measure sont obtenus avec une valeur de 64,31%. La rappel obtenu est 67,85% pour une précision de 61,13%. Le nombre moyen de catégories proposées est 1,9 (sur 47 catégories possibles). Pour l'application visée, nous sommes intéressés de savoir quel rappel nous pouvons obtenir en acceptant une précision moindre. Un rappel de 85,87% est atteint en conservant une précision *acceptable* de 32,48% (f-measure : 47,14%). En moyenne, l'indexeur humain disposerait de 5,6 catégories. Enfin, le meilleur taux de rappel obtenu se situe à 91,01% pour une précision de 13.52%, une f-measure de 23.54% et un nombre moyen de catégories proposées de 15,1.

La classification sur l'ensemble des 2514 catégories donne des résultats moins élevés. Cela s'explique aisément par le nombre bien plus important de catégories. Par la généralisation qu'elle implique, la classification dans les 47 microthésaurus permet

d'éviter une série d'erreurs telles que la classification dans une catégorie sœur ou dans une catégorie mère/fille. La meilleure f-measure est obtenue à 51,17%. Le rappel se situe alors à 52,96% et la précision atteint 49,49%. Le nombre moyen de catégories proposées est 2,3 (sur 2514 catégories possibles). Pour rappel, l'attribution de descripteurs dans notre corpus varie entre 1 et 37, la valeur moyenne étant de 1,92. La méthode de sélection par seuil *middle pivot* ne nous a pas fourni de valeur de précision avoisinant les 30%. La méthode *averaged pivot* indique par contre que pour une précision *acceptable* de 30,86% , le rappel peut atteindre 62,91% (f-measure : 41,41%). Le nombre moyen de catégories proposées à l'indexeur humain est de 10,1. Enfin, le meilleur taux de rappel obtenu se situe à 70,86% pour une précision de 8,07%, une f-measure de 14,5% et un nombre moyen de catégories proposées de 38,8.

Deux causes sont principalement responsables des catégories non trouvées : l'absence de synonymes dans le thésaurus et l'utilisation, dans le texte, de termes trop ou pas assez concrets en regard du thésaurus. La polysémie de certains termes génère quant à elle du bruit.

A titre de comparaison, nous avons testé KEA++ sur le même jeu de données. Les résultats obtenus sont assez décevants, surtout en terme de précision[15]. Ces chiffres sont à prendre avec précaution car ils sont très éloignés de ceux rapportés par les auteurs[16] (Medelyan *et al.*, 2005).

L'absence d'apprentissage est un point intéressant de notre méthode, qui permet de la positionner comme une solution adéquate lorsque le nombre de documents disponibles pour l'apprentissage n'est pas suffisant. Il est également imaginable de l'utiliser en tant que processus d'amorçage permettant la production de l'ensemble de documents annotés nécessaires aux algorithmes d'apprentissage. Nous pensons qu'il est également possible d'exploiter cette méthode seule ou en combinaison avec d'autres techniques. Un système hybride pourait être imaginé selon deux modes différents. Tout d'abord, notre méthode peut produire un nombre restreint d'expressions *pertinentes* pouvant être ensuite exploitées par des techniques d'apprentissage. C'est une approche qui ressemble fort à celle présentée par KEA++. La seconde possibilité est l'analyse en parallèle avec d'autres systèmes de classification, conduisant à une combinaison finale des résultats obtenus par les diverses méthodes. Ces deux modes de collaboration ont été sommairement testés. Les premiers résultats ne montrent pas d'amélioration significative lors de l'utilisation d'algorithmes SVM suite à notre système, alors que la combinaison en parallèle des deux techniques semble pouvoir mener à un gain de l'ordre d'environ 5%. Ces possibilités de développement constitueront un axe important pour notre travail futur.

Ces résultats sont donc encourageants, d'autant que plusieurs évolutions doivent encore être apportées. Parmi les point importants à développer, nous avons déjà cité la mise au point d'une méthode d'analyse de l'ambiguïté pour le thésaurus et son extension automatique à l'aide de diverses ressources. D'autres améliorations de la méthode

15. Pour un rappel compris entre 0,48 et 0,53, la précision n'a atteint que 0,10.
16. Rappel=0,53 ; Précision=0.48 ; F-measure=0,47

sont encore envisageables : prise en compte de la structure hiérarchique du thésaurus, reconnaissance des expressions temporelles, correction orthographique, amélioration de la pondération et des méthodes de sélection par seuil. Enfin, il nous semble indispensable de tester ce système à la fois sur un autre corpus et sur une langue différente.

8. Conclusion

Nous avons présenté une méthode semi-automatique permettant d'améliorer la rapidité et la cohérence de l'indexation manuelle de textes. Cette méthode est basée, comme pour le processus manuel, sur un thésaurus qui décrit le domaine d'indexation. Ce thésaurus a été converti automatiquement sous la forme de transducteurs qui, appliqués aux textes, repèrent les expressions *pertinentes* et y adjoignent la catégorie du thésaurus correspondante. Après pondération des expressions, on obtient finalement pour chaque document, un score total pour chaque catégorie représentée dans la liste. Un système de sélection par seuil permet ensuite de réduire le nombre de catégories proposées en ne gardant que les plus probables. L'indexeur humain peut alors faire son choix parmi cette liste réduite. L'évaluation a été effectuée sur des données en français : un ensemble de textes indexés manuellement par des indexeurs humains à l'aide d'un thésaurus ad-hoc. Les résultats encourageants obtenus (une f-measure située entre 0,51 et 0,64 selon les tests), les possibilités de développements futurs, l'absence d'apprentissage et la possibilité de démarrer une analyse sur un ensemble restreint de documents permettent d'envisager ce système en tant que méthode principale d'indexation, mais aussi en tant que méthode préliminaire ou parallèle à d'autres méthodes.

Remerciements

Ce travail a été effectuée dans le cadre du projet STRATEGO financé par la Région wallonne (Belgique). Nous tenons également à remercier IRIS, le partenaire industriel de ce projet, pour leur précieuse collaboration.

9. Bibliographie

AFNOR, « Règles d'établissement des thésaurus monolingues », December, 1981. NF Z47-100.

Baeza-Yates R., Ribeiro-Neto B., *Modern Information Retrieval*, 1st edn, Addison Wesley, May, 1999.

Crochemore M., Rytter W., *Text Algorithms*, Oxford University Press, October, 1994.

Da Sylva L., « Thésaurus et systèmes de traitement automatique de la langue », *Documentation et bibliothèques*, vol. 52, p. 149-156, 2006.

ISO, « Guidelines for the establishment and development of monolingual thesauri », 1986. ISO 2788.

Medelyan O., Witten I. H., « Thesaurus-Based Index Term Extraction for Agricultural Documents », *6th Agricultural Ontology Service (AOS) workshop at EFITA/WCCA 2005*, Vila Real, Portugal, 2005.

Medelyan O., Witten I. H., « Thesaurus based automatic keyphrase indexing », *6th ACM/IEEE-CS joint conference on Digital libraries*, ACM, Chapel Hill, NC, USA, p. 296-297, 2006.

Moens M.-F., *Information Extraction : Algorithms and Prospects in a Retrieval Context*, 1 edn, Springer, October, 2006.

Névéol A., Mork J. G., Aronson A. R., Darmoni S. J., « Evaluation of French and English MeSH Indexing Systems with a Parallel Corpus », *AMIA Annual Symposium Proceedings*, vol. 2005, p. 565-569, 2005. PMC1560460.

Névéol A., Rogozan A., Darmoni S., « Automatic indexing of online health resources for a French quality controlled gateway », *Inf. Process. Manage.*, vol. 42, p. 695-709, 2006.

Paumier S., *Unitex 2.0 User Manual*, October, 2008. http ://www-igm.univ-mlv.fr/ unitex/manuel.html.

Pereira S., Neveol A., Kerdelhué G., Serrot E., Joubert M., Darmoni S. J., « Using multi-terminology indexing for the assignment of MeSH descriptors to health resources in a French online catalogue », *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*p. 586-90, 2008. PMID : 18998933.

Porter M. F., *An algorithm for suffix stripping*, Morgan Kaufmann Publishers Inc., p. 313-316, 1997.

Pouliquen B., Steinberger R., Ignat C., « Automatic annotation of multilingual text collections with a conceptual thesaurus », *cs/0609059*, September, 2006. Proceedings of the Workshop 'Ontologies and Information Extraction' at the Summer School 'The Semantic Web and Language Technology - Its Potential and Practicalities' (EUROLAN'2003), pp 9-28. Bucharest, Romania, 28 July - 8 August 2003.

Schmid H., « Probabilistic Part-of-Speech Tagging Using Decision Trees », Manchester, UK, 1994.

Sebastiani F., « Machine learning in automated text categorization », *ACM Computing Surveys*, vol. 34, p. 1-47, 2002.

Van Slype G., *Les langages d'indexation : conception,construction et utilisation dans les systèmes documentaires*, Systèmes dInformation et de Documentation, Les éditions d'organisation, Paris, 1987.

Yarowsky D., « One sense per collocation », Association for Computational Linguistics, Princeton, New Jersey, p. 266-271, 1993.

Modèle d'indexation dynamique à base d'ontologies

Gilles Hubert[1], Josiane Mothe[1,2], Bachelin Ralalason[1], Bertin Ramamonjisoa[3]

[1] *IRIT, 118 route de Narbonne, 31062 Toulouse Cedex 9,*
[2] *Institut Universitaire de Formation des Maitres, Av. de l'URSS, 31078 Toulouse*
[3] *Université de Fianarantsoa, Ecole Nationale d'informatique, BP 1487 Tanambao – Fianarantsoa 301, Madagascar*

{hubert/mothe/bachelin}@irit.fr

bertin@mail.univ-fianar.mg

RÉSUMÉ. *Cet article propose un modèle de données pour une indexation basée sur une ontologie de référence représentant la sémantique des termes d'indexation. Le modèle proposé vise à permettre une indexation en temps réel qui suit la dynamique du corpus tout en assurant la disponibilité des documents et de l'index. Ceci permet de garder la cohérence entre les documents de la collection, l'index et l'ontologie de référence. Notre modèle permet ainsi d'éviter la reconstruction de l'index lors de la modification du corpus de documents car il reste à jour en permanence. Ainsi, le modèle que nous proposons permet l'indexation sémantique dynamique d'un corpus. Le modèle est illustré par des algorithmes expliquant sa mise en œuvre.*

ABSTRACT. *This paper proposes a data model for semantic indexing based on an ontology. The ontology represents the semantic of the index terms. The proposed model aims at enabling real-time indexing according to the dynamic of the corpus while insuring the availability of documents and the index. So, it permits to keep the coherence between the documents, the index and the reference ontology. Using our model, there is no need to rebuild the index from scratch because the index is permanently up to date. Thus, the model we propose makes it possible a dynamic and semantic indexing of documents collection. The model is illustrated by some algorithms showing its implementation.*

MOTS-CLÉS : *Recherche d'information, Indexation sémantique, Indexation à base d'ontologie, Dynamique des corpus, Structures de donnés.*

KEYWORDS: *Information retrieval, Semantic indexing, Ontology-based indexing, Dynamics in corpus, Indexing data structure.*

1. Introduction

L'objectif des systèmes de recherche d'information (SRI) est de fournir aux utilisateurs les documents pertinents par rapport aux besoins qu'ils expriment.

Les SRI utilisent des listes inversées qui rassemblent les différents termes d'indexation choisis pour représenter les contenus des documents et les liens vers ces documents. En complément, à chaque couple (terme d'indexation, document) est associé un poids qui représente l'importance du terme dans un document. Lorsqu'une requête est soumise au système, les termes qu'elle contient sont mis en correspondance avec les termes d'indexation extraits des documents pour en déduire les documents à restituer à l'utilisateur. La phase d'indexation est donc une phase primordiale dans le processus de recherche. Lorsque la collection de documents est figée, l'indexation est réalisée une fois pour toutes. Cependant, ce cas n'intervient que dans le cadre des campagnes d'évaluation des moteurs où il s'agit de confronter différents SRI sur des mêmes collections. Dans l'usage réel, le SRI doit être capable de faire face à des collections dynamiques dans lesquelles des documents sont modifiés, ajoutés et supprimés.

Dans la littérature, diverses méthodes et stratégies ont été proposées pour permettre la mise à jour des index lorsque la collection de documents est modifiée. Il s'agit par exemple de l'utilisation des délimiteurs [Salton et al, 1993] et [Baeza-Yates et Navarro, 2000], la mise à jour incrémentale d'index [Lim et al., 2007], ainsi que la méthode *diff* [Ukkonen, 1985]. Ces méthodes considèrent une indexation de type « sac de mots », dans laquelle les termes issus des documents sont considérés comme indépendants. Cependant, il existe en réalité des relations (équivalence, subsomption, association, …) entre les termes. De nouvelles approches tentent de les prendre en compte de façon automatique lors de l'indexation, par exemple au travers de l'indexation sémantique [Hernandez et al. 2007]. L'indexation sémantique via des ontologies à laquelle nous nous intéressons dans nos travaux s'appuie sur les technologies du web sémantique. Dans ce type d'approche, la connaissance du domaine (terminologique en particulier) est représentée sous forme d'ontologies, c'est-à-dire en particulier de concepts, d'instances de ces concepts et de relations.

Comme dans le cas de l'approche sac de mots, la collection de documents à indexer peut être dynamique et donc subir des modifications ; il est donc important de proposer des principes pour la mise à jour des index dans le cas d'une indexation sémantique. De plus, contrairement à l'approche sac de mots, le vocabulaire utilisé lors de l'indexation peut être amené à varier indépendamment des documents. Ainsi, l'ontologie qui sert de référence à l'indexation peut être modifiée. Dans ce dernier cas, il est important de considérer la mise à jour de l'indexation consécutive à une modification du vocabulaire de référence, cela afin de maintenir une cohérence entre les documents et le vocabulaire d'indexation.

Cet article présente les structures de données nécessaires ainsi que les stratégies utilisées pour permettre l'actualisation en temps réel des listes inversées issues de l'indexation sémantique basée sur une ontologie. Dans un premier temps, dans la section 2, nous présentons les différents travaux de la littérature liés à cet axe de recherche. Puis, dans la section 3, nous présentons notre modèle de données représentant les structures d'index. En section 4 nous discutons nos solutions en matière de structure de données et de stratégies des mises à jour d'index. Enfin, nous terminons, dans la section 5, par des conclusions et perspectives à notre travail de recherche.

2. Etat de l'art

La mise à jour des documents de la collection, l'arrivée de nouveaux documents ou la suppression de documents d'un corpus indexé nécessitent l'actualisation de l'index afin de garder la cohérence entre les documents et les index. Cette mise à jour est primordiale pour que le SRI puisse répondre au mieux aux besoins d'un utilisateur.

Dans cette section, nous présentons les travaux reliés relatifs à l'indexation dynamique des documents ainsi qu'à l'indexation sémantique à partir d'ontologies dans la mesure où notre approche s'intègre dans ce type d'indexation.

2.1. Indexation dynamique

L'indexation dynamique consiste à mettre à jour l'index après modification de la collection (ajout, modification et suppression de documents).

La toile correspond à ce cadre de collections hautement dynamiques. Ainsi des travaux dans le domaine de la recherche et de la collecte incrémentale des pages web visent à permettre aux collections d'un moteur de recherche d'être plus synchronisées avec le web réel. Dans [Cho et Garcia-Molina, 2000] la collecte vise à télécharger toutes les pages web relatives à un URL de départ. Ensuite, la collecte incrémentale ne télécharge que les pages qui ont été modifiées à la source. Dans cette méthode, la maintenance de la collection ne nécessite pas de télécharger les pages qui n'ont subi aucune modification. Cependant, la collection synchronisée, qui est l'image exacte des documents sources, ne peut pas être recherchée immédiatement car la reconstruction (à partir de zéro) de l'index de mots-clés est moins fréquente que la mise à jour de la collection. Il n'est donc possible de rechercher les nouveaux documents collectés qu'après la prochaine reconstruction de l'index. Le décalage entre la mise à jour de la collection et celle de l'index s'explique par le fait que la reconstruction complète de l'index est très coûteuse en termes de temps de traitement.

Face à ce problème, [Lim et al., 2007] propose une méthode de mise à jour incrémentale d'index inversé pour les documents qui ont changé sur le web. Cette méthode qui s'appelle *délimiteur-diff* [Landmark-diff en anglais] combine la technique de *délimiteur* [Landmark en anglais] [Salton et al, 1993], [Baeza-Yates et Navarro, 2000] avec la méthode *diff* [Ukkonen, 1985]. La technique de délimiteur consiste à subdiviser les documents en plusieurs blocs et à mémoriser les positions relatives des mots du document par rapport aux délimiteurs du bloc dans lequel les mots se trouvent. La méthode *diff* par contre mémorise la liste des modifications apportées dans un document pour obtenir sa nouvelle version. A chaque document est donc associé un annuaire de délimiteurs qui liste les différents délimiteurs et leurs positions absolues dans le document. La mise à jour d'un document entraine celle de l'annuaire des délimiteurs associé au document. Ce nouvel annuaire des délimiteurs associés à la *transcription des modifications* [edit transcript en anglais] permet de mettre à jour l'index inversé. La transcription des modifications correspond à la liste des modifications qui amènent vers la nouvelle version d'un document à partir de l'ancienne version). Une entrée dans l'index inversé est composée de l'identifiant de chacun des mots (wordID), la liste des documents contenant un mot donné (docID), l'identifiant du délimiteur (landmarkID) auquel le mot est rattaché, ainsi que la position relative (offset) du mot par rapport au délimiteur. La méthode Délimiteur-diff présente une vitesse de mise à jour de l'index inversé (pour les documents qui ont changé) trois fois plus rapide que la méthode *Premier Index* [forward index en anglais] utilisée par Google [Page et Brin, 1998]. De son côté, [Büttcher S. et Clarke C., 2006] propose une approche hybride qui combine les deux méthodes *In-Place* (mise à jour directe sur place) et *Merge-based* (mise à jour d'index basée sur la technique de fusion) qui sont deux techniques largement utilisées pour la mise à jour d'index dans un SRI dynamique basés sur les listes inversées. La stratégie *In-Place* consiste non seulement à transformer les structures de données en d'autres structures plus petites mais aussi à écraser les anciennes versions de documents. La technique *Merge-based* [Cutting et Pedersen, 1990], quant-à elle, consiste à minimiser le déplacement de la tête de disque pour la maintenance de l'index. La mise à jour des index sur disque se fait dès que l'espace mémoire alloué aux index commence à être saturé. L'inconvénient de la technique *Merge-based* est que la totalité de l'index inversé est lu et écrit sur le disque à chaque mise à jour, même si une petite partie seulement de l'index est affecté. Les accès au disque sont donc importants. La stratégie *In-place* essaie de résoudre ce problème en laissant assez de place à la fin des index inversés. Dès lors, [Büttcher S. et Clarke C., 2006] utilisent la méthode *In-place* (respectivement *Merge-based*) pour une longue liste inversée (respectivement une courte liste inversée). Cette approche hybride combinant l'utilisation de la méthode *In-place* et *Merge-based* donne une meilleure performance en termes de temps d'indexation que l'une ou l'autre de ces méthodes, tout en gardant la même performance au niveau de traitement de la requête.

La plupart des algorithmes d'actualisation d'index ne permet pas l'ajout de nouveaux documents pendant le processus de mise à jour de l'index. De plus, ce processus peut demander plusieurs heures pour les corpus de grande taille. Ainsi, [Galambos L., 2006] a développé un algorithme de mise à jour dynamique d'index. Cet algorithme est basé sur la mise à jour incrémentale d'index en utilisant une liste inversée de type *Citerne (Tanker).* Dans ce modèle, une *citerne* est un index composé de *Barils* (*barrels*) ; où un baril est un index réalisé sur un sous-ensemble de documents du corpus. L'actualisation d'index est réalisée à chaque arrivée de nouveaux documents et à chaque modification de documents. La modification d'un document est considérée comme une opération de suppression suivie d'un ajout d'un nouveau document.

En ce qui concerne Google [Page et Brin, 1998] qui a été conçu pour rechercher sur le www, il emploie plusieurs techniques (importance des pages ou PageRank, structure des liens, texte des liens, polices de caractères, position des mots dans les documents, etc…) pour améliorer la qualité de recherche. L'analyse des structures des liens à partir de la popularité des pages permet à Google d'évaluer la qualité des pages web [Page et Brin, 1998]. Pour atteindre ses objectifs, les structures de données utilisées par Google sont :

Liste d'importance (*Hit list*) : il s'agit d'une liste des occurrences d'un mot d'un document particulier, comprenant les informations de la position, la police et la taille (visuelle) du mot.

Baril (*Barrel*) : Index obtenu par la méthode *Premier Index* (*Forward-Index*) présentée plus haut, partiellement trié par le docID. Chaque baril stocke des listes d'importance pour un ensemble d'identifiant de mots (*wordID)*. Il y a deux types de barils : Les barils courts qui contiennent les listes d'importance incluant le titre ou les ancres et les barils longs pour toutes les listes d'importance.

Entrepôt (*Repository*) : Contient le code HTML des pages Web. Chaque document est préfixé par son identifiant docID, sa longueur, et son URL.

Index Doc: Garde les informations concernant les documents telles qu'un pointeur vers le dépôt, le nombre de liens provenant d'autres pages, le nombre de liens sortant qui pointent vers d'autres pages, l'état de chaque document ainsi que son URL.

Lexique : Table de hashage gérée en mémoire vive qui garantit l'appariement entre un mot et son identifiant (wordID)

Ancre : Stocke les destinations et les étiquettes des liens.

Dans nos travaux, nous nous inspirons de ces différentes techniques qui visent à indexer les documents sur le web et les adaptons à des collections indexées par des ontologies. Ainsi, la méthode proposée par Google associée avec la méthode *Délimiteur-Diff* va nous permettre de gérer dynamiquement l'évolution de l'indexation des documents d'une collection. Du point de vue sémantique,

l'indexation avec une ontologie permet d'exprimer les relations entre des expressions du document à l'aide de celles des concepts auxquels les expressions sont associées dans l'ontologie. De plus, l'utilisation des concepts d'ontologie comme vocabulaire de référence d'index sert à bien préciser les sens accordés aux termes d'un document.

2.2. Indexation sémantique

Suite au développement du web sémantique et de ses technologies [Berners-Lee, 2001], différents travaux s'intéressent à son application en RI. L'utilisation d'une ontologie lors de la phase d'indexation permettrait de lever les ambiguïtés des sens des termes utilisés et de mieux représenter les connaissances inhérentes dans les documents. En termes d'indexation sémantique, des concepts de l'ontologie sont associés à chaque document selon les sémantiques qui y sont véhiculées. Dans cette section, nous présentons des méthodes et structures de données utilisées pour manipuler les index et ontologies en vue de la RI sémantique.

Différents travaux ont montré l'intérêt d'utiliser une indexation sémantique à base d'ontologie. Dans le domaine de l'apprentissage en ligne, [Chang et al., 2007] propose une indexation basée à la fois sur une ontologie du domaine de l'apprentissage et sur une ontologie dérivée de LOM (Learning Object Metadata), qui représente les métadonnées décrivant les ressources pédagogiques. Dans le cadre des recherches d'objets pédagogiques relatifs aux mathématiques en secondaire, les résultats montrent une meilleure efficacité en termes de rappel et de précision par rapport aux mêmes recherches basées sur mots-clés. De même, [Hernandez et al., 2008] utilise les termes d'une ontologie de domaine, associée à une ontologie de tâche et de scenario d'apprentissage comme valeurs des métadonnées de LOM. Afin d'accéder aux instances d'ontologie d'une part et aux index associés aux documents d'autre part, [Hernandez et al., 2007] propose de les stocker dans une base de données relationnelles.

Par ailleurs, [Song et al., 2005] propose un modèle de RI basé sur des ontologies de domaine, définies avec OWL lite. Les différentes ontologies de domaines sont intégrées pour former une ontologie unique. Les termes définis dans l'ontologie sont alors utilisés d'une part comme métadonnée pour annoter les contenus du web et d'autre part comme termes d'indexation de la collection.

Dans ces différents travaux, les structures de données utilisées permettent d'associer des concepts issus d'une ontologie aux documents de la collection. Cependant, le suivi de la dynamique (ajout, suppression, modification) des documents de la collection ainsi que l'impact de ces évolutions au niveau de l'index sémantique n'a pas été traité. Dans la section suivante, nous présentons le modèle de données que nous avons défini et qui permet une indexation sémantique dynamique.

3. Modèle pour une indexation dynamique basée sur une ontologie

Dans cette section, nous proposons les structures de données et les stratégies d'indexation nécessaires en vue de la mise en place d'un SRI basé sur une indexation sémantique par ontologies. Ces structures sont conçues pour permettre la mise à jour dynamique des index tout en permettant la recherche d'information sémantique.

3.1. *Paramètres et contraintes*

Chaque SRI est jugé au travers de ses performances en termes de satisfaction utilisateur sur les pertinences des documents retrouvés, du temps de réponse aux requêtes utilisateurs ainsi que la disponibilité du système. Ainsi, afin de définir les structures de données à utiliser dans les listes inversées, plusieurs paramètres qui entrent en jeu dans la performance des SRI doivent être pris en compte. Parmi ces paramètres, nous pouvons citer : la taille du corpus (nombre de documents constituant la collection), la fréquence de mise à jour de la collection, et le format des documents. En effet, la mise à jour de la collection demande la ré-indexation des documents. La taille du corpus affecte la durée de ré-indexation, cela peut engendrer une lenteur du système, voire son indisponibilité pendant un certain temps. De même, la fréquence de mise à jour de la collection a un impact sur la disponibilité de l'index. En effet, plus la fréquence de mise à jour de la collection est élevée, moins l'index est disponible pour la recherche d'information car il est à tout moment en cours de modification. Enfin, le format des documents affecte le temps d'indexation car les durées d'extraction des termes et expressions d'un document ne sont pas les mêmes pour tous les formats.

Ces paramètres liés à la collection se combinent avec les exigences des utilisateurs qui souhaitent obtenir des documents pertinents dans un meilleur délai et les principes utilisés en RI au moment de l'indexation. Ainsi, notre modèle prend en compte les contraintes suivantes:

- Minimisation du délai de mise à jour dynamique d'index. Dans les SRI actuels, suivant la taille du corpus, la mise à jour de l'index peut prendre des heures de traitement. Ceci limite donc les ajouts de nouveaux documents ou modification de documents dans le corpus et leur prise en compte dans les index. Selon notre approche, à chaque arrivée de nouveaux documents (qui peut emmener de nouvelles instances d'expressions), nous mettons à jour seulement les données statistiques du nouveau document et de ses expressions (fréquences d'apparition suivant le type de texte). Le calcul des poids des expressions et concepts se fait au moment de l'évaluation de la requête. De plus, la mise à jour des données suite à une modification de la collection ne s'effectue qu'au niveau des documents concernés.

- Minimisation du temps de réponse (traitement des requêtes). Le temps qui s'écoule entre la saisie de la requête et l'affichage de résultat doit se situer dans la limite acceptable par l'utilisateur. Le fait de traiter dynamiquement l'indexation des documents ne doit pas pénaliser de trop le temps de réponse du système lors d'une recherche de documents. La minimisation du temps d'indexation assure la disponibilité permanente de l'index. A son tour, la disponibilité de l'index permet d'évaluer les requêtes utilisateurs à tout moment. Dans notre modèle, toutes les informations statistiques et sémantiques qui servent à l'évaluation des requêtes sont accessibles dans la base.

- Pondération des termes d'indexation pour rendre compte de leur pouvoir discriminant. La pondération des expressions dans les index tient compte de la mesure tf*idf. Ce poids associé aux termes d'indexation [Robertson, 1976] est utilisé lors de l'étape d'appariement d'une requête avec les documents. Tf (Term Frequency) est la fréquence d'apparition d'un terme dans le document et Idf (Inverse Document Frequency) est la valeur de l'importance du terme dans l'ensemble de la collection. Dans notre modèle, la pondération tient également compte de certains éléments associés aux expressions dans le document comme leur mise en forme.

- Recherche sémantique sur les contenus. Le système recherche dans l'ontologie les concepts qui correspondent aux termes de la requête puis restitue les documents qui sont indexés par ces concepts.

Les structures de données qui sont présentées dans la section 3.2 ont été conçues pour répondre à ces contraintes et objectifs. Elles contiendront donc les éléments de l'index et aussi de l'ontologie qui sert de référence à l'indexation. Notons que le format des documents de la collection n'affecte pas les structures de données.

3.2. Modèle

Nous avons défini un modèle de données servant de socle à la mise en place d'un SRI basé sur une indexation sémantique par ontologie. Ce modèle présenté dans la figure 1 prend notamment en compte le double objectif d'actualisation dynamique des listes inversées et d'utilisation d'ontologies lors de l'indexation.

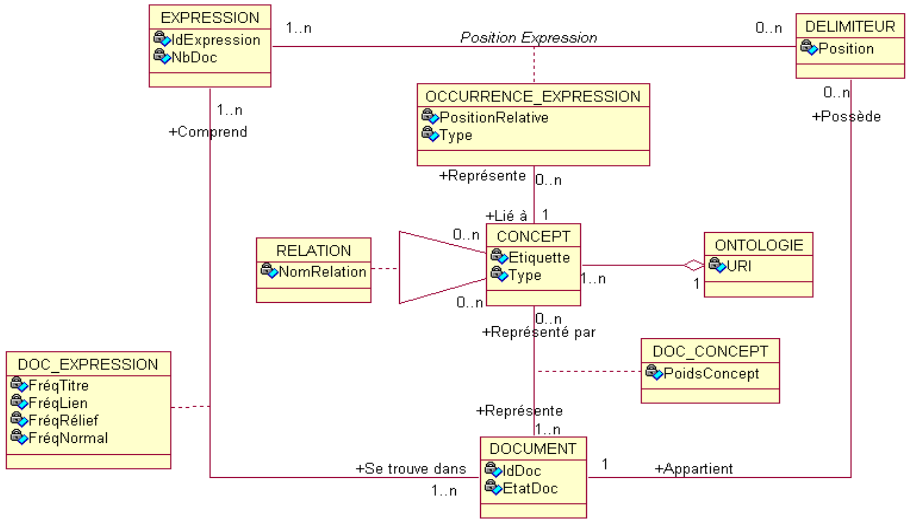


Figure 1 : Diagramme de classes représentant les données utilisées pour l'indexation.

La classe *DOCUMENT* représente les unités d'indexation. A chaque document est associé son identifiant (*IdDoc d'un document correspondant à son URI - Uniform Resource Identifier*), unique référence dans l'ensemble du système. Trois états possibles, impliquant des traitements différents, sont distingués pour un document. L'état d'un document peut être *normal* (cas des documents intégrés dans le système avec indexation à jour), *mis à jour* (cas des documents dont le contenu a été modifié depuis la précédente indexation) ou bien *effacé* (le document est retiré logiquement du système depuis la précédente indexation). Chaque document est subdivisé en plusieurs sections séparées par des marqueurs logiques. La classe *DELIMITEUR* représente ces marqueurs en précisant leur *Position* absolue dans le document.

Par ailleurs, chaque document est considéré comme un ensemble d'expressions décrites au travers de la classe *EXPRESSION*. Une expression est un groupe de termes représentant un concept du domaine. Le nombre de documents (*NbDoc*) où apparaît chaque expression est conservé. Chaque occurrence d'une expression apparaissant dans un document est repérée par sa position relative (*PositionRelative*) par rapport à un délimiteur du document. Cette idée est modélisée par l'association *Position Expression* et la classe d'association *OCCURRENCE_EXPRESSION*. La propriété *Type* conserve l'importance du texte incluant l'occurrence de l'expression (c'est-à-dire si celle-ci apparaît dans un titre, dans un lien hypertexte, dans un texte mis en relief ou dans un texte normal). La mémorisation des positions des

occurrences des expressions (*PositionRelative*) dans les documents offre des possibilités de localisation précise des index pour l'utilisateur.

De plus, chaque occurrence d'une expression peut être associée à un concept d'une ontologie de référence. La classe CONCEPT décrit les instances des concepts identifiées par leur *Etiquette* telle que définie dans l'ontologie. Cela permet de faire le lien avec la définition de l'ontologie repérée par son *URI* (classe *ONTOLOGIE*).

Une instance de concept peut être liée à d'autres par des instances de relations (ces relations étant définies entre concepts dans l'ontologie). Plusieurs relations pouvant exister entre deux concepts, il est nécessaire de conserver le nom de la relation concernée (*NomRelation*). La classe d'associations *DOC_EXPRESSION* rassemble les données statistiques concernant les expressions et leur apparition dans les documents (fréquence d'apparition dans les titres, dans des liens, dans des textes en relief, dans des textes normaux).

Enfin, la classe d'association *DOC_CONCEPT* précise les poids des concepts associés à un document. Un concept n'est pas forcément associé à une expression d'un document, il peut être affecté explicitement au document.

4. Exploitation du modèle

Le modèle de données sur lequel se base notre approche est axé sur l'utilisation d'ontologies lors de l'indexation, et principalement sur l'actualisation dynamique des listes inversées consécutive à l'évolution de la collection de documents et de l'ontologie. L'utilisation de la structure de données présentée dans la figure 1 est détaillée, d'une part, du point de vue de l'évolution de l'index, et d'autre part, de l'évolution de l'ontologie. Les données seront stockées dans une base de données Oracle pour faciliter leurs accès.

4.1 Ajout d'un nouveau document :

A chaque arrivée de nouveau document, l'utilisation de la structure de données suit l'algorithme 1 :

```
Début
   Créer une instance de DOCUMENT à l'état (EtatDoc) normal.
   Délimiter le nouveau document en blocs de paragraphes
   Pour chaque bloc lié à un délimiteur Faire    /* Intégration de
nouveau bloc  */
      Créer une instance dans DELIMITEUR
      Extraire les expressions décrivant le bloc
      Pour chaque expression Faire  /* Ajout d'expression */
         Si expression n'existe pas dans la classe EXPRESSION Alors
```

```
            Créer une nouvelle instance d'expression
            Créer une nouvelle instance de DOC_EXPRESSION
         FinSi
         Si instance de DOC_EXPRESSION n'existe pas Alors
            Créer une nouvelle instance de DOC_EXPRESSION liée à
         l'expression
            Mettre à jour la valeur de NbDoc dans EXPRESSION
         Sinon
            Mettre à jour les propriétés (fréquences d'apparition) pour
         l'instance de DOC_EXPRESSION
         FinSi
         Créer une instance de la classe d'association
         OCCURRENCE_EXPRESSION liée aux instances d'expression et de
         délimiteur en cours
         Moyennant d'une part des relations entre Termes et Concepts,
         et d'autre part les relations entre Concepts dans l'ontologie
         de domaine, identifier l'instance de CONCEPT correspondant à
         l'occurrence d'expression et les éventuelles relations
         auxquelles l'instance de CONCEPT participe.
      Fin Pour
      Identifier les éventuelles relations entre instances de CONCEPT
   Fin Pour
Fin
```

Algorithme 1 : Prise en compte de l'ajout d'un nouveau document

La suppression ou la modification d'un document implique dans un premier temps uniquement la mise à jour de son statut dans la classe *DOCUMENT* avant de mettre à jour toutes les autres classes.

4.2. Suppression d'un document :

Pour un document supprimé, son statut est changé en « Effacé » avant de mettre à jour les différentes informations relatives au document dans la base. Ainsi, ce document ne sera plus pris en compte par les requêtes. Après les mises à jour des données du document, le document sera supprimé physiquement de la collection. L'algorithme 2 correspond à la suppression d'un document.

```
Début
   Changer l'état du DOCUMENT à l'état (EtatDoc) Effacé. /* Le
   document sera ignoré par toutes les requêtes */
   Pour chaque délimiteur du document dans DELIMITEUR Faire
      Supprimer toutes les occurrences de OCCURRENCE_EXPRESSION liées
au délimiteur
      Supprimer le délimiteur
   Fin Pour
   Pour chaque instance de DOC_EXPRESSION liée au document Faire
```

```
      Décrémenter NbDoc de l'expression correspondante dans EXPRESSION
      Si NbDoc = 0 Alors Supprimer l'expression dans EXPRESSION
      Supprimer l'occurrence de DOC_EXPRESSION
   Fin Pour
   Supprimer les occurrences de DOC_CONCEPT relatif au document
   Supprimer le document dans DOCUMENT
   Supprimer physiquement le document de la collection
Fin
```

Algorithme 2 : Suppression d'un document

4.3 Modification d'un document :

A chaque modification d'un document, détectée à l'aide de la méthode *diff* [Ukkonen, 1985], son statut passe à l'état « Modifié » avant la phase de mise à jour des données relatives aux expressions du document. Une légère modification dans un document peut apporter ou supprimer des termes importants. L'importance de la modification en termes d'impact sur l'indexation ne sera connue qu'au moment de l'appariement du document avec une requête. Les éventuelles requêtes faisant appel au document en cours de modification utilisent les anciennes valeurs dans la base en attendant la fin des mises à jour des données relatives au document modifié. L'état du document redevient « Normal » après les mises à jour des données. Une simple réindexation du document modifié n'est pas viable sur un gros document. Les mises à jour qui peuvent être réalisées dans un document sont à l'origine des opérations suivantes :

- Ajout de nouvelles expressions
- Suppression d'expressions
- Modification d'une occurrence d'expression

Ajout de nouvelles expressions :

L'ajout d'expressions dans un document déjà indexé a pour conséquence de modifier l'indexation comme le montre l'algorithme 3.

```
Début
    Changer l'état du DOCUMENT à l'état  Modifié.
    /* L'affectation de cette nouvelle expression au bloc dans lequel
    il est inséré */
    Créer une occurrence dans OCCURRENCE_EXPRESSION
    Si instance de DOC_EXPRESSION n'existe pas Alors
       Créer une nouvelle instance de DOC_EXPRESSION liée à l'expression
       Incrémenter la valeur de NbDoc dans EXPRESSION
```

```
    Sinon
       Mettre à jour les propriétés (fréquences d'apparition) pour
       l'instance de DOC_EXPRESSION
    FinSi
    Mettre à jour les positions des délimiteurs (du document) dont la
    valeur de la propriété position est supérieure à celle du
    délimiteur du bloc contenant la nouvelle expression.

    Si la taille du bloc devient trop importante (le double de la
    taille normale d'un bloc) Alors /* le bloc courant est éclaté en
    deux. */
        Créer une nouvelle occurrence de délimiteur dans DELIMITEUR
        Pour chaque expression de la deuxième partie du bloc faire
               Affecter l'occurrence d'expression au nouveau délimiteur
               Mettre à jour la position relative par rapport au
        nouveau délimiteur
        FinPour
    FinSi
    Changer l'état du DOCUMENT à l'état  Normal
Fin
```

Algorithme 3 : Modification d'un document – ajout d'une expression

Suite à des successions de plusieurs ajouts d'expression, la taille du bloc peut devenir trop importante. Cela peut affecter la durée de mise à jour des instances de *DOC_EXPRESSION* dans le cas où la nouvelle expression est insérée au début du bloc. Pour palier ce problème, l'éclatement d'un grand bloc permet de réduire le nombre de mises à jour à effectuer dans *DOC_EXPRESSION*.

Suppression d'expression :

La suppression d'une expression entraîne une modification des instances du modèle comme l'indique l'algorithme 4.

```
Debut
    Changer l'état du DOCUMENT à l'état  Modifié
    Supprimer l'occurrence dans OCCURRENCE_EXPRESSION
    Décrémenter la propriété fréquence correspondante dans
    DOC_EXPRESSION
    Si la somme des fréquences est égale à zéro alors
        Supprimer l'entrée dans DOC_EXPRESSION
        Décrémenter NbDoc dans EXPRESSION
        Si Nbdoc=0 alors supprimer l'expression dans EXPRESSION
    FinSi
```

```
    Si la Taille du bloc courant < Seuil et le nombre de bloc du
    document >=2 alors /* Bloc devenu trop petit suite à plusieurs
    suppressions*/
        Si le bloc courant correspond au premier délimiteur du document
        alors Prendre le bloc suivant comme bloc courant
        Pour chaque expression du bloc courant Faire
            Affecter l'expression au bloc précédent
            Mettre à jour sa position relative par rapport au bloc
            précédent
        FinPour
        Supprimer le délimiteur du bloc dans DELIMITEUR
    Fin
    Changer l'état du DOCUMENT à l'état  Normal
Fin
```

Algorithme 4 : Modification d'un document – suppression d'une expression

Pour limiter le nombre de blocs dans un document, les blocs de petite taille seront fusionnés avec un autre bloc car un nombre important de blocs entraîne plusieurs mises à jour dans *DELIMITEUR* à chaque ajout d'un nouveau bloc. L'affectation des expressions à la fin du bloc précédent permet d'éviter de modifier toutes les positions relatives des expressions du bloc suivant.

Modification d'une occurrence d'expression :

La modification d'une occurrence d'expression se traduit par la séquence :

- Suppression d'une expression
- Ajout d'une nouvelle expression

Dans tous les cas (Ajout, Suppression ou Mise à jour de documents de la collection), la mise à jour dynamique de l'index n'est réalisée que sur les documents concernés. Cela permet de diminuer le temps d'indexation, et ainsi d'augmenter la disponibilité de la collection à tout moment. De plus, l'indexation dynamique permet de conserver une cohérence entre collection et index, ce qui permet au système de trouver les documents pertinents à tout moment. A l'indexation, aucune méthode de compression n'a été utilisée pour ne pas dégrader le temps de réponse.

5. Conclusions et perspectives

Cet article présente un modèle de données dans le cadre d'une indexation à base d'une ontologie de référence. Cette structure de données permet en outre une mise à jour dynamique et en temps réel des résultats de l'indexation lors de la mise à jour

de la collection de documents. Cette structure assure ainsi la cohérence permanente entre l'index, le corpus et l'ontologie de référence. L'avantage principal du modèle que nous proposons est qu'il n'est plus nécessaire de reconstruire l'index car il est à jour à tout moment. Ainsi, la structure que nous proposons permet de mettre en place une indexation sémantique dynamique.

Plusieurs perspectives sont envisagées pour ce travail. Une ontologie en tant que représentation de connaissances d'un domaine évolue pour représenter au mieux le domaine. L'ontologie qui a servi de base de référence pour le choix des termes d'indexation des documents peut donc évoluer. La cohérence de l'ontologie et de l'indexation des documents n'est alors plus assurée.. Notre prochaine étude consiste à l'étude de la prise en compte de cette évolution et son impact sur l'indexation des documents. Dans cet article, nous avons opté pour le calcul des poids des expressions au moment de l'évaluation d'une requête afin de simplifier le traitement et de gagner de temps à la mise à jour dynamique d'index. Partant de ce fait, nous comptons évaluer le temps nécessaire pour calculer le poids d'une expression, à partir des informations statistiques, et de le comparer avec celui de la lecture d'un poids pré calculé et enregistré dans la base. De la même manière, la complexité du système en temps d'exécution et de taille de données devra être évaluée suivant la taille de la collection, du format des documents et de la taille de l'ontologie. Nous sommes partenaire du projet Dynamo, soutenu par l'ANR. Les documents métiers des autres partenaires comme les fiches de diagnostic de pannes de voitures et les fiches d'incidents logiciels serviront de collection d'expérimentation.

6. Remerciements

Ces travaux s'inscrivent dans le cadre du projet de recherche Dynamo (Dynamic Ontology for information retrieval) soutenu par l'ANR. Cependant, les éléments présentés dans cet article n'engagent que leurs auteurs et n'est pas un résultat commun du projet Dynamo.

7. Bibliographie

Baeza-Yates, R.A., Navarro, G.: « Block addressing indices for approximate text retrieval». *Journal of the American Society on Information Systems* 51(1), p. 69–82 (2000).

Berners-Lee T., Hendler J., Lassila O., « The Semantic Web », *Scientific American*, p. 28–37, (2001).

Büttcher S., Clarke C.: « A Hybrid Approach to Index Maintenance in Dynamic Text Retrieval Systems ». *28th European Conference on IR Research, (ECIR),*, LNCS 3936, Springer Berlin / Heidelberg, p. 229-240 (2006).

Chang B., Ham D.H., Moon D. S., Choi Y. S., Cha J., «Using Ontologies to Search Learning Resources». *Computational Science and Its Applications – ICCSA 2007*, LNCS 4705, Springer Berlin / Heidelberg, p. 1146-1159, (2007).

Cho, J., Garcia M. H.: « The evolution of the web and implications for an incremental crawler ». *26th Intl. Conf. on Very Large Data Bases*, p. 200-209, (2000).

Cutting D. R., Pedersen and J. O.. « Optimization for Dynamic Inverted Index Maintenance ». *Proceedings of the 13th Annual International ACM SIGIR*. New York, USA ACM Press. P. 405–411, (1990).

Galambos L., « Dynamic Inverted Index Maintenance ». Proceedings of *World Academy Of Science, Engineering and Technology, Vol 11*, ISSN 1307-6884. p. 171-176, (2006).

Hernandez, N., Mothe, J., Chrisment, C., Egret, D.. « Modeling context through domain ontologies ». *Journal of Information Retrieval, Springer, Numéro spécial Contextual Information Retrieval Systems, Vol. 10 N. 2*, p. 143-172, (2007).

Hernandez, N., Mothe, J., Ralalason, B., Ramamonjisoa, B. , Stolf, P.: « A Model to Represent the Facets of Learning Objects ».: *Interdisciplinary Journal of E-Learning and Learning Objects, Informing Science Institute, Santa Rosa - USA, Vol. 4*, p. 65-82, (208).

Lim, L., Wang, M., Padmanabhan, S., Vitter, J.S., Agarwal, R.C.: « Efficient update indexes for dymnamically changing web documents ». Published online: 2 March 2007. http://www.cs.duke.edu/~jsv/Papers/LWP05.landmarkdiff.pdf, (2007).

Manber, U., Wu, S.: GLIMPSE: « a tool to search through entire file systems ». *Proceedings of the Winter 1994 USENIX Conf.*, p. 23–32. (1994).

Page, L., Brin, S.: « The anatomy of a large-scale hypertextual web search engine ». *Proceedings of the 7th Intl.WWW Conf.*, p. 107–117 (1998).

Robertson S. E., Sparck Jones K., « Relevance weighting of search terms ». In: *Journal of the American Society for Information Sciences*, 27 (3), p 129-146, (1976).

Salton, G., Allan, J., Buckley, C.: « Approaches to passage retrieval in full text information systems ». In: *Korfhage, R., Rasmussen, E.M., Willett, P. (eds.) Proceedings of the 16th Annual. Intl. ACM-SIGIR Conf*, p. 49–58. (1993).

Song J. F., Ming Z. W., Dong X. W., Hui L. G., Ning X. Z., « Ontology-based Information Retrieval Model for the Semantic Web ». *International Conference on e-Technology, e-Commerce and e-Service, EEE '05*, p 152 - 155, (2005).

Ukkonen, E.: « Algorithms for approximate string matching ». *Inf. Control 64*, p. 100–118 (1985).

Indexation et représentation comparative : Application au discours électoral

Jacques Savoy

Institut d'informatique
Université de Neuchâtel, rue Emile Argand 11, 2009 Neuchâtel (Suisse)
Jacques.Savoy@unine.ch

RESUME. *Cet article décrit quelques approches afin d'extraire les termes les plus représentatifs d'un site web ou d'un ensemble de documents en comparaison avec d'autres sites ou un corpus de référence. Nous montrons que la fréquence d'occurrence ou le rang des termes les plus fréquents peut fournir une première synthèse. Notre proposition s'appuie sur une distribution binomiale des mots et le calcul d'un score normalisé (score Z) mettant en lumière les termes comparativement les plus appropriés. Quelques exemples tirés des discours électoraux suisses ou français illustrent l'intérêt de l'approche suggérée.*

ABSTRACT. *This paper describes some possible approaches to automatic extraction of terms closely reflecting the content of a Web site or a set of documents by comparison of other sites or a given corpus. We show that the frequency of occurrences or the rank of the most frequent terms may provide a first overview. In the suggested method, we model the terms distribution according to a binomial process and we proposed to compute a normalized z-score to define the most appropriate terms within a comparative perspective. Examples based on Swiss and French political speeches show the usefulness of the suggested method.*

MOTS-CLES : *Résumé automatique, indexation, distribution lexicale, analyse du discours.*

KEY WORDS: *Summarization, Indexing, Probabilistic Word distribution, Discourse analysis.*

1. Introduction

Internet a mis à notre disposition un volume considérable d'information présentée sous divers formats (XML, HTML, pdf), médiums (texte, image, audio, vidéo) et couvrant tous les domaines de l'activité humaine. En recherche d'information (RI), on reconnaît volontiers que les moteurs de recherche ont joué un rôle de premier plan dans l'accroissement exponentiel du nombre de pages disponibles. Pour traiter automatiquement un tel volume d'information et face aux divers besoins des usagers, de nombreuses perspectives d'application s'ouvrent, domaines que l'on peut regrouper sous le terme de “fouille de textes” [KON 06].

Dans ce cadre, nous nous sommes intéressé à la mise au point d'outils automatiques permettant d'extraire les termes (mot isolé, bigramme ou trigramme) les plus caractéristiques d'un site Internet. Comme en RI, notre objectif consiste à indexer et à représenter d'une manière compacte une page, un site ou un ensemble de documents. Dans notre contexte, une telle représentation doit mettre en évidence le contenu sémantique d'un site en comparaison avec d'autres sites voire du même site à une (ou des) date(s) antérieure(s).

Une telle représentation permet de répondre à divers besoins comme le souci des entreprises de suivre en continu l'évolution de leurs concurrents via leurs sites web (veille technologique) ou le suivi d'événements sociaux ou politiques (TDT, *Topic Detection and Tracking*). La *blogosphère* [FAU 08] présente également un intérêt afin de suivre les sentiments ou opinions des internautes exprimés dans des billets d'information par rapport à un groupe de référence ou, dans une perspective dynamique, en mettant en lumière l'évolution temporelle des intérêts ou sentiments.

Dans cette communication, nous souhaitons explorer quelques pistes afin de définir et d'analyser diverses stratégies de représentation comparative. Nous nous limiterons cependant à des informations de nature textuelle pouvant correspondre à une page, à un site Internet ou à un ensemble de documents. Dans cette perspective, quelques travaux reliés à la génération automatique de résumé ou à l'extraction de termes significatifs seront présentés dans la deuxième section. Le domaine d'application, le discours électoral suisse, sera exposé dans la troisième section. Les diverses stratégies d'extraction seront décrites dans la quatrième section. Enfin, basé sur nos outils, une comparaison des discours électoraux en Suisse et une comparaison franco-suisse sera présentée dans la dernière section.

2. Extraction et résumé automatique

Dans la génération automatique de résumé, la phrase constitue la structure fondamentale la plus souvent retenue. En effet, il s'avère souvent trop difficile de comprendre, d'interpréter puis de générer un résumé sur la base de fractions de phrases que le système devra ensuite lier tout en garantissant une bonne lisibilité. Le choix de la phrase permet également de supprimer, partiellement pour le moins, les difficultés liées aux coréférences [STU 96] (e.g., références anaphoriques et pronominales). Bien que des travaux récents recourant à des méthodes sophistiquées aient permis de faire quelques progrès dans cette direction, la génération automatique de résumé peut être vue essentiellement comme un problème d'extraction des phrases les plus significatives.

Dans cette perspective, Goldstein *et al*. [GOL 99] distinguent entre deux types de résumé, à savoir le résumé générique ou en réponse à une requête. Cette distinction s'avère pertinente dans le choix des phrases à extraire du document. Ainsi on sélectionnera soit celles qui décrivent le mieux le contenu proprement-dit ou celles qui répondent le mieux à la requête. D'autres critères de choix peuvent s'ajouter comme la longueur et le style de la phrase mais la sélection s'opère essentiellement

sur la présence et la pondération des termes contenus dans la phrase. On considère généralement que la fréquence d'occurrence (ou fréquence lexicale notée *tf*) et l'inverse de la fréquence documentaire (*idf*) constituent des facteurs déterminants.

En ce qui concerne l'efficacité de la pondération *tf idf*, les expériences menées n'ont toutefois pas abouti à des conclusions toujours concordances [PAI 90]. Ainsi parmi les autres critères intéressants, on pourrait retenir la position de la phrase et sa longueur, deux caractéristiques qui semblent influencer la qualité du résultat final [KUP 95]. Ces auteurs ajoutent que les groupes nominaux fréquents devraient bénéficier d'un avantage, de même que les termes du titre du document, les mots écrits en majuscules ou les entités nommées.

Parfois le flux d'information n'est pas vraiment cohérent et la structure du discours (ou du document) demeure lacunaire, rendant l'extraction de phrases complètes plus ardue et générant souvent un résumé peu cohérent. Dans ce but, Berger & Mittal [BER 00] proposent de déterminer d'abord les mots à inclure dans un résumé selon leur fréquence d'occurrence (*tf*) ou selon leur probabilité d'apparition prédite par un modèle de langue (et nécessitant toutefois un apprentissage). Ensuite, l'ordre des mots dans le résumé final doit être établi en fonction de séquences similaires trouvées dans le (ou les) document(s). Une telle approche peut être appliquée sur des pages *web* ou la *blogosphère* marquée par l'absence de structure argumentative précise.

De manière plus simple, nous pouvons ignorer l'ordre des termes et limiter la représentation à une liste de mots isolés. L'expression “nuage de termes” (*term cloud*) a été suggérée pour une telle représentation décrivant de manière compacte le contenu d'une page (la figure 1 présente un tel exemple). L'importance relative de chaque élément s'illustre par des modifications de la taille de la fonte ou de l'ancre (gras, italique) voire de la couleur, de la police ou de l'emplacement (ou l’ordre). Une telle représentation peut également servir à illustrer le contenu global ou les divers passages d'une séquence audio ou vidéo [FUL 08].

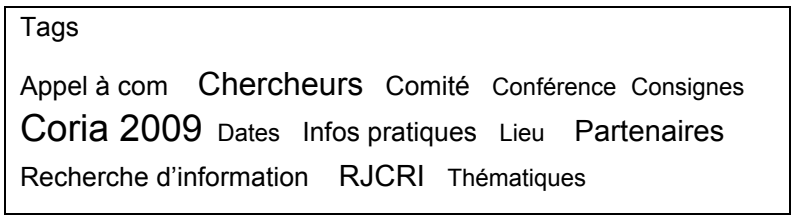


***Figure 1** : Exemple d'un nuage de termes du site CORIA 2009*

3. Champ d'étude et corpus d'évaluation

Dans l'exploration des diverses possibilités offertes afin de représenter de manière *comparative* un ensemble de documents, il nous est apparu important de pouvoir vérifier la qualité des résultats. Malheureusement, nous ne disposons pas

d'un corpus de test et d'une métrique adaptée. Notre démarche se voulant empirique, une justification expérimentale nous est alors parue importante afin de guider nos choix.

3.1. Justification du choix du corpus de référence

Afin de répondre à ces attentes, nous avons sélectionné les sites des partis politiques et, plus précisément, les discours ou programmes électoraux. Diverses raisons justifient notre choix. Premièrement, afin de garantir une grande homogénéité, nous avons repris uniquement des documents électoraux. Ces derniers sont rédigés afin de renforcer la motivation des partisans et de rallier d'autres électeurs, mais indiquent également les principaux choix politiques que les formations entendent suivre durant la prochaine législature. Le choix des mots et des expressions n'est pas le simple fruit du hasard et chaque intervenant prend un soin particulier à rédiger son intervention ou son programme. Les auteurs disposent donc d'une assez grande liberté de choix tant sur le plan du lexique, des formulations, ou des thèmes retenues. En campagne électorale, chaque parti peut insister ou négliger complètement telle ou telle question en recourant aux mots qu'il juge les plus pertinents. A contrario, un président ou un premier ministre doit tenir compte des diverses composantes de sa majorité ce qui entraîne des implications directes dans le choix de ses mots ou de ses formulations.

Deuxièmement, comme ces discours ont été rédigés durant la même période (2007), utilisant la même langue (le français) et désirant atteindre des objectifs similaires, leur comparaison directe s'en trouve facilitée. En effet, il est connu que la comparaison entre des œuvres littéraires de genres différents mais du même auteur sont parfois plus distantes que des œuvres de même genre mais d'auteurs différents [LAB 07]. De plus, des études antérieures portant sur le discours politiques existent [LAB 03], [LAB 08] permettant une comparaison avec nos travaux. Signalons toutefois que ces derniers portent sur le discours gouvernemental et non électoral.

Troisièmement, les documents disponibles seront rédigés avec un souci de garantir une qualité éditoriale. Contrairement à la *blogosphère*, nous ne pensons pas trouver de nombreuses fautes d'orthographe, de syntaxe ou l'usage abusif d'abréviations.

3.2. Acquisition des données

Dans le but de connaître les particularités comparatives du discours électoral suisse, nous avons constitué un corpus en téléchargeant les documents disponibles sur les sites Internet des quatre grands partis suisses. Ces textes présentent la plate-forme électorale en vue des élections fédérales d'octobre 2007 ou des positions du parti dans cette perspective.

Quelques statistiques concernant ce corpus sont reprises dans le tableau 1. Nous y retrouvons la taille des quatre sous-corpus (en octets), le nombre de mots ainsi que le nombre de vocables définis comme le nombre de mots distincts. Enfin, nous avons indiqué le nombre de documents extraits de chaque site et son URL.

	PS	**PDC**	**PRD**	**UDC**
Taille (octet)	236 821	339 047	181 381	612 134
Nb mots	35 846	50 302	26 639	90 559
Nb vocables	4 167	5 238	3 293	7 191
Documents	7	22	9	8
Site internet	www.sp-ps.ch	www.pdc.ch	www.prd.ch	www.svp.ch

***Tableau 1** : Quelques statistiques sur notre corpus*

Nus avons limité notre analyse comparative aux quatre grands partis[1] présents au Conseil fédéral (exécutif) à savoir, en partant de la droite, l'UDC (union démocratique du centre), le PRD (parti radical démocratique), le PDC (parti démocrate-chrétien) et le PS (parti socialiste). Ces formations disposent d'un nombre variable d'élus sous la Coupole fédérale. Ainsi après les élections d'octobre 2007, l'UDC dispose de 69 représentants sur 246 correspondants à 28,1 % des élus. Ce parti d'une droite dure et populiste constitue la formation ayant le plus progressée lors des dernières élections (dont un gain de six sièges en 2007). Le PS reste la deuxième force politique du pays avec 52 élus (ou 21,1 %) tandis que le parti du centre, le PDC disposera de 43 élus (ou 18,7 %). Le PRD représentant la droite modérée doit faire face à un recul (perte de sept sièges par rapport à 2003 pour un total de 43 élus soit 17,5 %).

3.3. Prétraitement du corpus

Afin de définir une représentation comparative de chaque site (ou parti dans notre cas), nous avons choisi de tenir compte des termes présents ou absents. Lors de la segmentation des documents, nous avons considéré comme mot toute séquence de lettres et / ou de chiffres. Cette définition laisse quelques imperfections. Ainsi la forme "chemin de fer" sera analysée comme trois mots et les termes "ne … pas" ou "parce que" mériteraient d'être comptés sous une entrée unique. D'autre part, les formes "l'école" ou "aujourd'hui" seront vue comme composée de deux mots.

Notre système d'analyse ne fera pas de distinction entre majuscule et minuscule et les formes "Suisse" ou "suisse" seront considérées comme identiques. Certes, les

[1] La Suisse connaît plusieurs partis de taille plus réduite dont, entre autres, le parti écologiste suisse (véritable cinquième force apparue dans les années quatre-vingt et qui renouvelle la gauche), le parti évangélique et le parti libéral. Ce dernier a fusionné avec le parti radical démocratique en octobre 2008.

formes “poste” et “Poste” correspondent à deux entités sémantiques distinctes dans la phrase “le poste ouvert à la Poste”. Toutefois, si un mot s’écrit exclusivement avec des majuscules, nous avons conservé cette forme en l’état car elle correspond souvent à un acronyme (UE, PS, ONU).

Nous n’avons pas effectué une analyse morphologique poussée afin de déterminer pour chaque mot son entrée dans le dictionnaire (lemmatisation). Dans notre cas, les formes “peux”, “pouvons” ne seront pas regroupées sous le même vocable “pouvoir”. Remarquons que ce dernier peut être ambigu et que le contexte précisera s’il s’agit du nom ou du verbe. Nous avons toutefois appliqué un enracineur léger [SAV 02] supprimant la marque du pluriel (le ‘-s’ final ou la transformation de la séquence finale ‘-aux’ en ‘-al’). L’application de cette procédure de normalisation nous a permis de réduire le nombre de vocables de 13 008 à 11 011 (soit une différence de 15,4 %). Parfois la forme au singulier ou au pluriel s’avère aussi fréquente l’une que l’autre (e.g. dans le discours de l’UDC le vocable “rente” (76 occurrences) ou “rentes” (fréquence de 71)) mais plus souvent une des formes tend à dominer (e.g. le vocable “enfants” (130 occurrences) comparée à “enfant” (9) auprès du parti PDC). En règle générale, nous pensons que les calculs effectués divergent quelque peu par rapport à une lemmatisation complète mais les conclusions que nous en tirons devraient demeurer identiques, très similaires pour le moins.

Finalement, nous garderons à l'esprit que le choix d’un vocabulaire sera sujet à des variations dues aux circonstances (le contexte, l’auditoire, intervention spontanée ou discours lue) ainsi qu’au type de communication choisi (programme général ou discours traitant une question particulière ou technique). Dans le contexte présent, ces diverses variations sont relativement neutralisées dans notre corpus. En effet, les textes proviennent de la même période, sont rédigés dans la même perspective et couvrent des objectifs très similaires.

4. Quelques stratégies d'extraction

L'indexation automatique en RI propose de définir l'importance de chaque terme des documents (et des requêtes) en tenant compte essentiellement de leur fréquence d'apparition (*tf*), de fréquence documentaire d'un terme (*df*, ou plus précisément de l’*idf* = log(*n*/*df*)) et de la longueur du document. Dans notre contexte, le nombre de documents ou de sites distincts demeure faible ($n = 4$ dans le cas présent) réduisant l'intérêt pour une mesure *idf*. En effet de nombreux mots apparaissent dans les quatre sites et leur valeur *idf* sera donc nulle. La suite de cette section examine les possibilités d'extraire les éléments comparatifs d'un site (ou ensemble de documents) en fonction d'autres sites (ou d'un corpus de référence).

4.1. Richesse lexicale et vocabulaire

Comme première approche, nous pourrions comparer la taille du vocabulaire utilisé dans les quatre sites. Dans cette perspective, nous pouvons estimer qu’un

lexique étendu correspond à un parti ayant de grandes ambitions, désirant couvrir tous les domaines [LAB 03]. A contrario, en présence d'un vocabulaire plus restreint, nous pourrions avancer l'hypothèse que le parti a opté pour la sobriété, pour un parler simple et direct, une communication qui se veut plus proche du peuple et dans un souci d'éviter toute formulation trop sophistiquée. Cependant, cette analyse doit s'effectuer sur un ensemble de documents possédant la même taille ou, à défaut de longueur très similaire. En effet, un corpus possédant un volume plus important proposera également un vocabulaire plus ample et sera donc ainsi favorisé [BAA 01]. Comme l'indique le tableau 1, les quatre grands partis présentent des volumes assez différents. Comme le PRD propose le corpus le moins long, nous avons réduit les trois autres corpus à cette taille en ne retenant que les 26 639 premières formes (voir tableau 2).

	PS	**PDC**	**PRD**	**UDC**
Nb de mots	26 639	26 639	26 639	26 639
Nb vocables	3 412	3 682	3 293	3 899
Hapax	1 676	1 811	1 511	1 964
Hapax en %	49,1 %	49,2 %	45,9 %	45,3 %

***Tableau 2** : Richesse lexicale en nombre de vocables (forme distincte) et nombre de vocables apparaissant qu'une seule fois (hapax)*

Si l'on compare, en prenant le même nombre de mots (soit 26 639), le nombre de formes différentes (vocables) utilisées par les quatre grands partis suisses, le vocabulaire le plus étendu se rencontre auprès de l'UDC avec 3 899 formes, suivi par le PDC (3 682 vocables), puis le PS (3 412) et, finalement, le PRD (3 293). Le parler simple et direct serait l'apanage du PRD tandis que les grandes ambitions et la couverture la plus large seraient plutôt du côté de l'UDC. D'un autre côté, si la rareté des mots serait un indice de la richesse lexicale avec des expressions savantes n'apparaissant qu'une seule fois, l'UDC remporte de nouveau le premier rang avec 1 964 vocables apparaissant qu'une seule fois (*hapax legomena*) contre 1 811 pour le PDC, 1 676 au PS et 1 511 au PRD. Ces informations nous fournissent un indice lexical global mais pas une représentation de la sémantique sous-jacente.

4.2. Fréquence d'occurrence

Afin de déterminer les mots nécessaires à refléter le contenu sémantique d'un document, on peut recourir à la fréquence d'apparition (*tf*). Parmi les formes très fréquentes, nous pouvons alors observer les mêmes termes et ceci quelle que soit la formation politique. Un regard plus attentif révèle que ces vocables abondants correspondent à des mots outils (de, la, les, l, et, des, le, d, en, une, dans, du est, pour, que, un, etc.). Après élimination de 64 termes peu porteurs de sens, nous voyons mieux émerger les thèmes récurrents et communs à l'ensemble des formations et par différence, ceux propres à chaque parti.

Le tableau 3 indique pour chaque parti les dix vocables les plus fréquents. A côté de chaque forme, nous avons noté sa fréquence d'occurrence (*tf*). Nous pouvons constater que certains mots apparaissent fréquemment dans les quatre discours comme "politique", "suisse", "être", "ne" et "pas" ou, dans trois des quatre, comme "doit" ou "nous". Certaines formes apportent peu d'information (être, ne, pas, doit, nous) tandis que d'autres laissent clairement voir l'origine commune du corpus (politique, suisse). Si l'on analyse le vocable "suisse", on constate que son rang diverge entre les partis. Pour l'UDC, ce terme s'avère le plus usité avec une fréquence d'occurrence (864) nettement supérieure au deuxième vocable le plus fréquent (vocable "pas", fréquence de 456). Pour les deux partis du centre-droit, ce terme "suisse" apparaît au deuxième rang, tandis que ce vocable semble moins utilisé au PS (septième rang).

PS		PDC		PRD		UDC	
tf	vocable	*tf*	vocable	*tf*	vocable	*tf*	vocable
237	nous	643	nous	178	être	864	suisse
198	politique	347	suisse	176	suisse	456	pas
192	doit	261	pas	166	doit	445	politique
190	pas	245	être	143	politique	384	ne
178	ne	230	notre	138	nous	323	être
150	être	222	ne	108	sécurité	321	état
133	suisse	177	politique	108	ne	320	AI
132	culture	174	PDC	91	pas	295	droit
106	culturelle	156	doit	90	doivent	286	UDC
104	sociale	144	formation	88	armée	248	étranger

***Tableau 3** : Les dix vocables les plus fréquents (et leur fréquence absolue) dans les discours des quatre partis*

Parmi les dix vocables les plus fréquents, nous pouvons voir se dessiner des tendances propres à chaque formation politique. Ainsi le PS semble se distinguer par l'usage fréquent du vocable "culture" ainsi que la forme reliée "culturelle" tandis que le PDC recourt volontiers à "formation". A droite, le PRD se positionne sur les thèmes de la "sécurité" et de l'"armée" tandis que l'UDC insiste sur le "droit", "étranger" et les problèmes de l'assurance-invalidité (AI).

Le recours à la fréquence d'occurrence permet de faire ressortir les vocables décrivant bien le contenu sémantique d'un document. De plus cette information peut définir l'importance de chaque terme dans une représentation (e.g., par des variations de fonte, taille ou style dans un interface de type "nuage de termes"). On prendra soin toutefois de normaliser cette fréquence lexicale en fonction de la longueur du document sous-jacent (e.g., *tf*/max *tf*).

Comme alternative, on peut retenir la fréquence lexicale uniquement comme clé de tri. L'importance de chaque terme se mesurerait alors en fonction uniquement de leur rang. Ce second choix ne s'avère pas toujours très satisfaisant car une différence de rang unitaire peut cacher des situations très différentes[2]. Par exemple, la différence de rangs entre les vocables “pas” et “suisse” pour le parti UDC (voir tableau 3) s'élève à 1 tandis que la différence de fréquence se monte à 408 (864 - 456) ou de manière relative à 0,472 (= 864/864 - 456/864).

4.3. Représentation comparative

Si l’on désire connaître le vocabulaire spécifique à un parti, nous devons le comparer à une norme, par exemple à un corpus composé des sites des quatre partis. Comme le tableau 3 l'indique, plusieurs vocables apparaissent fréquemment dans plusieurs sites et ne permettent donc pas de discriminer de manière comparative les contenus. Avec un corpus de référence, nous pouvons observer quel vocable apparaît de manière significativement plus fréquente dans l’un ou l’autre des discours ou, inversement, ceux dont l’occurrence s’avère significativement moins forte. Pour atteindre cet objectif, nous nous sommes inspirés de la méthode proposée par Muller [MUL 92].

Si nous regroupons tous les documents ou divers sites que l'on désire comparer, nous pouvons former un corpus **C**. Si nous désirons définir une représentation comparative par rapport à **C**, nous pouvons extraire les documents correspondant à un sous-ensemble noté **S**. Le reste du corpus sera noté **C-** (avec $\mathbf{C} = \mathbf{S} \cup \mathbf{C}$-).

Intéressons nous à un vocable particulier noté ω. Nous pouvons alors compter sa fréquence d'occurrence dans le sous-ensemble **S** (valeur notée *a* dans le tableau 4) et sa fréquence dans le reste du corpus **C-** (valeur notée *b*). Evidemment, la fréquence d'apparition dans tout le corpus **C** sera de $a + b$. De manière similaire, nous pouvons compter la fréquence lexicale de tous les autres vocables dans le sous-ensemble **S** (valeur notée *c*) et le reste du corpus **C-** (fréquence notée *d*). Le corpus **C** comprendra donc *n* mots avec $n = a + b + c + d$.

	S	**C-**	
ω	a	b	$a + b$
non ω	c	d	$c + d$
	$a + c$	$b + d$	$n = a + b + c + d$

***Tableau 4** : Exemple d'une table de contingence pour le vocable ω*

Sur la base des informations données dans le tableau 4, nous faisons l’hypothèse que le mot ω suit une distribution binomiale dans le sous-ensemble **S** avec comme

[2] La distribution de la fréquence d'occurrence suit une loi de puissance [BAA 01].

paramètres p et n'. Le paramètre p indiquant la probabilité d'apparition du terme ω qui peut être estimé par $(a+b)/n$ tandis que $n' = a+c$ correspond à la taille (en nombre de mots) du sous-ensemble **S**. Selon cette distribution binomiale, le nombre moyen d'occurrence du terme ω dans la partie **S** sera donc de $n' \cdot p$. Pour un vocable donné ω, le tableau 4 indique que le nombre observé dans le sous-ensemble **S** s'élève à a. Enfin, l'écart-type de la distribution binomiale s'estime par $n' \cdot p \cdot (1-p)$.

Sur ces éléments, nous pouvons calculer un score normalisé (ou score Z) pour chaque terme ω en tenant compte du nombre observé d'occurrences (valeur a) auquel on soustrait sa moyenne théorique (nombre prédit par la distribution binomiale) et que l'on divise par l'écart-type estimé.

$$\text{score Z}(\omega) = \left[\frac{a - n' \bullet p}{\sqrt{n' \bullet p \bullet (1-p)}} \right] \quad (1)$$

La différence entre le nombre réellement observé a et la moyenne théorique $(n' p)$ permet de déterminer les suremplois (différence positive) et les vocables sous-représentés (différence négative). Cette différence doit encore être divisée par un estimateur de l'écart-type pour retourner une valeur standardisée. Comme règle de décision, un score Z supérieur à 3 indiquera un suremploi significatif[3] du vocable dans le sous-corpus tandis que des valeurs négatives et inférieures à -3 signalent des sous-emplois marqués par rapport au corpus global.

5. Applications

Sur la base des discours électoraux suisses, nous avons pu déterminer les termes caractéristiques des quatre formations (section 5.1). Comme l'année 2007 a également connu l'élection présidentielle française, nous avons décidé de comparer les discours électoraux tenus dans les deux pays. Dans ce but, nous avons récupéré 11 discours prononcés par S. Royal et 17 prononcés par N. Sarkozy[4] (section 5.2). Une différence de style existe tout de même entre les deux pays. Du côté français, nous sommes en présence de discours prononcés tandis que du côté suisse nous avons une forme écrite, variation de forme qui peut influencer le choix du lexique.

5.1. Application aux discours des partis suisses

Sur la base de notre méthodologie décrite en section 4.3, nous avons déterminé les vocables sur-employés pour chaque formation politique (tableau 5) ainsi que ceux qui sont sous-représentés (tableau 6). Pour établir ces listes ordonnées, nous

[3] En admettant que la valeur z suit une distribution normale, les valeurs excédant les limites de 3 et -3 représentent 0,3 % des cas. En descendant cette limite à 2 et -2, on trouverait théoriquement 4,6 % des observations.

[4] Ces deux sous-corpus représentent respectivement 93 479 mots pour S. Royal et 116 212 mots pour N. Sarkozy, soit une taille un plus élevée que celle des discours suisses (tableau 1).

avons calculé le score z normalisé de chaque terme, valeur indiquée à côté de chaque entrée dans les tableaux 5 et 6.

PS		**PDC**		**PRD**		**UDC**	
Z	vocable	Z	vocable	Z	vocable	Z	vocable
15,2	état	21,8	nous	18,9	PRD	14,6	AI
14,0	II	18,9	PDC	16,0	radical	13,2	UDC
13,0	culture	11,8	demandons	12,2	mission	11,3	neutralité
11,9	culturelle	10,4	énergie	12,0	armée	10,0	gauche
11,7	artiste	10,1	internet	11,7	défense	9,6	naturalisation
10,3	encouragement	9,1	enfant	11,3	sécurité	9,0	rente
10,1	art	9,1	notre	9,6	militaire	8,8	état
10,0	autogestion	8,9	énergétique	9,6	*easy*	8,7	nationalité
10,0	CO2	8,2	thème	9,5	imposition	8,0	milliard
9,5	pro	8,1	jeune	9,2	*tax*	7,4	suisse

***Tableau 5** : Les dix vocables les plus surreprésentés dans les sites des partis suisses*

Les vocables apparaissant dans le tableau 5 forment une représentation comparative des thèmes privilégiés par chaque formation. Elle s'avère plus parlante et pertinente qu'une représentation s'appuyant uniquement sur la fréquence d'occurrence (voir tableau 3). Pour l'extrême droite (UDC), les thèmes récurrents sont les assurances sociales (“rentes”, “AI”), la politique de naturalisation, la neutralité de la Suisse et la défense de son identité face à l'étranger, l'affectation des ressources financières (“milliard”, “franc” (11e)) mais également le souci de se distinguer de la “gauche”. Pour la droite modérée (PRD), les sujets touchant la sécurité (“armée”, “défense”, “sécurité”, “mission”, “militaire”) forment une thématique centrale ainsi que les questions d'imposition fiscale (“*easy*” et “*tax*” dans l'expression “*easy swiss tax*”). Le parti du centre (PDC) axe son discours sur la famille (“enfant”) mais de manière un peu surprenante sur l'énergie et l'environnement (“énergie”, “énergétique”) d'une part et, d'autre part, sur la technologie (“internet”, “technologique” (11e) “électronique”(12e)). Le parti socialiste (PS) semble se caractériser par sa politique culturelle (“culture”, “artiste”, “art” ou “pro” dans la dénomination “pro helvetia”) mais également par une préoccupation écologique (taxe sur le “CO2”) à côté d'un thème plus traditionnel (“autogestion”)[5].

Ces vocables ne sont pas forcément très fréquents. Ainsi, le terme “easy” apparaît 16 fois et uniquement dans les discours du PRD, tandis que l'on compte 26 occurrences du terme “autogestion” utilisé uniquement dans le programme du PS.

[5] Le vocable « II » surreprésenté dans le discours du PS correspond au pronom « il ». Pour une raison inconnue, la forme « II » a été substituée au pronom « il » dans les documents disponibles sur Internet et décrivant la plate-forme de ce parti.

On se gardera d'en tirer la conclusion que les termes surreprésentés apparaissent seulement auprès d'un auteur. Ainsi, on compte 19 occurrences du vocable "tax" mais 17 fois dans le discours du PRD ce qui est fait un terme sur-employé pour cette formation.

De manière duale, les vocables peu usités dans chaque formation politique permettent de compléter ces conclusions (voir tableau 6). Ainsi, on constate que les sigles des autres partis ne sont que très peu fréquents dans le discours d'un parti donné. On ne compare pas son programme avec les autres et on se garde bien de mentionner les autres à l'exception de l'UDC avec ses vocables "PS" et "gauche". Les termes "neutralité" ou "AI" sont visiblement des termes propres à l'UDC. Etonnamment, le terme "neutralité" est sous-employé par le parti PRD dont l'un des thèmes majeurs concernait la sécurité et l'armée. Enfin, le PS n'utilise que fort peu le terme "suisse" ou "état" mais également "UE", lui qui est le seul parti à souhaiter l'ouverture de négociations en vue de l'adhésion à l'UE.

PS		**PDC**		**PRD**		**UDC**	
Z	vocable	Z	vocable	Z	vocable	Z	vocable
-8,2	suisse	-8,7	AI	-6,3	UDC	-17,2	nous
-8,1	état	-7,2	neutralité	-5,9	AI	-8,1	PDC
-7,6	AI	-7,2	UDC	-5,5	gauche	-7,5	notre
-6,9	UDC	-7,2	culture	-5,2	culture	-6,4	voulons
-5,9	gauche	-6,9	culturelle	-5,1	franc	-5,9	économique
-5,6	enfant	-6,5	armée	-4,9	PDC	-5,7	demandons
-5,3	jeune	-6,0	naturalisation	-4,9	PS	-5,5	formation
-5,3	neutralité	-6,0	rente	-4,6	ont	-5,4	état
-5,2	école	-5,4	PS	-4,5	neutralité	-5,3	II
-5,1	UE	-5,3	nationalité	-4,5	année	-5,1	cadre

***Tableau 6** : Les dix vocables les plus sous-employés dans les sites des partis suisses*

5.2. Comparaison Suisse - France

Entre les deux pays, les vocables fréquents montrent l'origine politique des documents mais des distinctions apparaissent également rapidement (voir tableau 7). Ainsi, on retrouve les formes "ne", "pas", "être" ou "nous" des deux côtés de la frontière, sans que ces éléments relèvent des informations très pertinentes. Nous aurions pu ajouter le vocable "politique" qui apparaît en onzième rang du côté français. De même, le recours différencié aux formes "suisse" et "france" ou "français" ne nous surprennent pas.

Parfois les petits mots font toute la différence et dans ce cas nous rencontrons un emploi marqué du "je" (ainsi que du vocable relié "j") dans les discours électoraux

français par rapport à ceux de la Suisse. Ce vocable "je" indique bien l'importance attachée à une personne, au chef du parti ou futur président dans l'Hexagone. Plus étonnant, la fréquence d'occurrence du pronom "je" s'avère statistiquement plus élevée pendant le deuxième tour de la campagne présidentielle que lors du premier tour [LAB 08b]. Le passage au second tour s'accompagne bien d'un changement au niveau lexical et cela se vérifie pour les deux candidats. Pour un chef de parti, une campagne ne forme pas un continuum lexical stable, mais des ruptures peuvent apparaître. En fin de course, il faut serrer les rangs autour du "moi", du chef qui insistera sur le "je veux". Ce vocable peut également s'expliquer, en partie, par la forme orale du discours électoral français.

	Suisse	**France**	**S. Royal**	**N. Sarkozy**
1	suisse	je	je	je
2	nous	pas	nous	pas
3	pas	ne	vous	ne
4	politique	nous	pas	france
5	être	france	france	nous
6	ne	vous	ne	veux
7	doit	veux	veux	si
8	droit	si	j	parce
9	notre	parce	notre	être
10	doivent	être	faire	français

***Tableau 7** : Les dix vocables les plus fréquents en Suisse et en France*

En complément, nous avons également calculé les quinze vocables sur-employés dans les discours des deux pays de même que les suremplois des deux candidats à l'Elysée (voir tableau 8). En premier lieu, on y retrouve les dénominations propres à chaque pays ("suisse", "france") de même que ceux rattachées à leurs institutions respectives ("canton", "conseil", "fédérale", "confédération", "UDC", "PDC" d'une part et, d'autre part, "présidentiel", "république"). De manière plus profonde, on retrouve, du côté français, les formes verbales "veux", "suis", "dire", "crois" (en 20^{e} rang) ou la conjonction "parce que" indiquant un besoin explicatif indéniable, phénomène qui s'explique, en partie, par le fait que le discours était oral. Du côté helvétique, les formes verbales abondantes sont "doit" ou "doivent" soulignant les obligations ou attentes ("l'Etat doit"), le besoin de quantifier ("franc" (7^{e} rang), "milliard" (18^{e} rang)) ainsi que les vocables "assurance" (17^{e} rang), "économie" (24^{e} rang) ou "culturelle" (27^{e} rang).

Il est également intéressant de noter que l'acronyme "UE" s'avère sur-employé dans le discours politique suisse. Les deux candidats à l'Elysée n'ont pas retenu cette forme et ont préféré parler d'"Europe", forme apparaissant au 41^{e} rang des

vocables les plus fréquents chez S. Royal et en 88e rang chez N. Sarkozy[6]. Ces résultats démontrent l'intérêt de la méthode proposée (tableau 8) en regard du recours à la fréquence d'occurrence absolue ou normalisé (tableau 7).

Suisse		**France**		**S. Royal**		**N. Sarkozy**	
Z	vocable	Z	vocable	Z	vocable	Z	vocable
28,1	suisse	39,8	je	29,1	vous	32,2	je
14,8	fédéral	26,4	france	23,7	je	25,4	pas
13,0	AI	23,0	vous	18,1	pacte	24,0	france
12,6	confédération	20,9	veux	13,8	sera	21,9	veux
12,5	UDC	17,3	français	13,7	salarié	20,8	français
12,4	étranger	16,8	parce	12,9	présidentiel	20,0	parce
11,6	franc	16,6	j	12,8	france	19,6	ne
11,4	doivent	16,2	pas	12,0	oui	15,9	république
10,7	canton	15,2	ai	11,9	femme	14,4	si
10,5	doit	13,1	république	11,6	logement	13,5	ai
10,4	neutralité	12,4	ne	11,1	construire	12,9	j
10,4	UE	12,3	suis	11,1	juste	12,5	ceux
10,3	conseil	12,0	dire	10,5	j	12,4	parler
10,0	fédérale	11,8	me	10,4	emploi	12,3	suis
9,8	PDC	11,6	ceux	10,3	nous	12,3	rien

***Tableau 8** : Les quinze vocables les plus surreprésentés de manière significative dans les deux pays*

6. Conclusion

Afin de déterminer les termes comparativement les plus représentatifs, nous avons étudié la possibilité de recourir à la fréquence lexicale ou au rang correspondant. Les résultats permettent certes de se faire une idée du contenu d'un site, d'un document ou d'un ensemble de documents. Toutefois, cette approche ne dispose pas d'une règle de décision claire et ne permet pas de distinguer entre les vocables fréquents et ceux qui caractérisent comparativement un sous-ensemble donné.

Comme approche, nous proposons de recourir à un score normalisé (score z) sous l'hypothèse que la distribution des termes suit une loi binomiale. En appliquant cette méthode pour l'analyse comparative des discours électoraux en Suisse (élection

[6] Le vocable « europe » apparaît en 282e position des formes les plus fréquentes des discours politiques suisses.

d'octobre 2007), nous pouvons mettre en lumière les thèmes porteurs et caractéristiques des quatre grandes formations suisses. Il en ressort clairement que le parti de la droite dure et populiste (UDC) situe son débat autour de la peur de "l'étranger", du refus de toute "naturalisation" facilitée, des "rentes" de l'assurance-invalidité "AI", ainsi qu'une volonté de maintenir une vision très stricte de la "neutralité". Le parti PRD de la droite modérée axe sa thématique sur la réforme fiscale tandis que le parti du centre (PDC) garde son orientation "enfant" et "famille". La gauche tient quelques positions classiques ("autogestion" ou "sociale") mais s'ouvre vers les problèmes de l'environnement (taxe sur le "CO2"), thème central de leur alliée, le parti écologiste.

Par rapport à la campagne présidentielle française (avril - mai 2007), les discours politiques suisses ne recourent que très peu à la forme "je" ou aux vocables "veux", "dire", "crois" ou "parce que" dénotant l'importance du chef unique et un souci explicatif indéniable du côté français. Le discours helvétique met un accent plus important sur les formes verbales "doit" ou "doivent" soulignant plutôt les obligations ou attentes ("l'Etat doit").

Enfin, la méthode proposée permet également de traiter des bigrammes ("adhésion UE", "taxe CO2" ou "pacte présidentiel") ou trigrammes ("camp rouge-vert", "nous autres radicaux" ou "je m'engage") permettant peut-être de mieux refléter la sémantique sous-jacente. De plus, si nous avons retenu les formes de surface, nous pourrions sans difficulté appliquer la même approche sur des lemmes. Dans ce cas, des formes différentes mais reliées au même lemme seraient réunies sous la même entrée. La solution proposée demeure simple à appliquer et ne requiert pas de corpus d'entraînement (pour construire un modèle de langue).

Remerciements

Cette recherche a été financée en partie par le Fonds national suisse pour la recherche scientifique (subside n^0 200021-124389).

Bibliographie

[BAA 01] Baayen, H.R. "*Word Frequency Distributions*", Kluwer Academic Publishers, Dordrecht, 2001.

[BER 00] Berger, A.L. & Mittal, V.O. "OCELOT: A system for summarizing web pages", Proceedings *ACM-SIGIR'2000*, p. 144-151.

[FAU 08] Fautsch, C. & Savoy, J. "Stratégies de recherche dans la blogosphère", *Document Numérique*, vol. 11, n° 1-2, 2008, p. 109-132.

[FUL 08] Fuller M., Tsagkias M., Newman E., Besser J., Larson M., Jones G.J.F. & de Rijke M. "Using term clouds to represent segment-level semantic content of podcasts", Proceedings *2nd SIGIR Workshop on Searching Conversational Speech*, Singapore, 2008.

[GOL 99] Goldstein, J., Kantrowitz, M., Mittal, V. & Carbonell, J. "Summarizing text documents", Proceedings *ACM-SIGIR'99*, p. 121-128.

[KIL 07] Kilgarriff, A. "Googleology is bad science", *Computational Linguistics*, vol. 33, nº 1, 1991, p. 147-151.

[KON 06] Konchady, M. "*Text Mining Application Programming*", Ch. River, Boston, 2006.

[KUP 95] Kupiec, J., Pedersen, J. & Chen, F. "A trainable document summarizer", *Proceedings ACM-SIGIR'95*, p. 68-73.

[LAB 03] Labbé, D. & Monière, D. "*Le discours gouvernemental. Canada, Québec, France (1945-2000)*", Champion, Paris, 2003.

[LAB 07] Labbé, C. & Labbé, D. "Baudelaire, Rimbaud et Verlaine", Actes *Aspects linguistiques du texte poétique*, Brest, 2007.

[LAB 08] Labbé, D. & Monière, D. "*Les mots qui nous gouvernent. Le discours des premiers ministres québécois : 1960-2005*", Monière-Wollank, Montréal., 2008.

[LAB 08b] Labbé, D. & Monière, D. "Je est-il un autre ?", Actes *JADT 2008* (Journées internationales d'Analyse statistique des Données Textuelles), 2008, p. 647-656.

[MAN 99] Mani, I. & Maybury, M.T. "*Advances in Automatic Text Summarization*", The MIT Press, Cambridge, 1999.

[MUL 92] Muller, C. "*Principes et méthodes de statistique lexicale*", Honoré Champion, Paris, 1992.

[PAI 90] Paice, C.D. "Constructing literature abstracts by computer: techniques and prospects", *Information Processing & Management*, vol. 26, nº 2, 1990, p. 171-186.

[SAV 02] Savoy, J. "Recherche d'informations dans des corpus en langue française : Utilisation du référentiel Amarylis", *TSI,* vol. 21, nº 3, 2002, p. 345-373.

[STU 96] Strube, M. & Hahn, U. "Functional centering", Proceedings of *Association for Computational Linguistics*, Morgan Kaufmann, 1996, p. 270-277.

Annexe : Liste de mots-outils ignorés

a	ces	est	n	s
à	cet	et	ni	se
ainsi	cette	été	on	soit
au	ci	il	ont	sont
aussi	comme	ils	or	sur
aux	d	l	ou	tous
avec	dans	la	p	tout
c	de	le	par	toute
car	des	les	plus	toutes
ce	donc	leur	pour	un
ceci	du	leurs	qu	une
cela	elle	mais	que	y
celle	en	même	qui	

***Tableau A.1** : Liste des 64 mots-outils éliminés avant de procéder à nos analyses*

Chapitre 5

Apprentissage et *Clustering*

Catégorisation automatique de pages web chinoises

Documents spécialisés *vs* grand public sur le tabagisme

Guiyao Ke, Pierre Zweigenbaum

LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France
pz@limsi.fr

RÉSUMÉ. *La catégorisation (ou classification supervisée) de textes concerne généralement le thème traité ou le type de document. Nous nous intéressons ici à une dimension particulière, le public visé, en distinguant deux grandes catégories : textes destinés au grand public, et textes destinés à des spécialistes du domaine traité. Nous testons la catégorisation, selon cette opposition, de pages web en langue chinoise sur le thème du tabagisme. Dans ce contexte, nous obtenons les conclusions suivantes : une segmentation des textes chinois en mots plutôt qu'en sinogrammes n'améliore pas la catégorisation mais facilite son interprétation ; des attributs supplémentaires relevés à la lecture humaine du corpus n'améliorent pas la catégorisation ; un arbre de décision ou un SVM sont plus performants sur un corpus de test proche du corpus d'entraînement ($F1 = 98,5$ %) que Naïve Bayes ou Kppv ; les Kppv ou un arbre de décision sont plus performants sur un corpus de test plus éloigné ($F1 = 93,4$ %) ; une sélection automatique d'un sous-ensemble des attributs est performante sauf pour les SVM.*

ABSTRACT. *Text categorization (or supervised classification) generally addresses the topic or the type of a text. We tackle here a different dimension, the intended audience, contrasting two broad categories: texts intended for the general public, or texts intended for specialists. We test the categorization, according to this contrast, of Chinese Web pages about smoking. In this context, we obtain the following conclusions: segmenting Chinese texts into words instead of into sinograms does not improve categorization but facilitates the interpretation of results; additional attributes elicited after human reading of the corpus do not improve categorization; decision trees or SVM outperform ($F1 = 98.5\%$) Naïve Bayes or kNN on a test corpus close to the training corpus; kNN or decision trees outperform ($F1 = 93.4\%$) Naïve Bayes or SVM on a more distant test corpus; automatic feature selection improves results except for SVM.*

MOTS-CLÉS : *Catégorisation de textes ; chinois ; public visé.*

KEYWORDS: *Text categorization; Chinese; Intended audience.*

1. Introduction

La recherche d'information sur l'internet ramène des pages web de types variés ; plusieurs moteurs de recherche ont proposé de regrouper les pages ramenées par classes homogènes, généralement thématiques. Un courant de travaux s'est intéressé à la catégorisation [1] des genres de documents (Karlgren 2000), et plus particulièrement de ceux trouvés sur le web. Par exemple, (Rosso 2005) cherche à caractériser une typologie de genres pour améliorer la recherche d'information sur le web ; et (Santini 2007) s'intéresse à sept genres du web : blogs, boutiques en lignes, FAQ, pages d'accueil, listes (annuaires), pages personnelles, pages de recherche.

Le public visé par un document est l'une des composantes qui entrent en jeu dans la définition d'un genre (Sinclair & Ball 1996, Lee 2001). De nombreux travaux, dont ceux cités plus haut, le mentionnent comme attribut à prendre en compte pour la catégorisation de documents. Nous n'avons cependant pas trouvé de travaux qui proposent de déterminer automatiquement le public visé par un document à partir du contenu de ce document, par exemple de distinguer des pages web destinées au grand public de celles destinées à des spécialistes. C'est l'un des aspects abordés par le projet C-Mantic [2], qui cherche à mettre en œuvre des méthodes de sémantique textuelle pour caractériser certains constrastes dans des documents du web. Le projet s'est fixé un thème, il s'agit du tabagisme, et différents contrastes à explorer : opposition grand public / spécialistes, pro- / anti-tabac, etc. Il a prévu de travailler sur trois langues : le français, l'anglais et le chinois. Nous nous intéressons ici à l'opposition grand public / spécialistes dans des textes chinois. La catégorisation de pages web chinoises a elle aussi fait l'objet de travaux depuis plusieurs années (par exemple, (He *et al.* 2000)), mais de nouveau, nous n'avons pas trouvé d'étude de la catégorisation du public visé.

Pour préparer un point de comparaison pour les méthodes ultérieures de sémantique textuelle, nous avons cherché à mettre en œuvre des méthodes de catégorisation automatique de textes (Sebastiani 2002) fondées sur une approche « sac de mots ». Les questions posées par ce travail sont les suivantes : une catégorisation du public visé par un texte peut-elle utiliser les méthodes classiques de la catégorisation en thèmes ; quels attributs employer ; quelles méthodes de catégorisation automatique sont les plus appropriées ? D'autre part, le chinois pose un problème spécifique de segmentation en mots, que nous présentons plus bas : une segmentation en mots est-elle nécessaire, ou une simple segmentation en sinogrammes (caractères chinois) unitaires suffit-elle ?

Dans le reste de cet article, nous commençons par présenter la préparation des corpus sur lesquels nous avons travaillé (section 2), puis les attributs retenus et les

1. Nous employons le terme « catégorisation » pour désigner la répartition d'objets dans un ensemble de classes prédéterminées (classification supervisée), et « catégoriseur » pour désigner un algorithme ou un programme qui réalise cette opération.

2. Ces travaux ont été partiellement financés par le projet C-Mantic (ANR-07-MDCO-002) coordonné par l'ERTIM, Inalco, et auquel collaborent le LINA, le LIMSI et l'Inserm UMR_S 872.

méthodes de catégorisation mises en œuvre (section 3). Nous commentons les résultats obtenus (section 4) et les discutons (section 5) puis concluons (section 6).

2. Préparation du corpus

2.1. *Constitution d'un corpus de textes chinois sur le tabagisme*

Un corpus sur le tabagisme a été collecté en partant de requêtes au moteur de recherche chinois Baidu (百度, `http://www.baidu.com/`), réputé pour mieux traiter les pages en langue chinoise que Google. Le terme de départ était 烟草, qui désigne le tabac. Ce terme a été affiné progressivement au vu des pages et des sites renvoyés, la sélection finale des pages étant faite manuellement. Le tableau 1 donne un extrait de la liste de mots clés employés. Cela a permis de repérer 20 sites particulièrement riches sur le tabagisme. Ces sites incluent des sites spécialisés dans le tabagisme ainsi que des sites généraux contenant des pages sur ce thème. Les pages HTML de ces 20 sites ont été téléchargées [3]. Un échantillon des fichiers résultants a été sélectionné aléatoirement et catégorisé manuellement en « spécialistes » ou « grand public ».

TAB. 1. *Mots clés chinois employés pour collecter des pages sur le tabagisme. *Le mot* 文化 *a le sens de « culture », « civilisation » (et pas celui d'« agriculture).*

mot clé	traduction
烟草文化	culture* du tabac
烟草医学	médecine et tabac
烟草科学	science du tabac
烟草健康	tabac et santé
烟草研究	recherche sur le tabac
烟草新闻	nouvelles sur le tabac
吸烟	fumer
戒烟	cesser de fumer

Nous avons pris pour cela les définitions suivantes de « spécialiste » et « grand public ». Nous considérons comme spécialistes les personnes qui connaissent bien le domaine du fait qu'elles y font de la recherche ou plus largement y exercent leur métier. Cela recouvre différentes disciplines (médecine, psychologie, pharmacie, sciences de la vie, écologie...) ou organismes (Ministère de la santé, organismes anti-tabac, administrations qui prennent en charge le contrôle, l'exportation et l'importation du tabac, fabricants de cigarettes, culture du tabac...). Ces personnes possèdent ainsi des

3. Le programme wget a été utilisé pour cela. Il a été appelé récursivement sur l'URL racine de chaque site, en le forçant à rester à l'intérieur du site. Même s'il reste difficile, du fait de l'existence de liens brisés ou de pannes de serveur ou de communication, de télécharger un site entier de façon réellement exhaustive, on obtient de cette manière une partie très importante des pages du site.

connaissances professionnelles sur le tabac. Par opposition aux spécialistes, le grand public comprend toutes les personnes qui ne possèdent pas de connaissances professionnelles sur le tabac.

Certaines pages sont cependant difficiles à classer. C'est le cas par exemple de pages de sites gouvernementaux donnant des informations administratives en relation avec le tabac (réglementations, lois), qui peuvent concerner aussi bien le grand public que des professionnels. Ou encore de chiffres de résultats du commerce du tabac, dont on n'est pas sûr qu'ils s'adressent uniquement aux professionnels. La distinction n'a pas toujours été simple à faire, et il se peut donc que notre choix ait été arbitraire dans quelques cas. Pour des raisons de temps, nous avons cependant conservé ces deux catégories. Idéalement, il faudrait créer des catégories intermédiaires ; ou encore, comme l'a suggéré l'un des relecteurs, avoir recours à plusieurs personnes pour effectuer la catégorisation (avec une part de spécialistes et une part de personnes du grand public), et considérer comme intermédiaires les pages sur lesquelles l'accord entre annotateurs est trop faible.

2.2. *Découpage en corpus d'entraînement et de test*

Cet échantillon a ensuite été découpé aléatoirement en corpus d'entraînement et corpus de test. Comme de coutume, le corpus d'entraînement servira de base d'exemples pour des systèmes de catégorisation automatique présentés plus bas ; le corpus de test servira à évaluer la performance de ces systèmes une fois entraînés. Le fait que les pages du corpus d'entraînement et de test proviennent des mêmes sites web rend le corpus de test homogène au corpus d'entraînement. Nous l'appellerons « corpus de test proche ». Pour tester sur un corpus plus éloigné la catégorisation qui sera apprise, nous avons collecté ensuite les pages de 16 sites supplémentaires. Nous appellerons le corpus correspondant « corpus de test éloigné ». Le tableau 2 donne les tailles respectives des corpus collectés. Chaque page des corpus d'entraînement et de test (proche et éloigné) a été catégorisée à la main en « spécialiste » ou « grand public ». La figure 1 montre une page « grand public » et une page « spécialiste ».

TAB. 2. *Constitution et taille du corpus collecté. Les tailles sont données en méga-octets et en nombre de caractères chinois (sinogrammes). La taille moyenne des textes est aussi indiquée.*

Corpus	URL	Mo	sinogrammes :	
			total	moyenne
corpus original (20 sites)	22646	115	64 297 265	2862
échantillon sélectionné	3178	20	11 232 248	3534
entraînement	2547	16	8 978 821	3525
test proche	631	4	2 295 505	3638
corpus de test éloigné	420	2	1 182 228	2815

(a) Page à destination du grand public

(b) Page à destination de spécialistes

FIG. 1. *Deux pages web chinoises sur le tabagisme.*

2.3. *Conversion en texte brut*

Les pages téléchargées étaient au format HTML. Le codage des caractères utilisé en Chine continentale est généralement GB2312. Lorsque c'était le cas, les documents ont été convertis en Unicode UTF-8 à l'aide du programme iconv. La conversion en texte brut a ensuite été réalisée en deux étapes : conversion en XML (plus précisément, XHTML, à l'aide du programme tidy du W3C) puis extraction des zones de texte par un programme XSLT. Enfin, chaque texte a été segmenté en phrases selon les points, points d'interrogation et d'exclamation chinois (donc en « pleine chasse » : 。？！).

2.4. *Faut-il segmenter en caractères ou en mots*

Le traitement du chinois pose le problème de la segmentation en mots [4]. En effet, les mots en chinois ne sont pas séparés par des espaces. Une phrase chinoise se compose d'une séquence de caractères, avec éventuellement des signes de ponctuation. Un mot se compose d'un ou plusieurs caractères, et aucune marque graphique n'indique où sont les frontières. Ce problème a motivé de nombreux travaux sur la segmentation en mots de textes chinois (Jin & Chen 2008), et des segmenteurs automatiques sont disponibles au téléchargement. Nous avons utilisé l'un de ces segmenteurs sur les textes de notre corpus. Le tableau 3 montre un exemple de passage d'une phrase et la segmentation en mots qui est produite par ce segmenteur.

TAB. 3. *Exemple de segmentation en mots (réalisée automatiquement).*

Original	中华人民共和国烟草专卖法
Segmentation en mots	中华人民共和国 / 烟草 / 专卖 / 法
Glose	*République Populaire de Chine / tabac / monopole / loi*
Traduction	*loi sur le monopole du tabac en RPC*

Une question qui se pose est de savoir si la segmentation en mots est utile pour notre objectif de catégorisation. De la même façon que des connaissances linguistiques n'apportent pas toujours une contribution positive en recherche d'information, il n'est pas certain qu'un texte segmenté en mots soit plus facile à catégoriser qu'un texte simplement segmenté en caractères. (Foo & Li 2004) ont observé un effet positif de la segmentation en mots en recherche d'information, mais citent de nombreuses autres expériences dont les résultats sont divergents. (He *et al.* 2000) effectuent une segmentation en mots avant la catégorisation de pages web chinoises, mais ne testent pas son apport à cette tâche.

4. Ce problème est commun à d'autres langues comme par exemple le japonais, le vietnamien, le thaï, etc. Par exemple, (Nguyen *et al.* 2006) prennent pour cadre la segmentation du chinois pour présenter leurs travaux sur celle du vietnamien.

Nous avons effectué pour notre part deux séries d'expériences, les unes avec segmentation en caractères, les autres avec segmentation en mots, afin de tester l'apport de la segmentation en mots.

3. Expériences de catégorisation de pages web en chinois

De nombreux algorithmes de catégorisation automatique (classification supervisée) existent (Sebastiani 2002), et de nombreuses implémentations de ces algorithmes sont disponibles au téléchargement. Plusieurs boîtes à outils d'apprentissage regroupent de telles implémentations, ce qui en fait des outils idéaux pour lancer des expériences systématiques. Nous avons fait le choix de la plate-forme Weka (Witten & Frank 2005) (`http://www.cs.waikato.ac.nz/ml/weka/`), dont l'interface graphique est pratique pour le test interactif et dont les fonctionnalités peuvent également être appelées en ligne de commande.

Une expérience de catégorisation suit le protocole habituel suivant :

1) Représentation de chaque texte du corpus d'entraînement et du test par un vecteur d'attributs.

2) Éventuellement, sélection d'un sous-ensemble d'attributs qui semblent suffire à véhiculer l'information nécessaire pour chaque texte.

3) Entraînement d'un catégoriseur d'un type donné sur la partie *entraînement* du corpus.

4) Application de ce catégoriseur entraîné sur la partie *test* du corpus.

Différents choix sont possibles à chaque étape ; le reste de cette section décrit ces choix.

3.1. *Attributs initiaux*

L'approche classique en recherche d'information ou en catégorisation de textes consiste à représenter un texte par un sac de mots. Nous suivons cette approche : chaque texte est décrit par les unités qui le composent. Ces unités sont ici soit les sinogrammes (caractères) soit les mots obtenus après segmentation. Techniquement, nous insérons des espaces entre les sinogrammes ou les mots. Cela nous permet d'utiliser ensuite la fonction de Weka qui convertit une chaîne de caractères en vecteur de mots (StringToWordVector) en gardant au maximum 3000 unités par texte. Chaque « mot » est accompagné d'une valeur indiquant s'il est présent ou pas dans le texte (nous n'avons pas cherché à prendre en compte le nombre d'occurrences de chaque unité dans les textes).

3.2. *Attributs supplémentaires*

Les unités élémentaires du texte sont-elles suffisantes à décrire le texte pour la tâche visée ? La lecture des textes fait apparaître certains indices qui semblent révélateurs d'un texte destiné aux spécialistes. Nous avons relevé les indices suivants :

– les marques d'énumérations ou d'alinéas : par exemple, « premièrement » peut être exprimé par "[1", "（一）", "1.", "1,", "1、", "(1)", "一，", "一：", "一、", "一.", "一要", "一是", "一方面", "首先" (le sinogramme "一" signifie « 1 ») ;

– des termes ou marques discursives typiques du discours scientifique : par exemple, "其次" (ensuite), "最后" (enfin), "总之" (en conclusion), "论文" (mémoire ou rapport), "(图"(« (figure ») , "(表" (« (tableau »), "《" (guillemet ouvrant) ;

– certains symboles mathématiques ou physiques ;

– la taille du texte en nombre de sinogrammes ;

– la taille moyenne des phrases en sinogrammes ;

– le fait que le texte contienne un nombre de chiffres supérieur à un seuil fixé ;

– le fait que le texte contienne un nombre de mots anglais supérieur à un seuil fixé ;

– etc.

Au total, 32 nouveaux attributs ont été définis et calculés sur chaque texte. Leur introduction ou pas dans les catégoriseurs est un paramètre de nos expériences. Cet ensemble d'attributs pourrait être étendu davantage, comme dans (Goeuriot *et al.* 2009), où sont recensés de façon plus systématique, pour un corpus comparable français et japonais, 13 critères structurels (en particulier obtenus à partir du marquage HTML), 18 critères modaux (actes allocutifs et élocutifs) et 11 critères lexicaux (vocabulaire spécialisé, caractères numériques, symboles, etc.).

Notons que ces attributs supplémentaires ont été définis manuellement. Des méthodes existent pour déterminer automatiquement des attributs supplémentaires, par exemple par recherche de termes associés (Qian *et al.* 2007), ce qui pourrait fournir des pistes intéressantes.

3.3. *Sélection d'attributs*

Certains attributs présents sont souvent en tout ou partie redondants ; de plus, un trop grand nombre d'attributs peut rendre plus difficile ou plus long l'entraînement. La sélection d'un sous-ensemble d'attributs cherche à réduire ces défauts potentiels. Différentes méthodes sont disponibles pour cela. Nous avons utilisé la fonction CfsSubsetEval de Weka, qui sélectionne des attributs fortement corrélés à la classe mais peu corrélés entre eux, alliée à BestFirst, qui effectue une recherche en meilleur d'abord avec possibilité de retour arrière. Ici aussi, cette possibilité de sélectionner un sous-ensemble d'attributs ou pas est un paramètre de notre jeu d'expériences.

3.4. *Catégoriseurs*

Parmi les divers algorithmes de catégorisation, nous avons choisi quatre catégoriseurs appartenant à des familles différentes :

– Arbre de décision : J48

– Bayésien naïf (Naïve Bayes) : NaiveBayes (binomial)

– K plus proches voisins (Kppv) : IBk

– Machine à points de support (SVM) : SMO

Nous avons utilisé ces catégoriseurs avec leurs paramètres par défaut.

3.5. *Expériences réalisées*

Au final, nous avons donc testé :

– quatre jeux d'attributs, comme indiqué dans le tableau 4 ;

– deux types d'unités, sinogrammes ou mots (section 2.4) ;

– trois corpus : ceux appelés *entraînement*, *test proche* et *test éloigné* dans le tableau 2 ;

– les quatre catégoriseurs ci-dessus (section 3.4) ;

ce qui fait un total de 96 expériences. Pour chacune, nous avons obtenu à travers Weka la précision P, le rappel R et la F-mesure de la catégorisation « grand public » (ou de la catégorisation « spécialiste »). Nous en rappelons ci-dessous les définitions (corrects = ensemble des documents « grand public » du corpus ; ramenés = documents que le catégoriseur a considérés comme « grand public ») :

$$P = \frac{\text{corrects} \cap \text{ramenés}}{\text{ramenés}} \; ; \; R = \frac{\text{corrects} \cap \text{ramenés}}{\text{corrects}} \; ; \; F = \frac{2PR}{P+R}$$

TAB. 4. *Les quatre jeux d'attributs employés. Six séries de telles expériences ont été menées, selon les « unités brutes » obtenues par la segmentation : soit des sinogrammes, soit des mots, selon la segmentation appliquée au préalable ; et selon le corpus de test : développement, test proche ou test éloigné. Soit au total 24 expériences par catégoriseur.*

	s/s attributs supplémentaires	avec attributs supplémentaires
s/s sélection	unités brutes	unités brutes + attributs ajoutés
sélection	sélection(unités brutes)	sélection(u brutes + attributs ajoutés)

4. Résultats

Un script lançant les expériences décrites ci-dessus a été exécuté. Nous présentons leurs résultats ci-dessous.

Le tableau 5 montre les résultats obtenus pour une catégorisation par arbre de décision, en utilisant tous les attributs initiaux, et en faisant varier le corpus de test et la segmentation. Comme on pouvait s'y attendre, la catégorisation est moins bonne sur les corpus de test que sur le corpus d'entraînement, et moins bonne sur le corpus de test éloigné que sur le corpus de test proche. La F-mesure sur le corpus de test éloigné est la plus pertinente pour évaluer l'application d'un tel catégoriseur sur de nouvelles pages.

TAB. 5. *Arbre de décision, attributs complets : catégorisation « grand public » ou « spécialiste ». P= précision, R= rappel, F = F-mesure, PICC = proportion des instances catégorisées correctement. S = segmentation en sinogrammes, M = segmentation en mots. 16 séries d'expériences similaires ont été réalisées pour les 4 jeux d'attributs et les 4 catégoriseurs. Le gras indique le meilleur résultat entre sinogrammes et mots.*

corpus	seg.	grand public			spécialiste			PICC
		P	*R*	*F*	*P*	*R*	*F*	
entraînement	S	0,982	**0,999**	0,990	**0,998**	0,979	0,989	0,989
	M	**0,987**	**0,999**	**0,993**	**0,998**	**0,986**	**0,992**	**0,992**
test proche	S	**0,971**	**0,985**	**0,978**	**0,983**	**0,966**	**0,974**	**0,976**
	M	0,968	**0,985**	0,976	**0,983**	0,963	0,973	0,975
test éloigné	S	0,926	**0,910**	**0,918**	**0,901**	0,920	**0,910**	**0,914**
	M	**0,930**	0,900	0,915	0,893	**0,925**	0,909	0,912

4.1. *Segmentation en sinogrammes ou en mots*

On observe de plus sur le tableau 5 que la segmentation en mots (M) amène des résultats sensiblements équivalents à ceux obtenus avec une simple segmentation en sinogrammes (S), et ce quel que soit le corpus de test : la différence observée est toujours largement inférieure à 1 point. Les quinze autres séries d'expériences confirment cette tendance.

4.2. *Ajout d'attributs supplémentaires, sélection d'attributs*

Le tableau 6 indique les attributs conservés après application de la sélection à l'ensemble d'attributs initial (sans ajout des attributs supplémentaires indiqués à la section 3.2). On y trouve aussi bien des termes plutôt associés à des documents à destination de spécialistes (分类 classification, 本文 cet article) que des termes associés à des documents destinés au grand public (下午 après-midi). Même si la segmentation en mots n'apporte pas d'avantage particulier pour la catégorisation, son application

permet de travailler sur des unités plus facilement interprétables humainement que des sinogrammes simples, qui sont souvent très ambigus hors contexte.

TAB. 6. *Attributs conservés après sélection, sans ajout d'attributs supplémentaires. La traduction en français a été ajoutée à la main pour en faciliter la lecture.*

terme	traduction	terme	traduction
2008年	année 2008	效果	effet
1998	(année ?)	数据	donnée
3	(mois ?)	是	être, oui, d'accord, correct
下午	après-midi	本文	cet article
分布	distribution	机械	mécanique
分类	classification	来源	source
北京	Pékin	栏目	rubrique
医院	hôpital	百科	encyclopédie
品种	espèce	综合	synthèse
品质	qualité, moralité	质量	qualité
差异	différence	近日	récemment
应用	application	量	mesurer, quantité, capacité
性	nature, sexe, genre, performance...	:	(ponctuation)
情况	situation		

Le tableau 7 montre l'influence de l'ajout d'attributs supplémentaires et de l'application d'une méthode de sélection d'attributs dans le cas de l'usage d'arbre de décision avec une segmentation en mots. Dans ce cas, la sélection d'attributs, bien que n'améliorant pas les résultats sur le corpus d'entraînement, donne de meilleurs résultats sur les corpus de test (chiffres en gras). L'ajout d'attributs supplémentaires avant une éventuelle sélection donne des résultats variables (chiffres en italiques) selon le corpus et l'application ou non de la sélection. Sur le corpus de test proche, sans sélection, l'ajout d'attributs supplémentaires améliore un peu les résultats ; mais si l'on applique la sélection d'attributs, les attributs supplémentaires font baisser la F-mesure. En revanche, sur le corpus de test éloigné, les attributs supplémentaires diminuent les performances avant sélection, mais les augmentent si l'on effectue une sélection. Nous revenons sur ces résultats ci-dessous dans la comparaison des catégoriseurs.

4.3. *Comparaison des catégoriseurs*

Les figures 2 et 3 synthétisent les F-mesures de la catégorisation « spécialiste » (Fs) des quatre catégoriseurs pour les différents jeux d'attributs, toujours avec une segmentation en mots, respectivement pour le corpus de test proche et pour le corpus de test éloigné. On y confirme que *la sélection d'attributs améliore généralement les résultats*, que ce soit sur les attributs originaux (« complets ») ou avec ajout d'attributs supplémentaires (« complets + ajoutés »), que ce soit sur le corpus de test proche

TAB. 7. *Arbre de décision, segmentation en mots. Influence de l'ajout d'attributs supplémentaires et de la sélection d'attributs. Fs = F-mesure de la catégorisation en « spécialiste » ; Fg = F-mesure de la catégorisation en « grand public ». Le gras indique le meilleur résultat entre sélection et non sélection, l'italique le meilleur résultat entre l'ajout ou pas d'attributs supplémentaires.*

Corpus	Cat.	originaux		originaux+supplémentaires	
			sélection		sélection
entraînement	Fs	**0,992**	0,987	**0,991**	0,984
	Fg	**0,993**	0,989	**0,992**	0,986
test proche	Fs	0,973	***0,985***	*0,976*	**0,981**
	Fg	0,976	***0,987***	*0,979*	**0,984**
test éloigné	Fs	*0,909*	**0,919**	0,905	***0,934***
	Fg	*0,915*	**0,924**	0,909	***0,941***

ou éloigné, *sauf* dans les deux cas suivants. D'une part, le catégoriseur à SVM fonctionne mieux sur des attributs sans sélection sur le corpus proche (ou sur le corpus éloigné si l'on n'ajoute pas les attributs supplémentaires) ; d'autre part, sur le corpus de test éloigné, le catégoriseur bayésien naïf est insensible à la sélection ou à l'ajout d'attributs supplémentaires.

Par ailleurs, *l'ajout de nos attributs supplémentaires n'améliore en général pas les résultats*, sauf dans de rares cas comme celui de l'arbre de décision avec sélection d'attributs sur le corpus de test éloigné, ou encore celui de SVM sans sélection d'attributs sur le corpus de test éloigné.

Globalement, sur le corpus de test éloigné (figure 3), k plus proches voisins ou arbre de décision combinés à la sélection d'attributs et pour les kPPV à l'ajout de nos attributs supplémentaires obtiennent les meilleures F-mesures (0,934). Sur le corpus de test proche (figure 2), SVM (sans sélection d'attributs, Fs=0,986) ou arbre de décision (plutôt sans nos attributs supplémentaires, avec sélection d'attributs, Fs=0,985) sont les plus performants. *Les arbres de décision semblent donc plus stables* sur notre type de tâche et nos corpus.

Nous avons réalisé des tests de significativité (test de Student unilatéral) sur les résultats obtenus avec les attributs « originaux + supplémentaires » avec sélection (groupes de droite « Sél(Orig+Sup) » sur les figures 2 et 3). Les différences entre corpus de test proche et éloigné pour un même catégoriseur sont toutes très significatives ($p < 0,001$). Pour le corpus de test proche, Naïve Bayes < {kPPV, AD} ($p = 0,02$) et SVM est moins distinct de {kPPV, AD} ($p = 0,07$), par contre Naïve Bayes n'a pas de différence significative avec SVM (et kPPV $\sim$AD). Pour le corpus de test éloigné, SVM < {Naïve Bayes, kPPV, AD} ($p < 0,05$) et Naïve Bayes n'a pas de différence significative avec {kPPV, AD} (et de nouveau kPPV$\sim$AD).

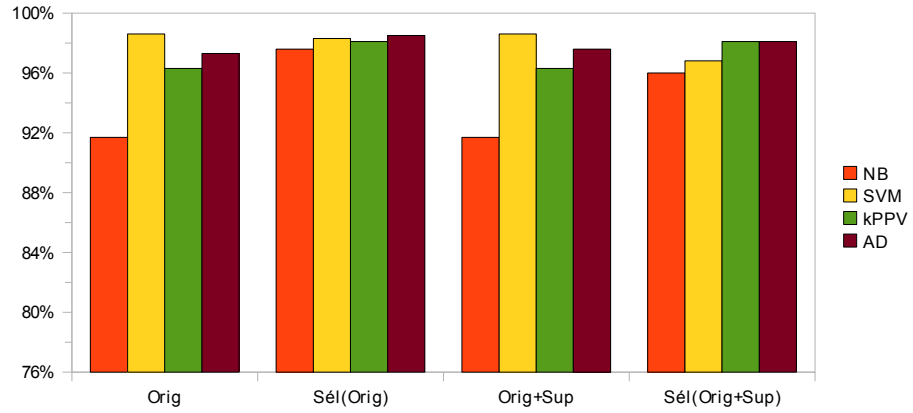


FIG. 2. *Comparaison des quatre catégoriseurs (NB = Naïve Bayes, SVM, kPPV = k plus proches voisins, AD = arbre de décision) pour les différents jeux d'attributs : corpus de test « **proche** », F-mesure de la catégorisation « spécialiste » (vs non spécialiste) (appelée Fs dans le tableau 7). Attributs complets = originaux ; ajoutés = supplémentaires.*

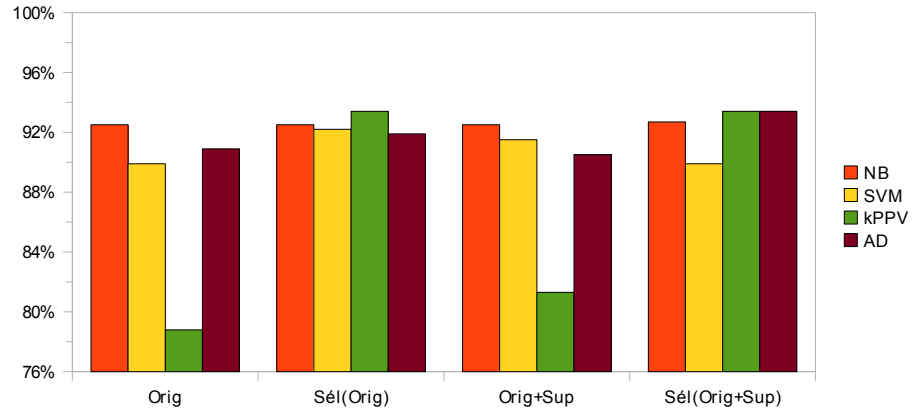


FIG. 3. *Comparaison des quatre catégoriseurs (NB = Naïve Bayes, SVM, kPPV = k plus proches voisins, AD = arbre de décision) pour les différents jeux d'attributs : corpus de test « **éloigné** », F-mesure de la catégorisation « spécialiste » (vs non spécialiste) (appelée Fs dans le tableau 7). Attributs complets = originaux ; ajoutés = supplémentaires.*

5. Discussion

Les différents catégoriseurs testés ont des résultats bons à excellents sur les deux corpus de test, avec des F-mesures en catégorisation « spécialiste » allant pour le corpus de test éloigné (avec attributs supplémentaires et sélection) de 0,899 (SVM) à 0,934 (kPPV, arbre de décision), et pour le corpus de test proche de 0,960 (Naïve Bayes) à 0,981 (kPPV, arbre de décision).

En considérant ces résultats positifs, on peut se demander si la tâche n'est pas trop facile, avec un corpus d'entraînement de plusieurs milliers de documents pour une catégorisation à deux classes. Il faut cependant s'attacher surtout aux résultats sur le corpus de test « éloigné », qui donne une meilleure idée des performances de catégorisation auxquelles on peut s'attendre sur de nouvelles pages web. Ceux-ci plafonnent à 0,934, ce que l'on pourrait vouloir encore améliorer.

D'après des sondages réalisés parmi les documents mal catégorisés, l'une des raisons qui peuvent expliquer les erreurs effectuées est que notre découpage en deux classes disjointes, « spécialiste » et « grand public », est une approximation trop abrupte de la réalité. Certaines pages web ont un caractère intermédiaire et s'adressent à un public sensibilisé au sujet sans être pour autant professionnel du domaine. (Sinclair & Ball 1996) distinguent par exemple les catégories suivantes de types de public : grand public, public sensibilisé, professionnels, spécialistes, étudiants [5]. De façon générale, un nombre de classes plus grand pourrait mieux refléter la réalité et aboutir à des classes plus homogènes, donc potentiellement mieux séparables. Il faut à l'inverse tenir compte du fait que la difficulté d'une catégorisation à N classes augmente généralement avec N ; et qu'il peut aussi exister un recouvrement entre les classes plus fines proposées.

Le corpus « éloigné » se compose de pages provenant de 16 sites web distincts de ceux du corpus initial. Ce nombre de sites est cependant encore trop faible pour que le test réalisé puisse être considéré comme général. Une nouvelle évaluation sur des pages web portant sur le tabagisme et tirées au hasard sur le web serait plus probante de ce point de vue.

6. Conclusion

Nous avons présenté ici des expériences de catégorisation selon le public visé de pages web chinoises sur le thème du tabagisme. Ces expériences obtiennent une catégorisation de bonne qualité sur le corpus de test préparé pour l'occasion (F-mesure jusqu'à 0,934 sur le corpus « éloigné »). Nous avons observé l'influence des

5. Notre traduction de : *members of the general public, informed lay people, professional people, specialists, students / trainees*.

différentes méthodes de catégorisation et du choix des attributs sur ces résultats, et noté qu'un corpus d'évaluation complémentaire plus divers serait nécessaire pour les confirmer.

Dans le cadre du projet C-Mantic, deux autres langues sont traitées : le français et l'anglais. Nous comptons tester les mêmes méthodes sur ces deux autres langues dès qu'un corpus suffisant sera disponible. Cela permettra de comparer les attributs et méthodes de catégorisation qui fonctionnent le mieux sur ces différentes langues, et d'étudier si certaines combinaisons sont stables à travers les langues. L'abord d'une autre opposition est également planifié : pages pro- et anti-tabac. Cette opposition, pour laquelle un corpus est en cours de constitution, sera sans doute à rapprocher de celle entre sites racistes et anti-racistes examinée lors du projet PRINCIP (Valette & Grabar 2004).

Remerciements

Nous remercions les relecteurs anonymes pour leurs commentaires constructifs. Les défauts restants sont notre fait.

7. Bibliographie

Foo S., Li H., « Chinese Word Segmentation and Its Effect on Information Retrieval », *Information Processing & Management*, vol. 40, p. 161-190, 2004.

Goeuriot L., Morin E., Daille B., « Reconnaissance du type de discours dans des corpus comparables spécialisés », *Actes CORIA 2009*, ARIA, 2009. Ce volume.

He J., Tan A.-H., Tan C.-L., « Machine Learning Methods for Chinese Web Page Categorization », *Actes ACL2000 International Workshop on Chinese Language Processing*, ACL, Hong Kong, 2000.

Jin G., Chen X., « The Fourth International Chinese Language Processing Bakeoff : Chinese Word Segmentation, Named Entity Recognition and Chinese POS Tagging », *Actes Sixth SIGHAN Workshop on Chinese Language Processing, IJCNLP 2008*, Asian Federation of Natural Language Processing, Hyderabad, India, 2008.

Karlgren J., Stylistic Experiments for Information Retrieval, PhD thesis, Stockholm University, 2000.

Lee D. Y., « Genres, Registers, Text Types, Domains, and Styles : Clarifying the Concepts and Navigating a Path Through the BNC Jungle », *Language, Learning, and Technology*, vol. 5, n° 3, p. 37-72, 2001.

Nguyen T. V., Tran H. K., Nguyen T. T., Nguyen H., « Word Segmentation for Vietnamese Text Categorization : An online corpus approach », *Actes IEEE RIFV 2006, Research, Innovation and Vision of the Future — The 4rd IEEE International Conference in Computer Science*, Hochiminh, Vietnam, 2006.

Qian T., Xiong H., Wang Y., Chen E., « On the strength of hyperclique patterns for text categorization », *Information Sciences*, vol. 177, n° 19, p. 4040-4058, 2007.

Rosso M. A., Using Genre to Improve Web Search, PhD thesis, University of North Carolina at Chapel Hill, 2005.

Santini M., « Automatic Genre Identification : Towards a Flexible Classification Scheme », *BCS IRSG Symposium : Future Directions in Information Access 2007 (FDIA 2007)*, Glasgow, Scotland, 2007. Held in conjunction with the European Summer School on IR (ESSIR 2007).

Sebastiani F., « Machine Learning in Automated Text Categorization », *ACM Computing Surveys*, vol. 34, n° 1, p. 1-47, 2002.

Sinclair J. M., Ball J., Preliminary recommendations on Text Typology, page WWW n° `http://www.ilc.cnr.it/EAGLES/pub/eagles/corpora/texttyp.ps.gz`, EAGLES (Expert Advisory Group on Language Engineering Standards), juin, 1996.

Valette M., Grabar N., « Caractérisation de textes à contenu idéologique : statistique textuelle ou extraction de syntagme ? l'exemple du projet PRINCIP », *in* G. Purnelle, C. Fairon, A. Dister (eds), *Le poids des mots, Actes 7èmes Journées internationales d'Analyse statistique des Données Textuelles (JADT)*, UCL — Presses Universitaires de Louvain, Louvain-la-Neuve, Belgique, p. 1106-1116, 2004.

Witten I. H., Frank E., *Data Mining : Practical machine learning tools and techniques*, 2nd edition edn, Morgan Kaufmann, San Francisco, Ca, 2005.

Impact de la reconnaissance de l'écriture en-ligne sur une tâche de catégorisation

Sebastián Peña Saldarriaga* — **Emmanuel Morin*** — **Christian Viard-Gaudin****

* *LINA - UMR CNRS 6241 - Université de Nantes*
2, rue de la Houssinière - BP 92208, 44322 NANTES Cedex 3
{sebastian.pena-saldarriaga, emmanuel.morin}@univ-nantes.fr

** *IRCCyN - UMR CNRS 6597 - École Polytechnique de l'Université de Nantes*
rue Christian Pauc - BP 50609, 44306 NANTES Cedex 3
christian.viard-gaudin@univ-nantes.fr

RÉSUMÉ. *Cet article s'intéresse à la problématique de la catégorisation automatique de documents manuscrits en-ligne et plus particulièrement à l'impact de la reconnaissance de l'écriture dans un processus de catégorisation utilisant des méthodes d'apprentissage automatique. Nous comparons les performances obtenues avec des documents issus d'un système de reconnaissance de l'écriture en-ligne et leur version originale électronique. Les résultats montrent qu'aucune perte significative des performances n'est à signaler lorsque 78 % des termes d'indexation sont correctement reconnus dans les documents à catégoriser. Nous montrons également que lorsque plus de la moitié de ces termes sont mal reconnus, l'utilisation d'une liste de candidats mots permet d'améliorer le taux de classification.*

ABSTRACT. *This paper deals with the automated categorization of on-line handwritten documents. We experimentally show the effects of word recognition errors on a categorization engine using machine learning algorithms. We compared the performances of a categorization system over the texts obtained through on-line handwriting recognition and the same texts available as ground truth. Results show that no significant accuracy loss is expected when about 78% percent of indexation terms are correctly recognized. Results also show that using the top n recognition-candidates increases categorization rates of texts where more than 50% of indexation terms are incorrectly recognized.*

MOTS-CLÉS : *Catégorisation de textes, documents en-ligne, reconnaissance de l'écriture*

KEYWORDS: *Text categorization, noisy text, on-line handwriting recognition*

1. Introduction

L'émergence de nouveaux dispositifs de saisie que sont les stylos numériques couplés à des supports papier, permet de produire de documents en-ligne de façon très efficace. De véritables documents peuvent être produits grâce à ces dispositifs, ils peuvent consister aussi bien en des prises de notes, des cours, des copies d'examens, des rédactions d'articles, etc. Cela élargit les champs d'application de la saisie d'écriture en-ligne cantonnés souvent à des terminaux de petites tailles (PDA, smartphone) où seule la reconnaissance des caractères se justifiait.

Les documents en-ligne constituent une nouvelle source d'information en langue naturelle pour laquelle peu d'applications de RI existent. La catégorisation permettrait d'apporter diverses fonctionnalités aux documents en-ligne telles l'organisation automatique, le routage ou la recherche par thème. De manière générale, la catégorisation peut servir de base à une recherche ou extraction d'information efficace.

Le travail que nous présentons ici, a pour but d'étendre la catégorisation à des données initiales qui ne sont pas des documents textuels. Dans le cas d'un document manuscrit en-ligne, il s'agit de la trajectoire échantillonnée de l'instrument d'écriture disponible sous la forme d'une séquence de points $(x(t), y(t))$ dans l'espace, ordonnés dans le temps. Il est donc possible de retracer un caractère trait par trait comme l'illustre la figure 1.

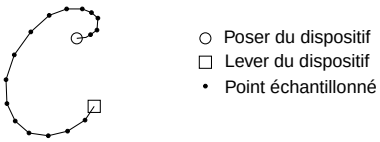


Figure 1. *Écriture en-ligne, appélée également encre numérique*

Une façon d'appréhender le problème de la catégorisation de ce type de documents est de se ramener à des données textuelles grâce à un système de reconnaissance de l'écriture. Dans ce travail nous explorons les effets produits par les erreurs de reconnaissance sur différents algorithmes de catégorisation que nous évaluons sur une base de documents reproduisant sous forme manuscrite les dépêches du corpus Reuters-21578 (Lewis, 1992) bien connu dans le domaine de la catégorisation de textes.

2. Travaux connexes

La collecte de grandes quantités de données manuscrites pour la catégorisation est une tâche difficile. À cause de cette difficulté, cette tâche n'a été explorée que très récemment (Vinciarelli, 2005, Koch, 2006, Peña Saldarriaga *et al.,* 2008, Milewski *et al.,* 2009). En revanche, des travaux sur la catégorisation de documents issus de la reconnaissance optique de caractères (OCR) existent depuis la dernière décennie (Ittner *et al.,* 1995, Junker *et al.,* 1998, Taghva *et al.,* 2000, Murata *et al.,* 2006).

Ces travaux utilisent différentes approches de catégorisation comme l'algorithme de Rocchio (Ittner *et al.,* 1995), des classifieurs bayésiens naïfs (Taghva *et al.,* 2000) ou des méthodes à base de n-grammes (Junker *et al.,* 1998). Cependant, la plupart s'appuient sur des données très spécifiques (Koch, 2006, Taghva *et al.,* 2000, Milewski *et al.,* 2009) ou des bases non standardisées (Ittner *et al.,* 1995, Junker *et al.,* 1998).

Les travaux les plus récents (Vinciarelli, 2005, Murata *et al.,* 2006, Peña Saldarriaga *et al.,* 2008) utilisent comme support un corpus bien connu dans le domaine de la catégorisation : le corpus Reuters-21578 (Lewis, 1992). Dans l'un des nos travaux précédents (Peña Saldarriaga *et al.,* 2008), nous avons évalué l'impact de la reconnaissance de l'écriture lorsque la catégorisation est effectué avec des modèles entraînés sur des documents électroniques. Ces expériences sont proches de celles décrites par (Vinciarelli, 2005).

La contribution présentée dans cet article se différencie des travaux antérieurs sur deux aspects. D'une part, ceci est à notre connaissance la première fois que la catégorisation est appliquée à des documents issus du domaine en-ligne en utilisant des documents manuscrits aussi bien pour l'entraînement que pour l'évaluation. Il faut également noter que les travaux en relation avec la RI dans le domaine se contentent souvent de rechercher des séquences de caractères dans l'encre numérique (Lopresti *et al.,* 1994, Jain *et al.,* 2003). D'autre part, nous utilisons la plus grande base de documents en-ligne existant à notre connaissance, elle est basée sur un corpus parfaitement standardisé. De plus nous donnons une analyse minutieuse des performances du système selon différentes perspectives applicatives.

3. Catégorisation automatique de textes

La catégorisation automatique de textes (CAT) est l'affectation d'une ou plusieurs étiquettes à un document en fonction de son contenu textuel. Un algorithme de catégorisation est un modèle mathématique dont l'objectif est de détecter le ou les thèmes abordés dans un texte. Plus formellement, la CAT peut être définie par une un fonction f telle que :

$$f : (d_i, c_j) \rightarrow \{vrai, faux\}, \qquad \forall (d_i, c_j) \in D \times C \qquad [1]$$

Avec d_i un document appartenant au domaine D et c_j une catégorie de l'ensemble $C = \{c_1, c_2, ..., c_{|C|}\}$.

L'apprentissage automatique permet d'approcher f par induction à partir d'un jeu d'entraînement (Sebastiani, 2002), c'est-à-dire, en utilisant un ensemble de textes dont la catégorie est connue au préalable.

Nous avons retenu deux méthodes de catégorisation parmi les nombreux algorithmes existants : la méthode des k-Plus Proches Voisins (kPPV) et les Séparateurs à Vaste Marge (SVM). Ces deux méthodes ont été choisies parce qu'elles figurent

parmi les approches les plus performantes développées durant la décennie (Yang *et al.,* 1999, Joachims, 2002, Debole *et al.,* 2005). De plus, elles nous permettent d'évaluer les effets de la reconnaissance sur deux méthodes de nature très différente.

Ces deux méthodes sont basées sur une représentation vectorielle des données (Salton *et al.,* 1975). Cela veut dire que la donnée de base, des textes bruts en langue naturelle, doit subir un certain nombre de transformations afin de se conformer au formalisme vectoriel. Dans ce formalisme, chaque dimension de l'espace vectoriel correspond à une entité représentative du sens, appelée communément terme, préalablement extraite du jeu d'apprentissage (*cf.* figure 2). La sélection des termes de l'espace vectoriel a été effetuée en utilisant l'algorithme de (Forman, 2004) sur les scores donnés par le test du χ^2 (Yang *et al.,* 1997) pour chacun des termes et chacune des catégories.

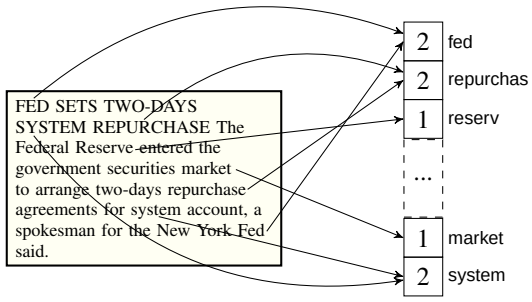


Figure 2. *Représentation vectorielle d'un texte*

L'ensemble des pré-traitements effectués pour transformer un texte brut en une liste de *termes*, est donné par la figure 3. Chacune de ces étapes est décrite ci-dessous.

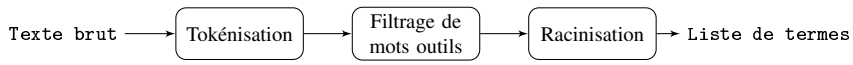


Figure 3. *Pré-traitements linguistiques*

La **tokénisation** permet de segmenter un texte en occurrences de formes (*tokens*). La liste de tokens obtenue après l'étape de tokénisation est filtrée en utilisant une liste de **mots outils** prédéfinie, celle-ci est constituée en particulier de prépositions, conjonctions, déterminants et auxiliaires. Enfin, la **racinisation** est utilisée afin de réduire les variations morphologiques d'un mot. Elle consiste à supprimer tous les affixes d'un mot, même si l'algorithme que nous utilisons n'effectue que la désuffixation des mots (Porter, 1980).

Vient naturellement ensuite une étape de pondération qui permet de déterminer de manière quantitative la représentativité de chacun des termes de la liste obtenue après l'étape de pré-traitements. La mesure $tf \times idf$ normalisée (Spärck Jones, 1979)

permet d'évaluer l'importance d'un terme en prenant en compte sa fréquence locale, c'est-à-dire relative à un document (*term frequency*) et sa fréquence globale, relative à un corpus (*inverse document frequency*). Le poids d'un terme i dans un vecteur, représentant un document, est donné par la formule suivante :

$$w_i = \frac{f_i \times \log \frac{N}{n_i}}{\sqrt{\sum_{j=1}^{M} \left(f_j \times \log \frac{N}{n_j} \right)^2}}, i = 1, \ldots, M \quad [2]$$

Avec f_i la fréquence du terme i dans un document, N le nombre de documents dans le corpus, M le nombre total de termes et n_i la fréquence du terme i dans le corpus.

3.1. *Les k-Plus Proches Voisins*

L'algorithme des kPPV est une méthode très connue dans le domaine de la catégorisation automatique. La catégorisation d'un document d, s'opère en comparant le vecteur du document à l'ensemble des vecteurs du jeu d'entraînement. Les k plus proches vecteurs sont sélectionnés et une probabilité d'appartenance à une catégorie c, pondérée par le cosinus des k documents les plus semblables, est calculée.

$$p(c|d) = \frac{\sum_{i=1}^{k} \cos(d, x_i) \times I(x_i, c)}{\sum_{i=1}^{k} \cos(d, x_i)} \quad [3]$$

avec

$$I(x_i, c) = \begin{cases} 1 & si\ categorie(x_i) = c \\ 0 & sinon \end{cases} \quad [4]$$

3.2. *Séparateurs à Vaste Marge*

Les Séparateurs à Vaste Marge (en Anglais *Support Vector Machines*) ont été introduits par Vapnik (1995). Ils sont devenus une méthode phare pour la classification supervisée et les travaux de Joachims, (2002) montrent qu'ils sont de par leur nature adaptés pour la CAT.

Les vecteurs des documents d'entraînement sont projetés dans un espace à grande dimension où il est possible de définir par apprentissage une surface de séparation entre des exemples positifs et négatifs appelée hyperplan. L'hyperplan optimal est

choisi de façon à minimiser les erreurs de catégorisation et à maximiser la marge de séparation entre les exemples.

La catégorisation utilisant l'approche par SVM a été réalisée grâce au package SVM^{light} V6.0 developpée par Joachims (2002)[1].

3.3. *Évaluation des méthodes de catégorisation*

Puisque nous disposons de plusieurs méthodes de catégorisation, nous devons mesurer la qualité des réponses données par le catégoriseur. Pour cela, nous disposons de deux mesures classiques : la précision (P) et le rappel (R).

Pour une catégorie c, la précision évalue la qualité du classifieur à ne pas introduire de documents d'une autre catégorie dans c. Il s'agit du nombre de documents bien classés sur le nombre de documents classés dans c.

$$P(c) = \frac{Documents\ bien\ classés\ dans\ c}{Documents\ classés\ dans\ c} \quad [5]$$

Le rappel, quant à lui, évalue le degré de complétude, c'est-à-dire le nombre de documents bien classés sur le nombre total de documents de la classe c.

$$R(c) = \frac{Documents\ bien\ classés\ dans\ c}{Documents\ de\ c} \quad [6]$$

Ces deux mesures prises l'une sans l'autre ne permettent d'évaluer qu'une facette du système de catégorisation : la qualité ou la quantité. Les courbes de précision *vs* rappel (Baeza-Yates *et al.,* 1999, chap. 3) permettent de mieux comprendre le comportement du classifieur, et de visualiser l'évolution de la précision en fonction du rappel pour les 11 niveaux standard $[0, 0, 1, \ldots, 1, 0]$. Comme le système est évalué sur un ensemble de catégories, nous utilisons les deux méthodes classiques pour moyenner la précision et le rappel : micro-moyenne et macro-moyenne (Sebastiani, 2002).

Les courbes de précision *vs* rappel sont utiles pour évaluer un système, lorsqu'il s'agit de retrouver un ensemble de documents étant donnée une catégorie (catégorisation centrée-catégorie). *A contrario*, lorsque le but est de retrouver un ensemble de catégories étant donné un document (catégorisation centrée-document), la mesure la plus appropriée est le taux de classification, il s'agit du nombre de documents bien classés sur le nombre de documents du corpus. Lorsque chaque document appartient à une catégorie et à une seule, le taux de classification correspond à la micro-moyenne de la précision ou le rappel (Beney, 2008).

1. L'application est librement téléchargeable à l'adresse suivante : `http ://svmlight.joachims.org`

4. Reconnaissance de l'écriture en-ligne

L'objectif de la reconnaissance de l'écriture en-ligne est de déterminer la suite de caractères la plus vraisemblable étant donné un signal correspondant au tracé manuscrit et les informations fournies par un ensemble de connaissances *a priori* sur la langue (*cf.* figure 4). L'objet de ce travail n'étant pas directement la reconnaissance de l'écriture, nous nous contenterons de décrire ci-dessous le moteur de reconnaissance en tant qu'outil plutôt qu'en tant que système.

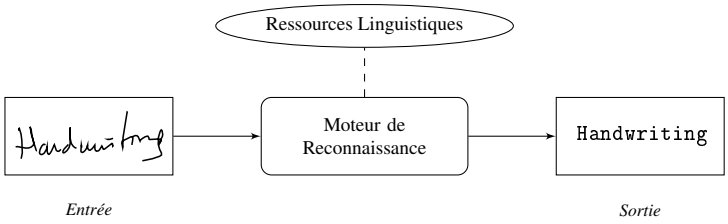


Figure 4. *Reconnaissance de l'écriture*

4.1. *Moteur de reconnaissance*

Le moteur de reconnaissance utilisé dans nos expériences est celui de MyScript Builder[2]. Il s'agit d'un ensemble de bibliothèques tournées vers la reconnaissance de l'écriture en-ligne.

MyScript Builder SDK est un outil stable, paramétrable et documenté. Il permet d'associer différentes ressources linguistiques afin de guider et d'optimiser la reconnaissance. Il est possible de définir des ressources spécifiques, soit sous forme de lexiques ou encore d'automates lexicaux, ou bien d'utiliser les deux ressources standard livrées avec le SDK :

– **lk-text** est une ressource contrainte par un modèle statistique du langage au niveau mot et un lexique standard. Le premier permet de favoriser la reconnaissance des séquences de mots les plus probables. Ainsi, *'je tue'* sera prioritaire par rapport à *'je tu'*. Cette ressource permet également de reconnaître des éléments hors-lexique comme les dates, les nombres, les codes postaux, etc.

– **lk-free** apporte peu de contraintes sur ce que l'on veut reconnaître. Il n'y pas de lexique mais seulement un modèle de langage au niveau caractère. Cette ressource permet principalement de reconnaître la séquence de caractères la plus vraisemblable. Ainsi, '$MATIN$' sera prioritaire par rapport à '$MAT1N$'.

2. `http ://www.visionobjects.com/products/software-development-kits/myscript-builder/`

Il faut noter que les deux ressources décrites ci-dessus sont des ressources livrées avec un système commercial, il nous est impossible de dire combien de mots sont présents dans le lexique, ni sur combien de documents ou mots ont été entraînés les modèles de langage. Il s'agit d'un système de reconnaissance totalement générique qui n'intègre aucune connaissance *a priori* sur le corpus de catégorisation.

4.2. *Évaluation de la reconnaissance*

Les mesures d'évaluation que nous décrivons ci-dessous vont nous permettre de vérifier le comportement du reconnaisseur lorsque telle ou telle ressource est utilisée.

Le *bruit* induit par la reconnaissance (insertion, suppression et remplacement de mots) est souvent mesuré au niveau des mots. Le taux d'erreur au niveau mot (Word Error Rate, WER) correspond au pourcentage de mots mal reconnus sur la totalité de mots à reconnaître pour une séquence donnée :

$$WER = 1 - \frac{\sum_{i=1}^{N} \min\left(wf(i), wf'(i)\right)}{\sum_{k=1}^{N} wf(k)} \quad [7]$$

Avec $wf(i)$ and $wf'(i)$ les fréquences du mot i dans le texte d'origine et le texte reconnu respectivement, et N le nombre de mots à reconnaître.

Une autre façon de mesurer le bruit, est de travailler au niveau terme. Le taux d'erreur au niveau terme (Term Error Rate, TER) est plus adapté à la catégorisation car il tient compte des pré-traitements qui normalisent les textes. Puisque '*rêvas*' et '*rêves*' ont la même racine, reconnaître l'un à la place de l'autre ne modifie pas la liste de termes reconnus. Reconnaître '*pour*' à la place de '*par*' ne la modifie pas non plus, car quelque soit le mot reconnu, il sera filtré puisque c'est un mot outil.

Le TER est calculé grâce à la formule suivante (Vinciarelli, 2005) :

$$TER = 1 - \frac{\sum_{i=1}^{N} \min\left(tf(i), tf'(i)\right)}{\sum_{k=1}^{N} tf(k)} \quad [8]$$

Avec $tf(i)$ and $tf'(i)$ les fréquences du terme i dans le texte d'origine et le texte reconnu respectivement, et N le nombre de termes de référence.

5. Corpus manuscrit

Le corpus utilisé dans nos expériences est un sous-ensemble du corpus Reuters-21578 (Lewis, 1992) reproduit sous forme manuscrite. Le corpus Reuters-21578 est l'un des plus utilisés dans la littérature scientifique pour l'évaluation de méthodes de catégorisation (Debole *et al.,* 2005). Ses documents sont distribuées en 135 catégories,

dont seulement 90 sont représentées dans l'ensemble d'entraînement et de test. Les 10 catégories ayant le plus d'effectifs comptent pour 90 % des documents du corpus.

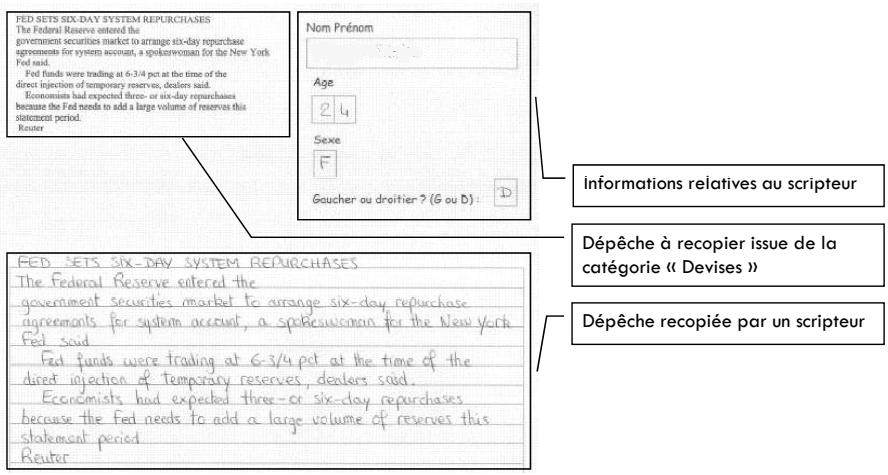


Figure 5. *Formulaire pour la collecte de données*

La collecte a été effectuée à l'aide de formulaires (*cf.* figure 5) et de stylos numériques. Nous avons mobilisé plus de 1 500 scripteurs pendant une période de 4 mois.

Pour cette collecte, nous avons dû choisir un nombre réduit de documents. Comme 90 % des documents appartiennent aux 10 premières catégories, nous avons choisi au hasard 2 000 documents d'entraînement et 500 documents de test parmi les documents de celles-ci. Les documents choisis n'appartiennent qu'à une seule catégorie et comportent au maximum 120 mots pour faciliter la tâche de saisie.

Une fois la collecte terminée, nous avons dû trier, anonymiser et associer le texte original à chacun des formulaires. À la fin du tri des documents, seuls 2 029 documents étaient exploitables. La distribution de ces documents par catégorie selon les ensembles d'entraînement et de test et donnée par le tableau 1.

Les documents du corpus abordent divers sujets comme les fusions-acquistions d'entreprises, les marchés de matières premières agricoles (céréales, sucre, etc.), le cours du pétrole et ses dérivés, les marchés de changes et du taux d'intérêt, etc.

6. Expériences

Afin de valider les méthodes de catégorisation sur le corpus que nous venons de présenter, nous avons mené plusieurs expériences. En premier lieu, nous avons effectué la reconnaissance des documents manuscrits (*cf.* section 6.1). Ensuite, nous avons catégorisé les documents tels qu'ils étaient donnés par le moteur de reconnaissance (*cf.*

Catégorie	Entraînement	Test
Dividendes (*earn*)	642	108
Fusion-Acquisition (*acq*)	349	72
Céréales (*grain*)	125	51
Devises (*money-fx*)	177	49
Pétrole (*crude*)	81	40
Taux d'intérêt (*interest*)	76	30
Import/Export (*trade*)	59	21
Transport Maritime (*ship*)	54	13
Sucre (*sugar*)	32	10
Café (*coffee*)	30	10
Total	1 625	404

Tableau 1. *Documents par catégorie*

section 6.2) ou alors en exploitant des informations supplémentaires en provenance du reconnaisseur, à savoir la liste de candidats mots à la reconnaissance (*cf.* section 6.3).

6.1. *Reconnaissance des documents*

Le moteur de reconnaissance de MyScript Builder a été utilisé pour effectuer la reconnaissance des documents en utilisant les deux ressources disponibles *lk-free* et *lk-text*. Le tableau 2 montre les performances de la reconnaissance en fonction de la ressource et du jeu de documents.

Ressource	WER	TER
lk-free	52,47 %	55,75 %
lk-text	22,30 %	23,01 %

(a) Ensemble d'entraînement

Resource	WER	TER
lk-free	52,48 %	55,85 %
lk-text	22,08 %	21,90 %

(b) Ensemble de test

Tableau 2. *Taux d'erreur à la reconnaissance*

Les textes reconnus avec la ressource *lk-free* sont fortement degradés, en effet plus d'un mot sur deux est perdu en moyenne. Le nombre de termes perdu est encore plus important. Les textes issus de la reconnaissance avec *lk-text* sont beaucoup moins bruités : 77 % des mots et autant de termes présents dans les textes sont correctement reconnus. Cette différence entre les deux ressources s'explique par la stratégie de reconnaissance de *lk-text* qui consiste à cherche la séquence de mots la plus vraisemblable.

Si nous comparons les performances sur le jeu d'entraînement et de test, nous remarquons également que MyScript Builder se comporte de manière cohérente sur les deux ensembles. De plus les taux de reconnaissance de *lk-text* sont bons si on consi-

dère que la couverture lexicale de la ressource est de 70,64 % et le WER théorique de 17,88 % (TER ≈19,45 %) par rapport au corpus électronique.

6.2. *Catégorisation des documents*

Notre première expérience a consisté à confronter notre système à d'autres systèmes de l'état de l'art. Pour cela nous avons utilisé le sous-ensemble $R(90)$ (Debole *et al.,* 2005) du corpus Reuters-21578. Cela permet de valider notre système de catégorisation en comparant nos résultats avec ceux disponibles pour le même jeu de données avec les mêmes méthodes de catégorisation et des paramètres expérimentaux aussi proches que possible (Yang *et al.,* 1999, Joachims, 2002).

Afin de nous conformer avec les résultats de ces travaux, nous présentons la micro-moyenne de la F_1-mesure (Manning *et al.,* 1999, p. 269) dans le tableau 3.

	(Yang *et al.,* 1999)	(Joachims, 2002)	*Cet article*
$\text{kNN}_{k=30}$	0,857	0,826	0,840
SVM	0,859	0,875	0,889

Tableau 3. *Résultats pour le corpus original - Micro-moyenne de la* F_1

Les performances rapportés ci-dessus montrent que nos résultats sont comparables aux résultats de l'état de l'art pour les mêmes algorithmes de classification. La prochaine étape est d'appliquer notre système de catégorisation aux données issues du système de reconnaissance.

Eu égard à la taille réduite du corpus ayant servi pour la collecte de données manuscrites, les différents paramètres de classifieurs ont été ajustés. Ceci a été fait par validation sur un sous-ensemble du jeu d'entraînement électronique de 1 625 documents. Les SVM sont utilisés avec 1 000 termes et les kPPV avec 15 plus proches voisins et 300 termes. Ces paramètres sont conservés pour la catégorisation des documents manuscrits. Il faut noter que ces paramètres peuvent ne pas être optimaux pour les données manuscrites, cependant notre but était de tester notre système sur la même tâche de catégorisation et non pas d'optimiser minutieusement les paramètres.

Les résultats de la catégorisation selon le type de documents et de classifieur sont indiqués dans le tableau 4. Nous avons utilisé les documents électroniques pour calculer des performances de référence. Lorsque les documents manuscrits sont utilisés, ils le sont aussi bien pour la sélection de termes et l'apprentissage que pour le test.

Ces résultats ont été obtenus en cherchant pour un document donné une liste de catégories potentielles et en assignant la catégorie ayant le score le plus important.

Le premier constat que nous faisons est une baisse d'environ 10 % du taux de classification avec *lk-free* par rapport à la référence, et ce quel que soit le type de classifieur. Cette ressource ne semble pas adaptée à une reconnaissance orientée catégorisation. Cependant, les performances ne sont pas aussi mauvaises, si nous considé-

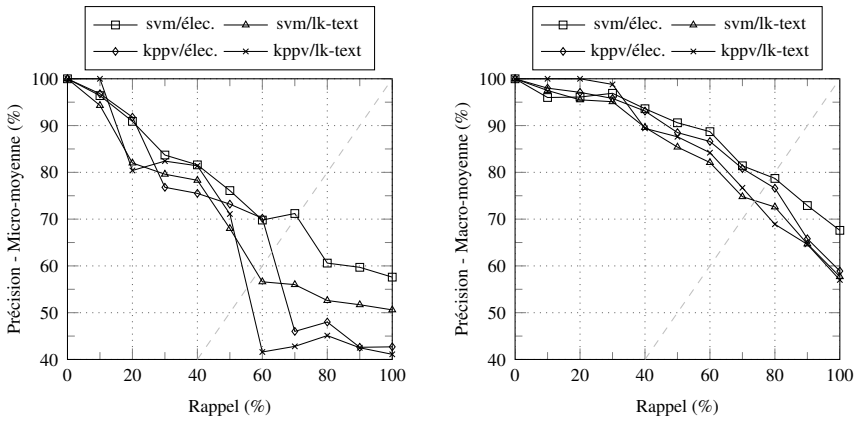
Documents	kPPV	SVM
électroniques	90,10 %	93,56 %
lk-text	89,36 %	91,83 %
lk-free	79,46 %	84,65 %

Tableau 4. *Taux de classification pour les différents documents*

rons que plus de la moitié de l'information textuelle contenue dans les documents est perdue à cause de la reconnaissance.

De son côté, la ressource *lk-text* obtient des résultats convenables avec les deux méthodes. Une baisse de 1 % est enregistrée avec kPPV tandis qu'avec les SVM cette baisse est d'environ 2 %.

Une vue détaillée des performances du système, lorsque les documents de *lk-text* sont utilisés, est donnée par les courbes de précision vs rappel (*cf.* figure 6). Ces courbes sont le résultat d'une catégorisation centrée-catégorie. L'intérêt de calculer les courbes sur le rang des documents est double. D'une part cela évite d'avoir à définir un seuil (Yang, 2001) dépendant foncièrement de l'application finale (avoir plus ou moins de précision ou de rappel). D'autre part ces courbes constituent une borne supérieure des performances d'un système utilisant un seuil[3].



(a) Micro-moyenne (b) Macro-moyenne

Figure 6. *Précision vs Rappel*

La figure 6(a) montre que jusqu'à 50 % de rappel, les courbes restent proches les unes par rapport aux autres. Au-delà, une nette différence apparaît. Les courbes des kPPV chutent de façon importante, mais restent proches pour les taux de rappel supé-

3. Cela découle du procédé d'interpolation (voir (Baeza-Yates *et al.,* 1999, p. 78))

rieurs à 70 %. La courbe des SVM avec les documents électroniques dépasse de façon visible toutes les autres à partir de 60 % de rappel.

En macro-moyenne (*cf.* figure 6(b)), aucune différence significative entre les courbes n'est observée avant 80 % de rappel. Au-delà de ce taux, seuls les SVM avec les documents électroniques surpassent les autres séries de manière visible.

Bien que les SVM donnent des meilleurs résultats que les kPPV dans tous les cas de figure, la baisse des performances qu'ils enregistrent est plus importante. Ceci est dû au choix des paramétres optimaux. En effet, les approches basées sur kPPV doivent leurs performances à l'identification d'un espace vectoriel discriminant dans le jeu d'entraînement (Dasarathy, 1991). Puisque les 300 termes utilisés avec kPPV sont les 300 plus discriminants d'un point de vue statistique, ils ont plus de chances d'être correctement reconnus dans l'ensemble de test.

6.3. *Utilisation des n-best*

En plus de donner le mot le plus probable, le moteur de reconnaissance peut donner une liste ordonnée des meilleurs candidats mots à la reconnaissance, où à chaque candidat est associé une probabilité. Ainsi, il est possible de garder l'information correspondant à un terme de l'espace vectoriel qui ne serait pas arrivé en première position.

Cependant cela introduit du bruit supplémentaire dans les textes, plus la liste est grande plus le bruit est important. Afin de réduire l'impact de ce bruit, nous utilisons la probabilité d'un terme pour la pondération plutôt que sa fréquence dans le document.

La figure 7 montre l'évolution du taux de classification en fonction de la taille de la liste de n-best. Dans cette expérience nous avons utilisé les documents de *lk-free* et *lk-text*, et les catégoriseurs tels qu'ils ont été paramétrés précédemment.

Lorsque nous utilisons la liste des n-best avec *lk-text*, le taux de classification baisse avec l'augmentation de n. En revanche, lorsque les documents de *lk-free* sont utilisés, pour tout $n > 1$ le taux de classification est supérieur à celui obtenu en ne prenant que le premier candidat, et ce quel que soit le classifieur utilisé. L'augmentation moyenne est de 2,1 % avec un écart-type de 1,2. Cependant cette augmentation n'est pas régulière et ne semble pas être correlée avec n.

7. Conclusion et perspectives

L'augmentation de la production de documents manuscrits en-ligne, intégrant de l'écriture dynamique, nécessite le développement de nouveaux outils de gestion adaptés à la nature même des documents. La catégorisation peut permettre d'organiser les documents de façon à pouvoir effectuer, ultérieurement, une extraction ou une recherche d'information efficace.

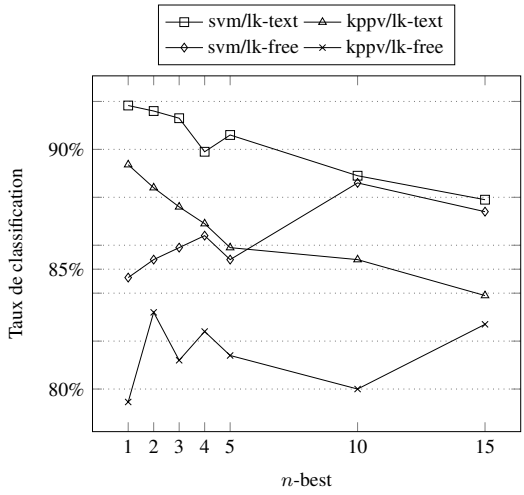


Figure 7. *Taux de classification en fonction de* n

Mais l'information textuelle contenue dans ces documents n'est accessible que grâce à un processus de reconnaissance. Ce processus induit des erreurs dans le texte résultant. Ce travail montre expérimentalement l'influence que peuvent avoir ces erreurs lorsque nous voulons catégoriser des textes en aval de la reconnaissance.

Deux algorithmes d'apprentissage automatique ont été évalués et comparés dans cette étude. L'évaluation des performances a été effectuée sur un sous-ensemble du corpus Reuters-21578 d'environ 2 000 documents manuscrits. La qualité du classifieur a été mesurée sur deux versions reconnues du corpus manuscrit et sa version électronique.

Lorsque nous comparons les performances obtenues avec les documents électroniques et les documents manuscrits, nous constatons qu'il n'y a pas de baisse significative des performances lorsque les documents reconnus avec *lk-text* sont utilisés : entre 1 % et 2 % pour un TER d'environ 56 % . Bien que peu significative, nous avons tenter de pallier cette baisse en utilisant la liste des n-best mots donnée par le moteur de reconnaissance, mais aucun effet positif sur le taux de catégorisation n'a été observé.

En revanche, l'utilisation de la liste des n-best mots s'est révélée bénéfique pour les documents issus de la reconnaissance avec *lk-free*. En effet, aussi bien pour les kPPV que pour les SVM, une augmentation du taux de classification allant de 1 % à 4 % a été observée. Bien qu'étant inférieur aux résultats de *lk-text*, les résultats obtenus avec *lk-free* en utilisant les n-best suggèrent que des niveaux convenables de catégorisation peuvent être atteints même dans des conditions extrêmes de dégradation des documents. Mais cette amélioration du taux de classification n'est pas regulière ou

correlée avec la taille de la liste des n-best. Cela montre que les interactions entre le niveau et le type de bruit présent dans un document reconnu, et le processus de catégorisation n'ont pas encore été comprises et explorées en détail. Puisque nous disposons aujourd'hui d'une base considérable de documents manuscrits, d'autres expériences doivent être effectuées dans ce sens.

Remerciements

Ces travaux ont ete soutenus par la Région Pays de la Loire à travers le projet MILES et par l'Agence Nationale de la Recherche à travers le programme Technologies Logicielles (ANR-06-TLOG-009).

8. Bibliographie

Baeza-Yates R., Ribeiro-Neto B., *Modern Information Retrieval*, Addison-Wesley, 1999.

Beney J., *Classification Supervisée de Documents*, Hermès Science / Lavoisier, 2008.

Dasarathy B. V., *Nearest Neighbor (NN) Norms - NN Pattern Classification Techniques*, IEEE Computer Society Press, 1991.

Debole F., Sebastiani F., « An Analysis of the Relative Hardness of Reuters-21578 Subsets », *Journal of the American Society for Information Science and Technology*, vol. 56, n^o 6, p. 584-596, 2005.

Forman G., « A Pitfall and Solution in Multi-class Feature Selection for Text Classification », *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, p. 38-46, 2004.

Ittner D. J., Lewis D. D., Ahn D. D., « Text Categorization of Low Quality Images », *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR '95)*, p. 301-315, 1995.

Jain A. K., Namboodiri A. M., « Indexing and Retrieval of On-line Handwritten Documents », *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR '03)*, vol. 2, p. 655-659, 2003.

Joachims T., *Learning to Classify Text using Support Vector Machines*, Kluwer Academic Publishers, 2002.

Junker M., Hoch R., « An Experimental Evaluation of OCR Text Representations for Learning Document Classifiers », *International Journal on Document Analysis and Recognition*, vol. 1, n^o 2, p. 116-122, 1998.

Koch G., Catégorisation Automatique de Documents Manuscrits : Application aux Courriers Entrants, PhD thesis, Université de Rouen, 2006.

Lewis D. D., « An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task », *Proceedings of the 15th Annual International ACM SIGIR Conference (SIGIR '92)*, p. 37-50, 1992.

Lopresti D., Tomkins A., « On the Searchability of Electronic Ink », *Proceedings of the 4th International Workshop on Frontiers in Handwriting Recognition (IWFHR '94)*, p. 156-165, 1994.

Manning C., Schütze H., *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

Milewski R. J., Govindaraju V., Bhardwaj A., « Automatic Recognition of Handwritten Medical Forms for Search Engines », *International Journal on Document Analysis and Recognition*, vol. 11, n^o 4, p. 203-218, 2009.

Murata M., Busagala L. S. P., Ohyama W., Wakabayashi T., Kimura F., « The Impact of OCR Accuracy and Feature Transformation on Automatic Text Classification », *Proceedings of the 7th IAPR Workshop on Document Analysis Systems (DAS '06)*, p. 506-517, 2006.

Peña Saldarriaga S., Morin E., Viard-Gaudin C., « Categorization of On-Line Handwritten Documents », *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems (DAS '08)*, p. 95-102, 2008.

Porter M. F., « An Algorithm for Suffix Stripping », *Program*, vol. 14, n^o 3, p. 130-137, 1980.

Salton G., Wong A., Wang C. S., « A Vector Space Model for Automatic Indexing », *Communications of the ACM*, vol. 18, n^o 11, p. 613-620, 1975.

Sebastiani F., « Machine Learning in Automated Text Categorization », *ACM Computing Surveys*, vol. 34, n^o 1, p. 1-47, 2002.

Spärck Jones K., « Experiments in Relevance Weighting of Search Terms », *Information Processing & Management*, vol. 15, p. 133-144, 1979.

Taghva K., Nartker T. A., Borsack J., Lumos S., Condit A., Young R., « Evaluating Text Categorization in the Presence of OCR Errors », *Proceedings of Document Recognition and Retrieval VII, IS&T/SPIE Int Symposium on Electronic Imaging (DRR '00)*, vol. 4307, p. 68-74, 2000.

Vapnik V., *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

Vinciarelli A., « Noisy Text Categorization », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n^o 12, p. 1882-1895, 2005.

Yang Y., « A Study of Thresholding Strategies for Text Categorization », *Proceedings of the 24th Annual International ACM SIGIR Conference (SIGIR '01)*, p. 137-145, 2001.

Yang Y., Liu X., « A Re-examination of Text Categorization Methods », *Proceedings of the 22nd Annual International ACM SIGIR Conference (SIGIR '99)*, p. 42-49, 1999.

Yang Y., Pedersen J. O., « A Comparative Study on Feature Selection in Text Categorization », *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, p. 412-420, 1997.

Système de recherche d'information à base d'inclusion graduelle

L. Ughetto[1]— **O. Pivert**[2]— **V. Claveau**[3]— **P. Bosc**[2]

1 *IRISA - Université Rennes 2 - Campus de Beaulieu, F-35042 Rennes cedex, France*
2 *IRISA - ENSSAT - BP 80518, F-22305 Lannion, France*
3 *IRISA - CNRS - Campus de Beaulieu, F-35042 Rennes cedex, France*
{laurent.ughetto,vincent.claveau}@irisa.fr
{pivert,bosc}@enssat.fr

RÉSUMÉ. *Cet article étudie, d'un point de vue expérimental, l'apport des inclusions graduelles issues de la théorie des ensembles flous pour la modélisation d'un système de recherche d'information (SRI), comme l'ont proposé de manière théorique (Bosc* et al.*, 2008b). Documents et requêtes sont représentés par des ensembles flous, appariés par des opérateurs flous, dont le choix est crucial pour obtenir un système adapté à la RI. S'ils sont bien choisis, le SRI flou obtenu est proche des SRI classiques et obtient des résultats aussi bons, en conservant l'avantage de son cadre théorique fort. À l'inverse, l'examen d'opérateurs inadaptées à la RI souligne les propriétés requises par ce SRI flou. Enfin, nous montrons la valeur ajoutée de ce modèle flou, qui permet d'envisager des extensions du modèle très naturelles. Un exemple simple montre comment utiliser une base de liens morphologiques entre mots dans ce cadre.*

ABSTRACT. *This paper investigates, from an experimental point of view, the use of graded inclusions, from fuzzy sets theory, to model an information retrieval system (IRS) as theoretically proposed by (Bosc* et al.*, 2008b). Documents and queries are represented by fuzzy sets, which are paired with fuzzy operators. It is shown that the fuzzy logic settings are crucial in order to obtain a system suited for IR. With appropriate settings, it is possible to mimic classical systems, thus yielding results rivaling those of state-of-the-art systems, while preserving its strong theoretical framework. Conversely, examining operators providing poor results sheds light on the necessary properties of such a system. Last, the added-value of using this model is shown by considering possible extensions; in particular, a short experiment shows how one can make the most of morphological links in fuzzy IRSs.*

MOTS-CLÉS : *modèles de SRI, logique floue, inclusion graduelle, implication floue, expressivité*

KEYWORDS: *IRS models, fuzzy logic, graded inclusion, fuzzy implication, query expressiveness*

1. Introduction

Recherche d'information (RI) et bases de données (BD) partagent l'objectif de fournir aux utilisateurs les informations qu'ils demandent. Cependant, il est bien connu que les approches classiques d'interrogation utilisées en BD ne sont pas utilisables en RI. Tout d'abord, elles n'ont pas la flexibilité requise en RI pour réaliser un appariement approximatif entre les mots des requêtes et les documents. Ensuite, elle proposent rarement un moyen de classer les résultats fournis. Cependant, de récentes études sur les BD floues et l'interrogation flexible des BD ont apporté de nouveaux mécanismes théoriques d'interrogation plus adaptés à la RI, comme par exemple (Bosc *et al.*, 2008b).

L'objectif principal de notre étude est de montrer la validité expérimentale du cadre théorique proposé dans (Bosc *et al.*, 2008b), et de construire un SRI flou, à base d'inclusion graduelle. Dans ce modèle, documents et requêtes sont représentés par des ensembles flous, appariés par une inclusion graduelle, c'est-à-dire en utilisant des opérateurs comme les implications floues et les T-normes. Les principaux paramètres du modèle sont l'implication floue qui calcule le degré d'inclusion, la T-norme d'agrégation des scores, et les pondérations de termes dans les documents et les requêtes. Pour définir les poids de façon automatique, des schémas de pondération classiques sont utilisés. Des expérimentations, dont seulement quelques résultats sont détaillés en Section 5.3, ont été menées en faisant varier ces nombreux paramètres. Avec un bon choix d'opérateurs flous et de pondérations, des résultats positifs montrent que le SRI flou proposé obtient des résultats comparables aux systèmes de type OKAPI. Les résultats négatifs, quant-à eux, permettent de définir des propriétés que le SRI flou doit vérifier pour être performant. Elles montrent ainsi comment le cadre théorique de (Bosc *et al.*, 2008b) peut être adapté à la RI. Certaines de ces propriétés sont d'ailleurs bien connues en RI.

Enfin, cette étude montre que le modèle de SRI flou proposé est susceptible d'offrir de meilleures interactions avec l'utilisateur, et lui permet d'écrire des requêtes plus expressives ou de pondérer facilement les termes de sa requête pour exprimer des préférences, ou encore inclure des informations négatives.

Tout d'abord, la section 2 compare notre travail à des approches existantes utilisant la logique floue (LF) en RI. Ensuite, la section 3 présente brièvement le contexte théorique des inclusions graduelles (le lecteur peut se reporter à (Bosc *et al.*, 2008b) pour plus de détails). L'implémentation et les résultats expérimentaux du SRI flou sont ensuite détaillés et commentés dans les sections 4 et 5, en reprenant principalement les résultats obtenus dans (Bosc *et al.*, 2009). Enfin, diverses extensions possibles du modèle sont proposées en section 6.

2. Travaux connexes

Certains aspects de la théorie des ensembles flous (ou de la logique floue) ont été utilisés dans les modèles de RI depuis le début des années 1980 (Buell, 1982, inter

alia). C'est assez naturel dans la mesure où le modèle de RI Booléen a rapidement été étendu à des modèles utilisant des degrés, et que la LF est une extension de la logique Booléenne utilisant des degrés (de vérité). Plusieurs travaux ont introduit de la LF dans des modèles de RI, avec des objectifs bien différents. Cela a été fait par exemple pour gérer l'incertitude dans la représentation des termes (Kraft *et al.,* 2006), améliorer le classement des documents (Boughanem *et al.,* 2005), ou accroître l'expressivité du langage de requête... D'autres ont étendu le modèle classique de RI pour prendre en compte des situations particulières, comme l'utilisation d'échelles de pondération ordinales des termes (Herrera-Viedma, 2001), ou utiliser à la fois des mesures de possibilité et de nécessité pour pondérer les termes (Brini *et al.,* 2005). Mais la plupart de ces travaux n'utilisent la LF que de façon ponctuelle, pour traiter un problème particulier, alors que notre approche propose un cadre théorique complet.

Parmi les travaux qui utilisent de la logique floue dans le mécanisme d'appariement entre requêtes et documents, on peut noter les travaux récents de Herrera-Vielma et al. (Herrera-Viedma *et al.,* 2007) ou Oussalah et al. (Oussalah *et al.,* 2008). Ces derniers proposent aussi d'utiliser une implication floue pour calculer la similarité entre un document et une requête. Toutefois, leur modèle calcule $D \rightarrow Q$, comme cela se fait souvent dans les approches logiques de la RI (voir (Lalmas, 1998)), alors que l'implication est utilisée dans l'autre sens dans notre modèle, pour des raisons détaillées dans la section suivante.

Notre modèle pourrait aussi être comparé à des travaux comme ceux de Salton et al. (Salton *et al.,* 1983), dans la mesure où c'est aussi une extension du modèle Booléen, qui se situe entre ce modèle Booléen et l'approche algébrique (VSM).

3. RI et division de relations

Les systèmes de recherche d'information (SRI) sont fondés sur des modèles caractérisés par 3 aspects principaux : la représentation des documents, le langage de requête et le mécanisme d'appariement. Cette section montre que le SRI flou proposé ici est une généralisation du modèle Booléen sur ces 3 aspects. Les sous-sections détaillent successivement l'approche Booléenne et la façon dont elle est liée à la division de relations, puis comment l'extension de la division aux relation floues (ou graduelles) est liée à une approche de RI graduelle, et enfin les fondements théoriques de notre SRI flou.

3.1. *Division de relations et approche Booléenne en RI*

Dans le modèle de données relationnel, un univers est modélisé par un ensemble de relations manipulées par les opérateurs de l'algèbre relationnelle. Parmi ces opérateurs, la division de la relation $C(A, X)$ par $Q(A)$, notée $C[A \div A]Q$, où A est l'ensemble des attributs communs à C et Q, détermine les valeurs sur X reliées dans

C à toutes les valeurs de A présentes dans Q. Cette opération peut être définie de plusieurs façons équivalentes :

$$x \in C[A \div A]Q \quad \Leftrightarrow \quad \forall a \in Q,\ (x,a) \in C\ , \qquad [1]$$

$$x \in C[A \div A]Q \quad \Leftrightarrow \quad Q \subseteq \Omega^{-1}(x) \quad \text{où} \quad \Omega^{-1}(x) = \{a | (x,a) \in C\}\ . \qquad [2]$$

Dans le modèle Booléen de RI, un document d est un ensemble de termes, et une requête peut être représentée par un ensemble P de termes désirés (ou positifs) et éventuellement un ensemble N de termes à exclure (ou négatifs). Un document d est pertinent s'il contient tous les termes positifs ($P \subseteq d$), et aucun des termes négatifs ($d \cap N = \emptyset$). Les opération ensemblistes jouent donc un rôle central dans de tels SRI.

Supposons que chaque document d est représenté par un ensemble de termes $d = \{t_1, \ldots, t_m\}$, où $t_i \in T$, l'ensemble des termes d'indexation, et une requête q contient seulement des termes positifs $P = \{t'_1, \ldots, t'_n\}$ où $t'_i \in T$. L'ensemble des documents de la collection peut être représenté par une relation normalisée (C_N) dans laquelle un document de m termes est représenté par m n-uplets $\langle d, t_1\rangle, \ldots, \langle d, t_m\rangle$, et la requête par une relation unaire P. La réponse à la requête est alors le résultat de la division de C_N par P.

L'approche Booléenne, à l'origine des SRI, est donc clairement liée aux mécanismes d'interrogation des BD, et en particulier la division de relations. Toutefois, elle a rapidement montré ses limites et n'est plus utilisée en RI depuis longtemps. Parmi les raisons, cette approche ne permet pas de représenter et d'utiliser l'importance relative des termes d'indexation des documents ou des requêtes.

3.2. *Extension graduelle de l'approche booléenne, et division de relations floues*

La plupart des extensions du modèle booléen ont pris en compte la notion d'importance relative des termes par des mécanismes de pondération.

Ce mécanisme est naturel en logique floue. Il consiste à représenter un document par un sous-ensemble flou des termes d'indexation T (Buell, 1982). Chaque terme t appartient à un document d de la collection C à un certain degré $\mu_C(d,t)$, qui représente son degré de représentativité (Waller *et al.*, 1979, Buell *et al.*, 1981). En théorie des sous-ensembles flous, la fonction μ est la fonction d'appartenance d'un élément à un ensemble, et la valeur $\mu_E(x)$ (dans l'intervalle unité) représente le degré d'appartenance de l'élément x à l'ensemble E. De même, une requête q peut aussi être un sous-ensemble flou de T, ou une requête plus complexe, structurée avec des opérateurs logiques flous (ET, OU, NON) (Bookstein, 1980). La pondération des termes des requêtes $\mu_q(t)$ a posé le problème de l'interprétation des poids attribués. La sémantique de ces poids $\mu_q(t)$ est discutée plus bas.

Ensuite, on retrouve les deux étapes classiques des SRI. Tout d'abord, l'utilisation d'une fonction d'appariement qui calcule des scores individuels $S_q(d,t)$ pour chaque terme t d'une requête q et chaque document d. Ensuite, l'utilisation d'une fonction d'agrégation des scores individuels $S_q(d,t)$, $t \in q$, pour obtenir un score global $S_q(d)$ pour chaque document, qui évalue le degré de satisfaction du document pour la requête. Ce degré permet un classement des documents jugés pertinents pour la requête. Dans un SRI flou, on peut utiliser des fonctions d'appariement et d'agrégation floues, donc à valeur dans l'intervalle unité.

Cette extension graduelle peut sembler ad hoc. Toutefois, comme précédemment, on peut obtenir la réponse à une requête q en effectuant la division de 2 relations, mais des relations floues, c'est-à-dire dont les n-uplets sont pondérés : celle des documents $C(d,t)$ et celle de la requête $Q(t)$. Pour cela, on généralise l'expression (2) au cas des relations floues, en remplaçant l'inclusion classique par un opérateur d'inclusion graduel g :

$$C[T \div T]Q(d) = g(Q \subseteq \Omega^{-1}(d)) \ , \qquad [3]$$

où $\Omega^{-1}(d)$ est un ensemble flou de termes défini par : $\Omega^{-1}(d) = \{\mu/t | \mu/(d,t) \in C\}$. Dans la notation μ/t, μ est le degré d'appartenance de l'élément t. Cette notation sert souvent à décrire en extension les sous-ensembles flous définis sur des domaines discrets.

La sémantique de la division ainsi obtenue dépend à la fois de l'opérateur d'inclusion et de la sémantique des poids associés aux n-uplets dans les relations C et Q (Bosc *et al.*, 1997). Une façon de modéliser l'inclusion graduelle $g(Q \subseteq \Omega^{-1}(d))$ consiste à utiliser une implication floue (notée $\rightarrow$ dans la suite), ce qui conduit à la formule :

$$g(Q \subseteq \Omega^{-1}(d)) = \min_{t \in Q} \left(\mu_Q(t) \rightarrow \mu_C(d,t)\right) \ . \qquad [4]$$

Dans cette formule, on retrouve la fonction d'appariement (l'implication), et la fonction d'agrégation (le $\min$).

3.3. *Inclusions graduelles et implications floues*

Dans la formule (4), l'opérateur d'appariement est une implication. Selon la nature de l'implication utilisée (R- ou S-implication), on obtient une interprétation différente des degrés d'inclusion, pour une sémantique différente des poids des termes dans les requêtes. On peut trouver dans (Bosc *et al.*, 2008b) un exemple de calculs réalisés avec diverses implications, qui illustre bien ces différences.

R-implication et notion de seuil. Une première approche consiste à interpréter le degré $\mu_Q(t)$ d'un terme t dans une requête Q comme un seuil à atteindre. On est alors totalement satisfait d'un document dès que ce seuil $\mu_Q(t)$ est atteint pour chaque terme t de la requête Q. Lorsque le seuil n'est pas atteint, le document reçoit une pénalité.

Ce comportement est obtenu avec une implication résiduée (ou R-implication), notée $\rightarrow_R$ et définie par (Fodor *et al.,* 1999) :

$$p \rightarrow_R q \ = \ \sup \{u \in [0,1] | \top(p,u) \leq q\} \ , \qquad [5]$$

où $\top$ est une norme triangulaire (i.e. une conjonction floue). Toute R-implication peut aussi s'écrire sous la forme :

$$p \rightarrow_R q = 1 \text{ si } p \leq q, \ f(p,q) \text{ sinon}, \qquad [6]$$

où $f(p,q)$ exprime une satisfaction partielle (une valeur < 1) lorsque l'antécédent p n'est pas atteint par la conclusion q. L'élément minimal de cette classe d'opérateur est connue sous le nom d'implication de Gödel :

$$p \rightarrow_{Gd} q = 1 \text{ si } p \leq q, \ q \text{ sinon},$$

obtenue en choisissant la plus grande T-norme $\top(a,b) = min(a,b)$ dans la formule (5). Parmi les R-implications très utilisées, on peut noter les implication de Goguen et de Lukasiewicz, obtenues respectivement avec le produit $\top(a,b) = a \cdot b$ ou la T-norme de Lukasiewicz $\max(a+b-1,0)$) :

$$p \rightarrow_{Gg} q = 1 \text{ si } p \leq q, \ q/p \text{ sinon},$$
$$p \rightarrow_{Lu} q = 1 \text{ si } p \leq q, \ 1-p+q \text{ sinon}.$$

Avec des R-implications, on voit clairement sur la formule (6) que $\mu_Q(t)$ doit être interprété comme un seuil, puisqu'on obtient une satisfaction maximale (le degré 1) dès que $\mu_C(d,t)$ atteint $\mu_Q(t)$.

S-implication et notion d'importance. On peut aussi voir $\mu_Q(t)$ comme un degré d'importance du terme t dans la requête, par rapport à l'information recherchée. Le cadre logique des implications conduit alors à imposer un degré de satisfaction garantie pour un document lorsque l'importance du terme recherché t est inférieure à 1. En effet, lorsque $\mu_Q(t) < 1$, le terme n'est pas totalement requis et peut donc être absent dans une certaine mesure. La satisfaction totale nécessite que $\mu_C(d,t) = 1$ pour chaque valeur t de Q quelle que soit son importance. Un document n'est pas du tout satisfaisant ($\mu_{C[T \div T]Q}(d) = 0$) seulement lorsque, pour au moins un terme de la requête, $\mu_Q(t) = 1$ (le terme est d'importance maximale) et $\mu_C(d,t) = 0$ (le terme n'est pas du tout représentatif du document). Ce comportement peut être modélisé en utilisant une S-implication (Fodor *et al.,* 1999) notée $\rightarrow_S$:

$$p \rightarrow_S q \ = \ \perp(1-p,q) = 1 - \top(p,1-q) \ , \qquad [7]$$

où $\perp$ est une conorme triangulaire (ou T-conorme, ou S-norme).

Comme dans le cas des R-implications, il existe une infinité de S-implications, en fonction de la norme génératrice choisie. La plus utilisée est l'implication de Kleene-Dienes définie par :

C	t_1	t_2	t_3	t_4
d_1	1	0.9	1	0.2
d_2	0.7	0.6	0.3	0.8

	t_1	t_2	t_3	t_4
q	1	0.4	0	0.6
q'	0.6	0.6	0.3	0.5

	sémantique du poids des requêtes	implication	d_1	d_2
q	importance	Kleene-Dienes	0.4	0.6
		Reichenbach	0.52	0.76
q'	seuil de satisfaction	Gödel	0.2	1
		Goguen	0.4	1
		Lukasiewicz	0.7	1

Tableau 1. *En haut à gauche : relation floue C représentant la collection — En haut à droite : chaque ligne est une relation floue Q représentant une requête — En bas : résultats de la division selon l'implication choisie.*

$$p \rightarrow_{KD} q \;=\; \max(1-p, q) \;\; .$$

C'est l'élément minimal dans la formule (7), obtenu avec la plus petite conorme $\perp$, i.e., le maximum. Lorsqu'on choisit pour $\perp$ la somme probabiliste, on obtient l'implication de Reichenbach :

$$p \rightarrow_{Rb} q \;=\; 1 - p + p \cdot q \;\; .$$

L'implication de Lukasiewicz, vue plus haut, est aussi une S-implication générée par $\perp(a, b) = \min(a + b, 1)$.

Enfin, on peut remarquer que la formule (4) correspond à la division de relations classiques lorsque les termes ne sont pas pondérés, puisque les implications floues généralisent l'implication classique (en particulier elles conservent $1 \rightarrow 0 = 0$ et $1 \rightarrow 1 = 1$).

Effet d'absorption. L'approche logique proposée est de type conjonctif, et produit un *effet d'absorption*. En effet, l'opérateur de division, et en particulier l'agrégation par le $\min$ dans (4), donne comme résultat global le plus petit des degrés d'implication entre un terme de la requête et le document. Pour éviter cet effet néfaste, la formule (4) est relâchée par l'utilisation d'une autre T-norme que le $\min$ dans notre modèle de RI.

Exemple. La table 1 donne les relations floues représentant une collection de deux documents d_1 et d_2, deux requêtes q et q', et les résultats, en fonction de la sémantique choisie. Un effet de seuil apparaît clairement avec q' et d_2. Cet exemple est tiré de (Bosc *et al.*, 2008b).

4. Implémentation et caractéristiques du SRI

Notre SRI implémente l'approche floue décrite dans la section 3. Ainsi, le score d'un document d, pour une requête q est calculé de la façon suivante :

$$S(d,q) = \top_{t \in q}(w_q(t) \rightarrow w_d(t)) \ , \qquad [8]$$

où t est un terme de la requête, $w_q(t)$ son poids dans la requête, $w_d(t)$ (qu'on peut aussi noter $w_C(d,t)$) son poids dans le document, $\rightarrow$ l'implication floue correspondant à l'inclusion graduelle choisie, et $\top$ la T-norme d'agrégation. On voit dans la formule (8) que de nombreux paramètres peuvent être ajustés : le poids des termes dans le document ou la requête, et les opérateurs d'implication et d'agrégation.

Opérateur d'agrégation. Lorsque $\top$ est l'opérateur $\min$, le score $S(d,q)$ obtenu par le document d est le degré d'appartenance de d au quotient de la division floue de la collection par la requête q. Comme $\min$ est la plus grande T-norme, elle fournit le plus grand score $S(d,q)$. Ce score correspond au degré d'inclusion dans d du terme $t \in q$ le moins inclus dans d. Cette vision correspond à une approche BD classique, dans laquelle chaque terme de la requête doit se retrouver dans les n-uplets du résultat. C'est le degré d'inclusion du *terme le plus faible* qui donne la mesure de pertinence du document. Cette approche ne donne pas de bons résultats en RI.

En RI, un document pertinent ne contient pas toujours tous les termes de la requête. Dans la plupart des modèles vectoriels, lorsqu'un terme recherché est absent d'un document, il ne modifie pas le score du document ; il est neutre, ce qui correspond bien à l'agrégation des scores individuels des termes par une somme. Par contre, un terme très représentatif (fréquent dans le document et rare dans la collection) augmente beaucoup le score. Du point de vue de la RI, les *meilleurs termes* sont donc plus importants que les *plus mauvais*. De plus, pour pouvoir classer les documents, leur score final doit tenir compte de tous les scores individuels des termes, alors que l'agrégation par le $\min$ ne conserve que le poids d'un seul terme (le plus mauvais). C'est pour cette raison que l'équation (4) a été relâchée en (8) qui reste une mesure d'inclusion, et que de nombreuses T-normes ont été testées, comme : min, Drastic, Einstein, Lukasiewicz, Product, ou des T-normes paramétrées comme : Dubois et Prade, Hamacher, Yager...

Opérateur d'inclusion graduelle. Comme on l'a vu section 3, deux classes d'opérateurs ont été testées : les R- et les S-implications. Les plus représentatives (et utilisées) de chaque classe ont été choisies pour la première série de tests. Parmi les R-implications : Gödel, Goguen, Lukasiewicz. Parmi les S-implications : Kleene-Dienes, Lukasiewicz, Reichenbach, Willmott. Le lecteur peut trouver leur définition dans de nombreux articles, par exemple (Fodor *et al.*, 1999).

Poids des termes dans les documents. Dans le contexte de la division de relations floues, les poids doivent avoir une sémantique claire (importance, seuil, préférence...). Le schéma de pondération classique d'OKAPI-BM25 a été choisi car il véhicule la notion d'importance relative des termes. Toutefois, comme ils sont ma-

nipulés par des opérateurs flous, les poids doivent être ramenées à l'intervalle unité $[0, 1]$. Les poids d'OKAPI-BM25 ($w_{BM25}(t, d)$) ont donc été normalisés et bornés.

Poids des termes dans les requêtes. Comme pour les poids des termes dans les documents ($w_d(t)$), les poids des termes dans les requêtes ($w_q(t)$) doivent avoir une sémantique claire, de façon à pouvoir être comparés à ceux des documents. C'est d'une importance particulière avec des R-implications, pour lesquelles $w_q(t)$ est un degré de satisfaction à atteindre par $w_d(t)$. Pour l'instant, la seule façon d'obtenir de tels poids dans les requêtes est de les fixer manuellement. Cependant, ils auraient été attribués de façon subjective, et n'auraient pas permis une comparaison équitable avec d'autres SRI. C'est pourquoi un mécanisme de pondération classique et automatique a été choisi, aux dépends de la sémantique, pour ces premiers tests dont l'objectif est la validation de notre modèle de RI. Les poids choisis dépendent de la fréquence des termes dans les requêtes, et sont normalisés et bornés.

5. Résultats expérimentaux

Les expérimentation ont été menées en faisant varier les différents paramètres du SRI. Les résultats ont été comparés à ceux du modèle OKAPI, avec des paramètres (k_1 et b) identiques à ceux utilisés dans Lemur. Cette section montre à la fois de bons résultats, qui valident le modèle proposé, et de mauvais résultats, accompagnés d'explications sur les causes probables. Par manque de place, seuls quelques résultats jugés caractéristiques sont détaillés.

5.1. *Collections de documents*

Le SRI proposé a été testé sur 3 collections. La première, nommée ELDA, est une petite collection en français, contenant 3499 documents (des questions/réponses de la commission européenne), et un ensemble de 19 requêtes. La deuxième est le collection INIST, qui contient 163.308 documents (des résumés d'articles de diverses disciplines scientifiques) et 30 requêtes. Ces deux collections proviennent de la campagne d'évaluation de RI Amaryllis. La troisième, TIPSTER, est une collection de TREC-3 contenant 173.252 articles du Wall Street Journal et 50 requêtes.

Pour toutes les collections, les documents et requêtes ont été lemmatisées. Les requêtes sont composées de plusieurs champs : titre, corps, description et concepts associés. Dans les expérimentations, seuls le titre et les concepts associés ont été utilisés (sauf pour TIPSTER qui ne contient pas de concepts associés).

5.2. *Propriétés qui conduisent à de mauvais résultats*

Les effets d'absorption et de seuil sont les principaux responsables des mauvais résultats. Ils peuvent intervenir à différents niveaux, impliquant l'opérateur d'agrégation, l'implication ou les poids.

Élément absorbant des T-normes. Dans notre modèle, les scores individuels des termes sont agrégés par une conjonction. Les conjonctions ont un élément absorbant (le zéro) qui pose le problème suivant : dès qu'un terme reçoit le score 0, le document aussi, quels que soient les poids des autres termes.

Le score d'un terme est donné par : $w_q(t) \rightarrow w_d(t)$. Avec la plupart des R-implications, ce score vaut 0 dès que $w_d(t) = 0$, i.e. dès que le terme est absent du document. Avec une S-implication, le score vaut $1 - w_q(t)$ dans ce cas, et il n'est nul que lorsque $w_q(t) = 1$, i.e. lorsque le terme est absent du document et a une importance maximale.

Pour contourner ce problème, la stratégie adoptée est la même que dans les modèles de langue, qui utilisent des techniques de lissage : un mot absent d'un document reçoit un score faible prédéfini, strictement positif. Cela signifie qu'un terme, même absent d'un document *peut* être représentatif de ce document (par exemple il peut être synonyme d'un terme du document).

Effet de seuil des R-implications. Avec une R-implication, $w_q(t)$ est le degré minimal attendu pour $\mu_d(t)$ dans les documents totalement pertinents. Dès que $w_d(t) \geq w_q(t)$, ce degré est donc atteint et le score du terme t, obtenu par $w_q(t) \rightarrow w_d(t)$, vaut 1. La mauvaise conséquence est que deux documents ayant des poids différents $w_{d_1}(t) \neq w_{d_2}(t)$, mais tous deux au dessus du seuil $w_q(t)$, obtiennent le même score 1 et ne peuvent donc être classés. Ici encore, ce comportement reflète une approche BD classique, dans laquelle le système doit seulement retrouver tous les n-uplets pertinents, et n'a pas à les classer. En RI, les documents doivent être présentés par ordre de pertinence décroissante, comme cela se fait dans les approches de requêtage flexible des BD. C'est la raison pour laquelle les R-implications conduisent à de mauvais résultats dans le cas général. On pourrait contourner ce problème en choisissant dans les requêtes des poids $w_q(t)$ (les seuils requis) toujours supérieurs aux poids de ces termes dans les documents. Dans ce cas, le seuil n'est jamais atteint, les scores individuels sont toujours strictement inférieurs à 1, et les documents peuvent être classés. Les deux types d'implications donnent alors des résultats proches, aux dépends de la sémantique.

Effet d'absorption des opérateurs de type min. Certains opérateurs d'agrégation ont un effet d'absorption, comme $\min$, $\max$... Avec cette classe d'opérateurs, seulement un des termes (ou un petit nombre) est pris en compte pour le calcul du score global du document. La mauvaise conséquence est, encore une fois, un mauvais classement des documents, qui conduit à de mauvais résultats. Ce type d'opérateur est donc à éviter.

5.3. *Résultats*

Parmi les nombreuses combinaisons possibles d'opérateurs et poids que nous avons testées, cette section présente seulement quelques jeux de paramètres, représentatifs aussi bien des bons que des mauvais résultats. La table 2 donne les résultats pour l'implication de Reichenbach associée au produit et à la T-norme d'Einstein ($a \top_E b = (a.b)/(2 - a + b - a.b)$), et pour l'implication de Lukasiewicz, associée au Produit et à la T-norme de Lukasiewicz ($a \top_L b = \max(0, a + b - 1)$).

Les résultats sont évalués en terme de précision moyenne non interpolée (MAP), précision moyenne interpolée (IAP), R-précision (Rprec), et précision pour les k premiers documents (Pk). La différence relative (Rdiff) avec OKAPI est aussi indiquée. Les valeurs en gras sont considérées comme statistiquement significatives par un T-test.

INIST	OKAPI	SRI à base d'inclusion graduelle							
implic.		Reichenbach				Lukasiewicz			
t-norme		Einstein	Rdiff	Product	Rdiff	Lukasiewicz	Rdiff	Product	Rdiff
MAP %	21.75	**23.22**	**(+6.79%)**	**23.13**	**(+6.37%)**	**0.03**	**(-99.85%)**	**23.03**	**(+5.90%)**
IAP %	24.13	**25.60**	**(+6.10%)**	**25.50**	**(+5.70%)**	**0.20**	**(-99.17%)**	25.38	(+5.17%)
Rprec %	25.85	28.20	(+9.09%)	27.94	(+8.08%)	**0.03**	**(-99.90%)**	**28.09**	**(+8.69%)**
P5 %	50.00	45.33	(-9.33%)	49.33	(-1.33%)	**0.00**	**(-100.00%)**	48.00	(-4.00%)
P10 %	42.67	42.67	(0.00%)	42.00	(-1.56%)	**0.00**	**(-100.00%)**	43.67	(+2.34%)
P100 %	17.03	**18.27**	**(+7.24%)**	**18.20**	**(+6.85%)**	**0.03**	**(-99.80%)**	**18.23**	**(+7.05%)**
P500 %	5.39	**5.64**	**(+4.70%)**	**5.61**	**(+4.08%)**	**0.03**	**(-99.38%)**	**5.63**	**(+4.58%)**

ELDA	OKAPI	SRI à base d'inclusion graduelle							
implic.		Reichenbach				Lukasiewicz			
t-norme		Einstein	Rdiff	Product	Rdiff	Lukasiewicz	Rdiff	Product	Rdiff
MAP %	57.14	56.86	(-0.49%)	56.91	(-0.42%)	**1.11**	**(-98.06%)**	56.29	(-1.50%)
IAP %	58.09	57.89	(-0.36%)	57.88	(-0.37%)	**1.98**	**(-96.59%)**	57.38	(-1.23%)
Rprec %	55.33	53.82	(-2.73%)	54.64	(-1.26%)	**0.67**	**(-98.78%)**	**53.03**	**(-4.16%)**
P5 %	77.24	76.55	(-0.89%)	74.48	(-3.57%)	**1.38**	**(-98.21%)**	75.17	(-2.68%)
P10 %	68.28	68.62	(+0.51%)	68.97	(+1.01%)	**0.69**	**(-98.99%)**	67.93	(-0.51%)
P100 %	27.00	26.86	(-0.51%)	26.83	(-0.64%)	**1.00**	**(-96.30%)**	26.83	(-0.64%)
P500 %	6.67	6.66	(-0.10%)	6.67	(+0.00%)	**0.87**	**(-86.97%)**	6.66	(-0.10%)

TIPSTER	OKAPI	SRI à base d'inclusion graduelle							
implic.		Reichenbach				Lukasiewicz			
t-norme		Einstein	Rdiff	Product	Rdiff	Lukasiewicz	Rdiff	Product	Rdiff
MAP %	18.14	18.61	(2.61%)	18.66	(2.87%)	**2.53**	**(-86.08%)**	18.66	(2.87%)
IAP %	20.09	**20.83**	**(3.69%)**	20.90	(4.06%)	**2.70**	**(-86.55%)**	20.90	(4.02%)
Rprec %	22.42	22.85	(1.91%)	23.31	(4.00%)	**3.47**	**(-84.54%)**	23.32	(4.02%)
P5 %	31.60	32.40	(2.53%)	32.80	(3.80%)	**5.60**	**(-82.28%)**	32.80	(3.80%)
P10 %	30.40	32.00	(5.26%)	31.80	(4.61%)	**6.00**	**(-80.26%)**	32.00	(5.26%)
P100 %	17.14	17.14	(0.00%)	17.08	(-0.35%)	**3.64**	**(-78.76%)**	17.06	(-0.47%)
P500 %	7.33	7.37	(0.49%)	7.34	(0.11%)	**0.85**	**(-88.43%)**	7.35	(0.27%)

Tableau 2. *Résultats pour les collections INIST, ELDA et TIPSTER*

Lorsque les différents paramètres sont choisis pour éviter les mauvaises propriétés répertoriées plus haut, et grâce à la pondération BM-25, les résultats de notre SRI sont positifs, et comparable à ceux d'OKAPI (parfois même un peu meilleurs), ce qui se vérifie par l'absence de différences jugées statistiquement significatives.

Opérateurs. Pour l'ensemble des collections, les meilleurs résultats sont obtenus avec l'implication de Reichenbach associée à la T-norme produit ou Einstein. Dans quelques cas, l'implication de Lukasiewicz, et la pseudo-implication de Larsen (le produit), donnent aussi de bons résultats. Les implications paramétrées donnent aussi de bons résultats, mais principalement lorsque leur comportement se rapproche du produit.

Il est intéressant de constater que l'utilisation de T-conormes (des disjonctions) pour l'agrégation des scores, produit souvent des résultats similaires (bien qu'inférieurs de quelques pourcents) à ceux obtenus avec les T-normes associées. Surprenant au premier abord (car OU signifie « un au moins », alors que ET veut dire « tous »), ce résultat s'explique par le fait que ce n'est pas la valeur finale du score qui importe, mais la façon de prendre en compte les scores individuels des termes pour produire le score final du document, qui conditionne le classement. Or, T-norme et T-conorme associées ont souvent un comportement similaire de ce point de vue.

6. Expressivité du modèle à base d'inclusion graduelle

6.1. *Expressivité de la requête*

Un schéma de pondération classique a été utilisé pour valider notre modèle. Toutefois, la fréquence des termes dans les requêtes ne représente pas vraiment l'importance des termes par rapport au besoin d'information de l'utilisateur, en particulier lorsque les requêtes sont des phrases plutôt que seulement des mots-clés. S'il n'est pas possible en général de demander à l'utilisateur de pondérer les termes de ses requêtes par des nombres réels, l'approche graduelle proposée permet cependant de simplifier la pondération manuelle, par exemple en demandant à l'utilisateur de présenter ses termes par ordre d'importance, d'utiliser une échelle d'importance ordinale, ou de les saisir par catégorie d'importance (par exemple en remplissant 3 à 5 cases d'importance décroissante dans un formulaire).

Ce modèle flou permet également de prendre en compte très naturellement l'utilisation d'opérateurs flous dans les requêtes. De nombreux travaux se sont intéressés à ces opérateurs, que ce soient des ET/OU flous (Herrera-Viedma *et al.*, 2007), ou des opérateurs moins standard (Mercier *et al.*, 2006). Dans notre SRI, cela peut servir à prendre en compte les concepts associés aux requêtes, en particuliers lorsqu'ils sont composés de plusieurs mots. Par exemple, *« pollution de l'air », « effet de serre »* peuvent donner de meilleurs résultats lorsqu'ils sont exprimés par (*pollution* AND *air*) OR (*effet* AND *serre*), plutôt que par 4 mots indépendants. La richesse des opérateurs de logique floue permet aussi de moduler la sémantique des conjonctions et disjonctions. Par exemple, min/max véhiculent la notion d'indépendance. Dans l'expression :

$$\max(\min(\mu_d(\textit{pollution}), \mu_d(\textit{air})), \min(\mu_d(\textit{effet}), \mu_d(\textit{serre}))) ,$$

la disjonction max signifie que le « meilleur » des concepts associés suffit : il donne le score global. Le min signifie que les 2 termes doivent être présents dans le document, car le score du concept est égal au score du « plus mauvais » terme. D'autres opérateurs, comme le produit ou la somme probabiliste, véhiculent la notion de renforcement. Par exemple, avec la somme probabiliste à la place du max, dans l'expression précédente, plus il y a de concepts associés dans le document, plus le score est élevé.

Le modèle flou permet aussi de modéliser et d'exploiter des parties de requêtes négatives, c'est-à-dire des termes que l'on ne souhaite pas voir apparaître. Cela peut se faire en utilisant l'antidivision (Bosc *et al.*, 2008a), opération duale de la division. L'antidivision de $C(A, X)$ par $Q(A)$, renvoie les éléments x de C tels que $\forall a \in Q$, $(a, x) \notin C$, c'est-à-dire, dans notre modèle, les documents qui ne contiennent pas les termes négatifs.

Si la plupart de ces extensions ne sont pas vraiment originales, dans notre modèle elles peuvent reposer sur une approche bien fondée. Les opérateurs, les poids, et les résultats peuvent alors tirer profit d'une sémantique claire. Cela peut aider à obtenir de meilleurs résultats.

6.2. *Utilisation d'informations lexicales*

L'utilisation d'informations lexicales externes (e.g. des synonymes, des liens morphologiques...) peut aussi se faire naturellement. En RI, ces informations sont souvent utilisées pour étendre des requêtes (Voorhees, 1998, inter alia). Se pose alors le problème de leur pondération par rapport aux termes originaux de la requête et éventuellement de savoir quand arrêter l'enrichissement (doit-on prendre les termes liés aux termes liés...). Ces problèmes sont souvent résolus de manière ad-hoc (Moreau *et al.*, 2007, Voorhees, 1998, inter alia). Dans notre approche floue, ces informations peuvent servir à dilater le dividende de notre division, c'est-à-dire à enrichir le document. Plus formellement, l'inclusion basée sur ce principe peut se définir comme précédemment par :

$$g(Q \subseteq \Omega^{-1}(d)) = \min_{t \in Q} (\mu_Q(t) \rightarrow \mu_{dil(C)}(d, t)) \quad [9]$$

mais avec le degré d'appartenance du terme dans le document défini par :

$$\mu_{dil(C)}(d, t) = \top_{t' \in U} \perp(\mu_C(d, t'), \mu_{\text{rsb}}(t, t')) \quad [10]$$

Cette dernière équation rend compte de l'importance d'un terme dans un document comme dépendant de $\mu_{\text{rsb}}(t, t')$, la proximité du terme et d'un de ses mots liés (synonyme ou autre), et de $\mu_C(d, t')$, le poids de ce mot lié dans le document. Ces deux éléments sont combinés par une T-norme, et la T-conorme permet en quelque sorte de choisir le « meilleur » des mots liés au terme initial, ou de renforcer un terme dont beaucoup de mots liés apparaissent. Un exemple de l'utilisation de la dilatation est détaillé dans (Bosc *et al.*, 2008b).

L'avantage de cette formulation est de combiner simplement avec une T-norme la force du lien entre le mot requête et les mots liés présents dans le document et leurs poids. Les différentes T-normes et T-conormes possibles donnent différentes sémantiques à cette combinaison.

Pour illustrer l'intérêt de ce mécanisme, nous avons mis en place une expérience simple dans laquelle nous exploitons une base de liens morphologiques. Dans cette base construite automatiquement (voir (Moreau *et al.*, 2007) pour les détails de sa construction), un mot comme *pollution* est lié à *pollutions*, *polluant*, *anti-pollution*, *etc.* avec un score représentant la fréquence du lien morphologique. Ce score, une fois ramené dans l'intervalle $[0, 1]$, représente $\mu_{\text{rsb}}(t, t')$. La dilatation du dividende est mise en œuvre avec le produit comme T-norme, et la somme bornée comme T-conorme.

Pour ces expériences, nous utilisons de nouveau la collection INIST, mais sans lemmatisation pour faire ressortir au mieux l'intérêt des liens morphologiques. On se place donc dans le cas où on ne connait pas de techniques pré-existantes de lemmatisation (voir (Moreau *et al.*, 2007) pour une étude de l'influence de la lemmatisation dans ce cadre). Le tableau 3 présente les résultats obtenus avec les mêmes conventions que précédemment. Pour comparaison, nous indiquons les résultats obtenus par OKAPI dans lequel les requêtes sont étendues avec les mêmes mots liés morphologiquement aux termes des requêtes.

	OKAPI avec requêtes étendues	SRI flou avec dilatation du dividende	
MAP	15.36	**17.81**	**(+16.00 %)**
IAP	17.67	**20.13**	**(+13.92 %)**
Rprec	20.13	**23.14**	**(+14.97 %)**
P5	40.67	**42.67**	**(+4.92 %)**
P10	35.67	36.33	(+1.87 %)
P100	14.63	**16.23**	**(+10.93 %)**
P500	5.14	5.19	(+0.91 %)

Tableau 3. *Inclusion d'informations morphologiques par dilatation du dividende*

Les résultats de cette expérience abondent largement dans le sens de l'intérêt de ce mécanisme de dilatation puisque l'on constate une amélioration significative des résultats comparés à l'utilisation brute des liens morphologiques en extension de requête dans OKAPI. Le poids donné à ces variantes a été obtenu naturellement, à l'aide de la formule donnée ci-dessus. Bien entendu, une étude des différentes T-normes et T-conormes utilisables pour la dilatation et leur influence sur les résultats reste cependant à mener.

7. Conclusion

Le modèle de RI à base d'inclusion graduelle présenté dans cet article semble prometteur. Lorsque les paramètres sont bien choisis (cf. sections 4 et 5.3), on a montré que ce modèle donne des résultats comparables à ceux du marché, tout en apportant un cadre théorique fort. Des propriétés que le modèle doit avoir pour produire de bons résultats ont aussi pu être identifiés lors de cette étude.

Le plus intéressant réside cependant dans les extensions assez naturelles que permet ce modèle. Nous avons montré qu'il était prometteur pour l'utilisation de ressources lexicales externes habituellement utilisées en extension de requêtes. Il doit permettre également de construire et d'exploiter des requêtes beaucoup plus expressives, tout en conservant le cadre théorique et une sémantique claire des pondérations utilisées. En particulier, des procédures simples et intuitive de pondération des requêtes, ou d'enrichissement peuvent être exploitées. Nous serons cependant confrontés au problème de l'évaluation de ces techniques à grande échelle, par manque de collections de RI adaptées.

D'autres travaux concernant ce modèle sont aussi à l'étude. Par exemple, l'utilisation des inclusions tolérantes quantitative et qualitative proposées pour ce modèle (Bosc *et al.*, 2008b), doivent être validées expérimentalement. Ces inclusions tolérantes pourraient permettre de lever certaines restrictions que nous avons relevées dans cet article, quant-à l'emploi de certaines famille de T-normes. Enfin, les différentes expérimentations présentées dans cet article font ressortir le fait que les mécanismes de RI habituels se concentrent principalement sur l'intersection entre requêtes et documents, alors que ceux opérant dans le domaine des BD s'intéressent plutôt à l'inclusion, par la mesure de la quantité de termes des requêtes hors du document. Une définition de l'inclusion entre ensembles, basée cette fois sur la cardinalité, pourrait rapprocher plus encore ces deux philosophies.

8. Bibliographie

Bookstein A., « Fuzzy requests : an approach to weighted Boolean searches », *Journal of the American Society for Information Science*, vol. 31, p. 240-247, 1980.

Bosc P., Claveau V., Pivert O., Ughetto L., « On the use of tolerant graded inclusions in information retrieval », *Proceedings of the European Conference on Information Retrieval, ECIR'09*, Toulouse, France, 2009. à paraître.

Bosc P., Dubois D., Pivert O., Prade H., « Flexible queries in relational databases – The example of the division operator », *Theoretical Computer Science*, vol. 171, p. 281-302, 1997.

Bosc P., Pivert O., « On a Parameterized Antidivision Operator for Database Flexible Querying », *Proceedings of the 19th International Conference on Database and Expert Systems Applications, DEXA'08*, Turin, Italy, p. 652-659, 2008a.

Bosc P., Pivert O., « On the use of tolerant graded inclusions in information retrieval », *Actes de la 5e Conférence en Recherche d'Information et Applications, CORIA'08*, Trégastel, France, p. 321-336, 2008b.

Boughanem M., Loiseau Y., Prade H., « Improving Document Ranking in Information Retrieval Using Ordered Weighted Aggregation and Leximin Refinement », *Proceedings of the European Society for Fuzzy Logic and Technology Conference, EUSFLAT'05*, Barcelona, Spain, p. 1269-1274, 2005.

Brini A., Boughanem M., Dubois D., « A Model for Information Retrieval Based on Possibilistic Networks », *Proceedings of the 12th String Processing and Information Retrieval International Conference, SPIRE'05*, Buenos Aires, Argentina, p. 271-282, 2005.

Buell D., « An analysis of some fuzzy subset applications to information retrieval systems », *Fuzzy Sets & Systems*, vol. 7, p. 35-42, 1982.

Buell D., Kraft D., « Threshold values and Boolean retrieval systems », *Information Processing & Management*, vol. 17, p. 127-136, 1981.

Fodor J., Yager R., *Fundamentals of Fuzzy Sets — The Handbook of Fuzzy Sets Series (D. Dubois and H. Prade eds.)*, Kluwer Academic Publishers, chapter Fuzzy Set-theoretic Operators and Quantifiers. Chap. 1.2, p. 125-193, 1999.

Herrera-Viedma E., « Modeling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach », *Journal of the American Society for Information Science and Technology*, vol. 52, p. 460-475, 2001.

Herrera-Viedma E., López-Herrera A., Luque M., Porcel C., « A Fuzzy Linguistic IRS Model Based on a 2-Tuple Fuzzy Linguistic Approach », *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 15, n^o 2, p. 225-250, 2007.

Kraft D. H., Pasi G., Bordogna G., « Vagueness and uncertainty in information retrieval : how can fuzzy sets help ? », *Proceedings of IWRIDL'2006*, p. 1-10, 2006.

Lalmas M., « Logical Models in Information Retrieval : Introduction and overview », *Information Processing & Management*, vol. 34, n^o 1, p. 19-33, 1998.

Mercier A., Imafouo A., Beigbeder M., « Using a Fuzzy Proximity Matching Function », *ENSM-SE at CLEF 2005*, vol. 4022/2006 of *LNCS*, p. 187-193, 2006.

Moreau F., Claveau V., Sébillot P., « Automatic morphological query expansion using analogy-based machine learning », *Proceedings of the European Conference on Information Retrieval, ECIR'07*, Rome, Italie, avril, 2007.

Oussalah M., Khan S., Nefti S., « Personalized information retrieval system in the framework of fuzzy logic », *Expert Systems with Applications*, vol. 35, p. 423-433, 2008.

Salton G., Fox E., Wu H., « Extended Boolean Information Retrieval », *Communications of the ACM*, vol. 26, n^o 12, p. 1022-1036, 1983.

Voorhees E., *C. Fellbaum (ed.), WORDNET : An Electronic Lexical Database*, The MIT Press, chapter Using WORDNET for Text Retrieval, p. 285-303, 1998.

Waller W., Kraft D., « A mathematical model of a weighted Boolean retrieval system », *Information Processing & Management*, vol. 15, p. 235-245, 1979.

Interactions entre le calcul de collocations et la catégorisation automatique de textes

Rémi Lavalley — Patrice Bellot — Marc El-Bèze

Laboratoire Informatique d'Avignon (UPRES 931)
339, chemin des Meinajaries
Agroparc – B.P. 1228
F-84911 Avignon cedex 9
{remi.lavalley, patrice.bellot, marc.elbeze}@univ-avignon.fr

RÉSUMÉ. *Nous proposons dans cet article d'étudier les interactions entre l'extraction de collocations et la catégorisation automatique de textes. C'est-à-dire, dans un premier temps, utiliser la répartition des textes dans les différentes classes afin d'extraire des chaînes spécifiques à chacune (calculées par agglutination de collocations) ; puis, dans un second temps, utiliser ces chaînes spécifiques pour améliorer la catégorisation.*

ABSTRACT. *In this paper we describe some interactions between collocations and automatic text categorization. First, we use the differents categories to extract strings (through collocations agglutinations) related to each categorie. Then we use these categories-specific strings to improve categorization.*

MOTS-CLÉS : *collocations, catégorisation automatique de textes.*

KEYWORDS: *collocation, automatic text categorization.*

1. Introduction

La masse croissante de textes disponibles nous pousse continuellement à chercher des méthodes facilitant leur manipulation. La catégorisation automatique de textes fait partie des nombreuses tâches de la recherche d'informations. Le problème consiste à rattacher un texte à une ou plusieurs catégories prédéfinies, ces catégories pouvant être par exemple le sujet du texte, son thème, l'opinion qui y est exprimée, ... Nous diposons pour cela d'un ensemble de textes pour lesquels la catégorie est connue (corpus d'apprentissage) et qui nous servent à entraîner nos modèles, modèles que nous exécuterons par la suite pour étiqueter automatiquement des documents de catégorie indéterminée (corpus de test).

Il existe un très grand nombre de méthodes numériques de catégorisation automatique de textes, qui se basent pour cela sur les mots contenus dans le texte. Des exemples de méthodes sont fournis dans (Yang *et al.,* 1999). La méthode proposée ici présente la particularité de n'utiliser non plus les mots isolés composant le texte, mais d'en regrouper certains (par la recherche de collocations) afin d'obtenir des termes ou expressions plus porteurs de sens.

Pour l'évaluation de nos propositions, nous avons utilisé le corpus de la campagne d'évaluation Défi Fouille de Textes 2007 (DEFT 07 (Grouin *et al.,* 2007)) portant sur la classification de textes selon l'opinion qui y est exprimée. Une partie du défi était basée sur des textes de critiques de jeux vidéos (4000 critiques, soit 28,3 Mo), pour lesquelles il fallait, en se basant sur le texte d'une critique, détecter si son auteur avait attribué une note bonne (classe 2), neutre (classe 1) ou mauvaise (classe 0) à ce jeu. Nous avons utilisé ce corpus car nous possédions déjà des systèmes de classification écrits spécifiquement pour lui lors de la campagne : le système présenté par le Laboratoire Informatique d'Avignon (LIA) (Torres-Moreno *et al.,* 2007) consistait en une combinaison de neuf classifieurs. Nous avons, pour notre travail, utilisé le classifieur *LIA_cosine* (calcul d'une distance - cosinus - entre les représentations vectorielles des documents (les poids des vecteurs sont fournis par TF.IFD (Salton *et al.,* 1988), combiné avec un facteur discriminant inspiré du critère d'impureté de Gini)). Par ailleurs, la campagne TREC 07 comportait une tâche de détection d'opinion dans les blogs, proche de celle étudiée ici. Un tour d'horizon de méthodes de détection d'opinion employées pour la tâche *blog track* de la campagne TREC 07 est fourni dans (Macdonald *et al.,* 2007).

Notre approche comporte deux parties : dans un premier temps (corpus d'apprentissage), on utilise la répartition des textes dans les différentes classes afin d'extraire des collocations (regroupements de mots) propres à chacune ; puis, dans un second temps, on utilise ces collocations pour améliorer la catégorisation du corpus de test. Ce travail offre de plus la possibilité d'indiquer à l'utilisateur quels sont les regroupements influents dans le choix de l'attribution d'une classe plutôt que d'une autre.

Des travaux similaires ont été proposés par (Wiebe *et al.,* 2001) (extraction de collocations pour la détection d'opinion), (Ferret, 2002) (utilisation pour la segmentation de textes), ou encore (Roche, 2006) (pour l'extraction de connaissances dans les

textes). Nous proposons pour notre part d'étudier l'influence qu'elles peuvent avoir sur la catégorisation de textes, en extrayant pour cela des collocations propres à chacune des classes.

Notre approche peut aussi être comparée à la classification à base de SVM à noyau de type séquence de mots (Gaussier *et al.,* 2003) : celle-ci permet en effet de considérer des n-grammes de mots en tant que composantes des vecteurs. Cependant, cette prise en compte des co-occurrences par les SVM opère de façon implicite. De ce fait, il est pratiquement impossible de visualiser les traits les plus discriminants. À l'inverse, l'agglutination de mots découlant des collocations offre de façon explicite la possibilité d'identifier les chaînes qui contribuent le plus à l'attribution d'une classe plutôt qu'une autre, ce qui peut notamment être intéressant dans un cadre de détection d'opinion. Bien sûr, il ne serait pas impensable de ré-utiliser ensuite ces chaînes en tant que composantes des vecteurs en entrée d'un autre système de classification (SVM par exemple).

La démarche est la suivante : le programme de calcul de collocations est inclus dans un système de catégorisation automatique de documents adapté à DEFT 07 (ce classifieur utilisait auparavant les mots pris séparément pour créer ses modèles et effectuer par la suite une répartition des textes dans les différentes classes) : une fois la classification effectuée, le programme de calcul de collocations (voir section 2) est lancé, proposant une liste de termes qu'il juge être des collocations avec un score associé. Puis, nous avons ajouté à ce système un programme englobant, qui permet d'automatiquement itérer, les meilleures propositions étant ré-injectées dans le système qui les applique (agglutination des termes composant une collocation dans l'ensemble des corpus), recommence la classification, ... De cette façon, le système utilise, à la place de mots isolés, les termes agglutinés (collocations), ainsi que les mots non-inscrits dans une collocation. On a donc pu suivre, au fur et à mesure des itérations, l'évolution des résultats de la classification suivant les propositions faites. De plus, on a obtenu - par agglutination d'agglutinations - des collocations allant au-delà de deux mots (par exemple : *graphisme-particulièrement-soigner, acheter-en-connaissance-de-cause* (les corpus ont été préalablement lemmatisés). L'évaluation des résultats est faite par le calcul du F-score (voir section 3).

Le but n'est pas ici de chercher à obtenir une analyse des textes selon une norme grammaticale (vraie dans l'absolu), mais leur modélisation selon un ensemble d'usages de la langue (propres au corpus considéré), qui ont une influence sur la classification, avec la possibilité pour l'utilisateur de voir quels sont ces éléments influents. Les résultats font d'ailleurs apparaître un ensemble de chaînes spécifiques parlantes pour l'utilisateur, bien que ne possédant pas forcément d'appellation dans la classification linguistique.

Dans un premier temps, nous définirons les collocations et la méthode utilisée pour les extraire, puis nous étudierons l'impact de ces collocations sur les résultats de notre système de catégorisation et présenterons des exemples de chaînes spécifiques à chaque classe, enfin nous énoncerons les problématiques restantes et les ouvertures possibles.

2. Collocations

Calculer des collocations consiste à trouver des mots qui "vont ensemble" (mots qu'il est naturel de trouver proches dans le langage, ces mots pouvant être contigus ou non). Il existe un certain nombre de méthodes pour trouver des collocations (voir par exemple (Yu *et al.,* 2003) et (Smadja *et al.,* 1990) pour des méthodes numériques, ou encore (Seretan *et al.,* 2004) pour une méthode utilisant des filtres syntaxiques appliqués au corpus du Web), la plus simple étant celle qui retourne les bigrammes les plus fréquents[1]. Nous allons présenter ici une méthode numérique se basant sur le rapport de vraisemblance.

De même, plusieurs méthodes ont déjà été proposées pour tenter d'évaluer les collocations (notamment des méthodes utilisant des dictionnaires dans (Thanopoulos *et al.,* 2002) ainsi que (Pearce, 2002) qui propose d'ailleurs une discussion sur la façon d'évaluer des propositions de collocations).

Nous allons expliquer pourquoi nous pensons que le fait de considérer ensuite ces mots comme une seule entité permet d'améliorer les performances d'un système de classification. Tout d'abord pour augmenter la significativité du terme : par exemple, si on arrive à repérer l'expression *effet particulièrement désagréable* dans un texte, on pourra supposer que la critique est négative, alors qu'un système classique aurait pris les mots séparément et aurait pu juger par exemple que :

– *effet* fait pencher vers une critique positive (comme dans l'expression "majestueux effets" ou "ce jeu fait bon effet") ;

– *particulièrement* vers une classe indéterminée ;

– *désagréable* vers une critique négative.

Ainsi, en considérant l'expression dans son intégralité nous pensons augmenter son pouvoir discriminant. En fait, l'agglutination de mots composants une collocation peut même intégralement remplacer l'utilisation d'un modèle n-grammes. Il s'agit alors d'un modèle unigramme dont les unités de base sont des n-grammes avec n variable. Le problème posé par le regroupement de termes est de savoir où s'arrêter (nombre de termes agglutinés), car créer des regroupements trop grands entraîne des problèmes de couverture (faible probabilité d'apparition de ces regroupements). La seconde raison qui nous pousse à penser que l'on peut améliorer les résultats vient du fait que l'on s'est ici servi de la répartition des textes dans les catégories, il est donc par exemple envisageable de créer des collocations propres à une catégorie.

Pour extraire les collocations présentes dans le corpus d'apprentissage, nous nous sommes appuyés sur la méthode du Rapport de Vraisemblance (*likelihood ratio*), car elle se distingue des autres par son efficacité selon (Daille, 1996) et (Dunning, 1993).

---

1. cette méthode ayant une tendance à plutôt proposer les combinaisons de mots-outils (Manning *et al.,* 2000)

Cette méthode permet d'évaluer la vraisemblance d'une hypothèse par rapport à une autre, les deux hypothèses étant ici :

– les occurrences du mot m_1 sont indépendantes de celles du mot m_2 ;

– les occurrences du mot m_1 sont dépendantes de celles du mot m_2 - cas de collocation.

Le logarithme de ce rapport (LRV) se calcule ainsi (développement de la formule exposée dans (Manning *et al.,* 2000)) :

$$
\begin{aligned}
log\Lambda = 2 \times ([C_{12} * \log \frac{C_{12}}{C_1} + (C_1 - C_{12}) \times \log (1 - \frac{C_{12}}{C_1})] \\
+[(C_2 - C_{12}) * \log \frac{C_2 - C_{12}}{N - C_1} + ((N - C_1) - (C_2 - C_1)) \times \log (1 - \frac{C_2 - C_{12}}{N - C_1})] \\
- [C_{12} * \log \frac{C_2}{N} + (C_1 - C_{12}) \times \log (1 - \frac{C_2}{N})] \\
- [(C_2 - C_{12}) * \log \frac{C_2}{N} + ((N - C_1) - (C_2 - C_{12})) \times \log (1 - \frac{C_2}{N})])
\end{aligned}
$$

[1]

avec :
C_1 le nombre d'occurrences de m_1 dans le corpus ;
C_2 le nombre d'occurrences de m_2 dans le corpus ;
C_{12} le nombre d'occurrences du bigramme $m_1 m_2$ dans le corpus ;
N le nombre de mots du corpus.

Ce logarithme fait correspondre un score à un couple de mots, score qui nous permettra par la suite de juger si ce couple peut être considéré comme une collocation significative.

Nous avons choisi, puisque l'on travaille sur une tâche de catégorisation, d'utiliser cette répartition des textes pour améliorer nos collocations, en extrayant des collocations propres à chacune des catégories. C'est-à-dire que le programme est lancé plusieurs fois (trois dans notre cas), en ne considérant pour chaque exécution que les textes du corpus d'apprentissage appartenant à une des catégories. Grâce à cela, nous obtenons trois sous-listes de collocations, spécifiques à chaque catégorie. Ces sous-listes seront regroupées en une seule lors de l'utilisation (agglutination des collocations possibles dans l'ensemble des corpus).

3. Expériences

Des exemples de collocations obtenues sont fournis dans le tableau 1. Il s'agit des collocations proposées après une itération du programme : des collocations sont calculées sur le corpus d'apprentissage, celles obtenant les meilleurs scores pouvant alors être utilisées comme pré-traitement du texte à la prochaine itération pour améliorer la catégorisation. On agglutinera alors les mots composants les collocations retenues, à la fois dans le corpus d'apprentissage et dans celui de test. Ceci nous permettra éventuellement, au fil des itérations, d'agglutiner des mots à des collocations déjà repérées, et ainsi d'obtenir des chaînes agglutinées de plus de deux mots. La figure 1 présente un schéma général du système.

Après expériences, on a choisi empiriquement de ré-injecter pour chaque classe toutes les propositions de collocations ayant un score supérieur à 200 et apparaissant plus de 10 fois dans le corpus d'apprentissage. Comme le montre le tableau 2, on a assez vite épuisé toutes les propositions que le système pouvait retourner (pas d'ajout de nouvelles collocations au-delà de la sixième itération).

classe 0			**classe 1**			**classe 2**		
score	**mot1**	**mot2**	**score**	**mot1**	**mot2**	**score**	**mot1**	**mot2**
6769	note	général	15674	note	général	12029	note	général
4537	bande	son	10717	bande	son	7733	bande	son
2490	de	vie	5777	de	vie	4101	de	vie
2291	durée	de	5151	il	faillir	4083	il	faillir
2076	ne-est	pas	5015	durée	de	3867	de	de

Tableau 1. *Les 5 premières propositions retournées après une itération par la méthode LRV pour les classe 0 (mauvaise note), 1 (note neutre), 2 (bonne note).*

Les exemples visibles dans le tableau 1 correspondent à des collocations propres au domaine considéré et que l'on peut retrouver dans les différents textes, quelle que soit l'opinion qu'ils expriment (commentaires sur la note attribuée, la bande-son du jeu, etc.). Ce sont celles qui ont obtenu le meilleur score. Mais, au-delà des premiers résultats, on pourra observer des rassemblements qui caractérisent les catégories. Ainsi, pour la catégorie 2 (bonne note), on trouve en 21^e position la collocation *très bon* (score de 1786) qui possède encore un bon score, mais est absente de la liste de la catégorie 1 et a un très mauvais score pour la catégorie 0 (82) ; à l'inverse, une expression comme *manquer cruellement* a un score de 617 pour la catégorie 0 et un score de 192 pour la catégorie 2.

L'évaluation de la catégorisation se fait par calcul du F-score (avec $\beta = 1$), selon la formule employée pour le classement des participations au défi DEFT 07 (voir formule 2), c'est-à-dire en utilisant une moyenne non-pondérée des scores de précision (formule 3) et rappel (formule 4) obtenus pour chacune des *n* catégories.

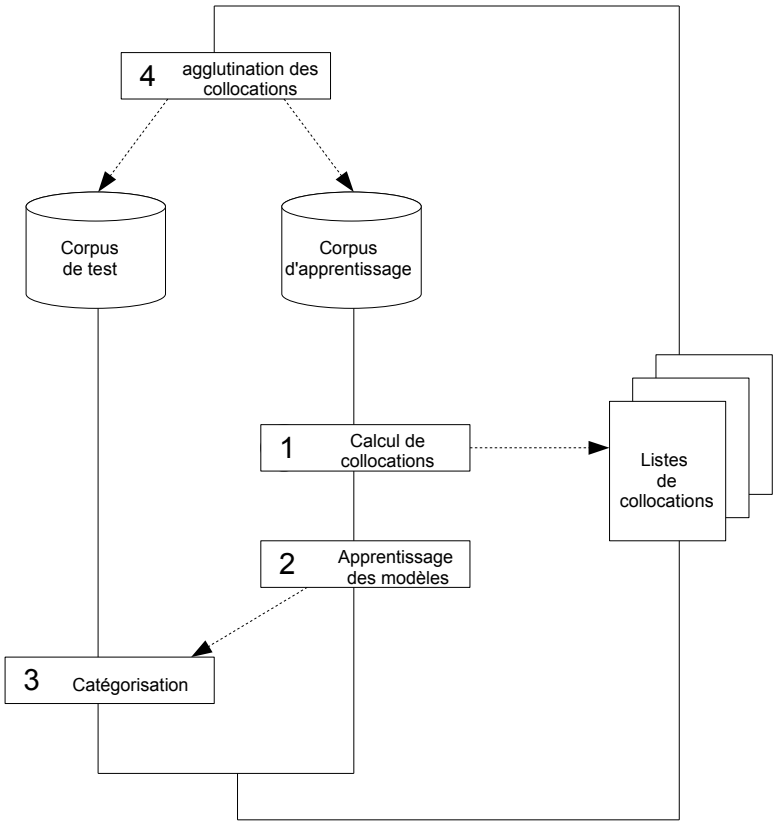


Figure 1. *Fonctionnement du système. Dans l'ordre, à chaque itération : 1) on extrait du corpus d'apprentissage une liste de collocations par catégorie (trois dans notre exemple) ; 2) on entraîne le classifieur sur le corpus d'apprentissage ; 3) on effectue la catégorisation du corpus d'apprentissage (+ calcul du F-score) ; 4) on utilise l'ensemble des listes de collocations réunies en une seule pour agglutiner dans l'ensemble des corpus les termes correspondants.*

$$F_{score}(\beta) = \frac{(\beta^2 + 1) \times Precision \times Rappel}{\beta^2 \times Precision + Rappel} \quad [2]$$

$$Precision = \frac{\sum_{i=1}^{n} Precision_i}{n} \quad [3]$$

$$Rappel = \frac{\sum_{i=1}^{n} Rappel_i}{n} \quad [4]$$

Ainsi, chaque catégorie compte à égalité, ce qui permet d'éviter qu'une catégorie plus peuplée qu'une autre ait un impact plus important sur les résultats.

itération	**modèles**	**F-score**
1	0	0,7530
2	1506	0,7653
3	2101	0,7610
4	2218	0,7672
5	2232	0,7683
6	2234	0,7688
7	2234	0,7688

Tableau 2. *Nombre de modèles de collocations utilisées pour le pré-traitement*

Les résultats de la catégorisation, fournis par la figure 2, sont assez intéressants : on voit que les résultats sont stables dès la 6^e itération (plus aucun ajout de collocations à partir d'ici) et on obtient un meilleur F-score qu'en n'utilisant pas de collocations (correspondant à l'itération 1). Cette méthode a ainsi le triple avantage de :

– converger rapidement ;

– se stabiliser au niveau du meilleur résultat (pas d'oscillations) ;

– fournir de meilleurs résultats qu'une classification sur des mots isolés (le gain observé de 1,5% est en effet important si on considère que notre classifieur obtenait déjà des résultats relativement élevés (voir section 4.1)).

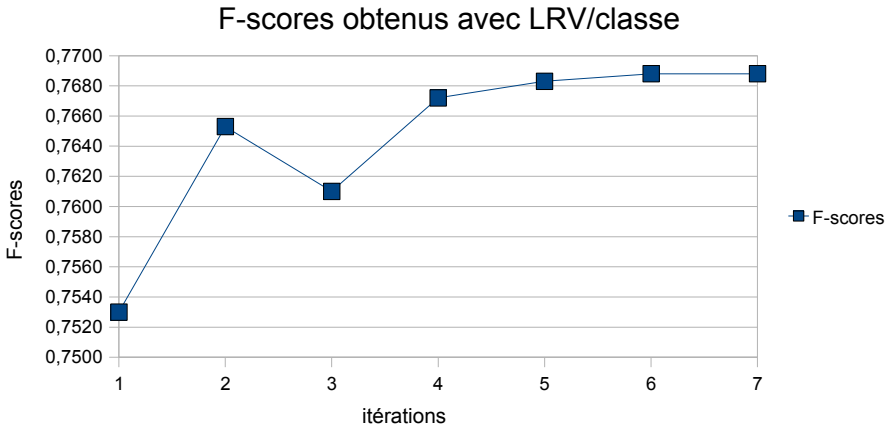


Figure 2. *Ce graphe indique les F-Scores obtenus selon l'itération. À chaque itération on ré-injecte les meilleures propositions de collocations calculées avec le LRV pour chaque catégorie.*

4. Résultats

4.1. *Résultats comparatifs*

Le tableau 3 permet de situer les résultats obtenus par rapport à d'autres méthodes.

système	**F-score**
LIA_cosine	0,7530
LIA_cosine + LRV	0,7634
LIA_cosine + LRV par classe	**0,7688**
LIA (vainqueur Deft)	0,7840
LGI2P (2^e à Deft)	0,7830
moyenne Deft	0,6638
LIA_cosine + manuel	**0,7882**
SVM	0,7410

Tableau 3. *Résultats obtenus par divers systèmes sur la tâche jeuxvideo de DEFT 07. Notre système a obtenu un F-score de 0,7688 - le meilleur score obtenu (sélection manuelle de collocations propres aux différentes catégories) est de 0,7882*

Les systèmes présentés sont :

– LIA_cosine : il s'agit de la base de notre classifieur (évoqué en introduction), utilisé sans applications de collocations ;

– LIA_cosine + LRV : le système précédent, prenant cette fois en compte des collocations calculées sur l'ensemble du corpus d'apprentissage, sans distinction de classe - cette méthode obtient de moins bons résultats que des collocations calculées par catégorie, mais possède l'avantage de pouvoir être entraînée sur l'intégralité du corpus (apprentissage + test), puisqu'on ne considère pas la catégorie pour le calcul de ce que peut être une collocation (voir section 5.2) ;

– LIA_cosine + LRV par classe : notre méthode : prise en compte de l'ensemble des collocations calculées séparément pour chacune des classes ;

– LIA (vainqueur Deft) : la participation du LIA au défi DEFT 07 (Torres-Moreno *et al.,* 2007) : fusion de 9 systèmes de classification (dont *LIA_cosine*, un système de boosting (BoosTexter), des machines à vecteurs de supports, un algorithme des k plus proches voisins, des arbres de classification sémantique, un modèle de probabilités n-gramme, ...) - il s'agit de l'équipe ayant remporté le défi[2] ;

– LGI2P (2^e à Deft) : la participation du LGI2P+LIRMM au défi DEFT 07 (Plantié *et al.,* 2007) : pré-traitements (lemmatisation, anti-dictionnaire, réduction par in-

2. Nous n'avons, pour l'instant, testé notre méthode que sur un seul des classifieurs (*LIA_cosine*) que nous avons utilisés lors de DEFT 07. Il serait intéressant d'observer le gain possible si nous utilisions nos chaînes agglutinées pour l'ensemble des neuf systèmes. De même, nous n'avons effectué nos tests que sur le corpus jeuxvideo de DEFT 07 : nous pourrions réaliser des tests sur les trois autres corpus fournis pour le défi.

formation mutuelle) et catégorisation par SVM - il s'agit de l'équipe ayant terminé deuxième du défi sur cette tâche ;

– moyenne Deft : moyenne de l'ensemble des participations à DEFT 07 pour la tâche portant sur les critiques de jeux vidéos ;

– LIA_cosine + manuel : sélection manuelle des collocations sur 50 itérations ;

– SVM : catégorisation par Machines à Vecteurs de Supports (bilbiothèque LIBLINEAR (Fan *et al.,* 2008)) entraînées spécifiquement pour cette tâche (noyau linéaire, antidictionnaire[3] de 726 entrées, les poids sont les TF.IDF normalisés (compris dans l'intervalle [0 ;1]), les paramètres *c=16* et *e=0,03125* ont été calculés sur le corpus d'apprentissage en validation croisée (5-folds) par tests de valeurs exponentielles successives *c*=2^{-5}, 2^{-4}, ..., 2^{6} et *e*=2^{-6}, 2^{-5}, ..., 2^{3}), sans prise en compte de collocations.

Les résultats *LIA (vainqueur Deft)*, *LGI2P (2^{e} à Deft)* et *moyenne Deft* sont ceux obtenus officiellement au défi DEFT 07 par les équipes participantes (Paroubek *et al.,* 2007) ; les autres correspondent à des expériences que nous avons réalisées dans le cadre de ce travail.

Afin d'évaluer la significativité des résultats, nous avons découpé le corpus de test en 10 sous-ensembles de taille identique (9 sous-ensembles de 170 critiques et le dernier de 164) et effectué le test de catégorisation - sans collocations ou avec la liste de 2234 règles d'agglutination obtenue à l'itération 6 de la méthode *LIA_cosine + LRV par classe* - sur les sous-ensembles pris séparément. La catégorisation avec collocations a été meilleure pour 8 des sous-ensembles. Les intervalles de confiance à 95% de ces deux tests sont [0,7119 ; 0,7742] sans prise en compte de collocations et [0,7378 ; 0,7826] avec. Les résultats se distinguent donc de façon significative, même si l'amélioration reste faible.

On observe ainsi que si le gain que nous avons obtenu pouvait sembler dans un premier temps quelque peu léger, cela provient notamment du fait que l'on parte avec un système qui donne déjà d'assez bons résultats (F-Score de 0,7530 alors que la moyenne des participations se situe à 0,6638 et les deux premiers à 0,7840 et 0,7830. Afin de situer notre système par rapport à un classifieur se basant sur les mots isolés, nous avons effectué des tests avec des Machine à Vecteurs de Support qui obtiennent un résultat de 0,7410. Enfin, la ligne intitulée "LIA_cosine + manuel" correspond à un travail de sélection manuelle des collocations sur 50 itérations. Cette sélection "propre" permet de nous représenter un objectif qu'il doit pouvoir être possible d'atteindre avec des méthodes numériques (gains en temps de travail, d'adaptation à un corpus différent, ...).

3. http ://www.up.univ-mrs.fr/~veronis/data/antidico.txt

4.2. *Aspects qualitatifs*

Au-delà du gain obtenu sur la catégorisation, l'autre avantage de notre méthode est la possibilité de présenter à l'utilisateur les chaînes (agglutinations de collocations) utiles pour la répartition dans les classes (i.e. les chaînes qui, présentes dans un texte, influenceront l'attribution d'une classe plutôt que d'une autre). Le tableau 4 présente quelques-uns des exemples retournés.

chaîne	**classe correspondante**
de-nombreux-défaut	0
tout-ce-que-il-y-de-plus-classique	0
aspect-répétitif	0
aucune-originalité	0
la-désagréable-impression	0
cumuler-défaut	0
beaucoup-trop-limiter	0
décevoir-un-peu	1
rien-de-bien-transcendant	1
bénéficier-d'-un-soin-tout-particulier	2
tenir-en-haleine	2
exempt-de-défaut	2
graphisme-particulièrement-soigner	2

Tableau 4. *Exemples de chaînes spécifiques à chacune des classes.*

On observe ainsi que *de-nombreux-défaut*[4] est une chaîne caractéristique des textes ayant obtenu une mauvaise note (catégorie 0). C'est-à-dire qu'une critique comportant cet ensemble sera très probablement mauvaise. Il en va de même pour les expressions *aucune-originalité*, *cumuler-défaut* ou encore *la-désagréable-impression*. A l'inverse, les expressions *exempt-de-défaut* ou *graphisme-particulièrement-soigner* seront caractéristiques d'une critique positive (catégorie 2).

Cet aspect est particulièrement intéressant dans un contexte applicatif : par exemple extraire, comme ici dans le cas de critiques de produits sur des sites, les points-clés sur lesquels s'expriment les clients, les défauts les plus souvent formulés, ... Par ailleurs, ce travail peut intéresser certaines entreprises dans le cadre de leur Gestion de la Relation Client. Par exemple, pour le traitement de grandes quantités de réponses à des enquêtes de satisfaction ou des remarques d'usagers : notre méthode pourrait leur permettre de détecter l'opinion exprimée par le client (catégorisation automatique de textes) et d'extraire les remarques "pertinentes" (les chaînes spécifiques à chaque classe).

4. les corpus ont été lemmatisés

5. Perspectives

5.1. *Choix des collocations à appliquer*

Un des grands problèmes dans l'utilisation des collocations est de savoir lesquelles retenir : certaines peuvent en effet se recouper, toutes n'ayant pas la même influence sur la classification finale. Actuellement, notre algorithme applique les règles dans l'ordre dans lequel il les rencontre (parcours gauche-droite de la phrase). De plus, on ne traite que des règles agglutinant les éléments deux à deux (et pas directement trois à trois par exemple). Il serait utile de mettre en œuvre une véritable stratégie d'application des règles. En effet, chercher à agglutiner la plus grande chaîne possible (première idée instinctive) n'est pas forcément une bonne idée. Ainsi, si on rencontre la suite de termes *m1 m2 m3 m5*, peut-être qu'agglutiner seulement *m3-m5* d'un côté et *m1-m2* de l'autre offre des meilleurs résultats que si on regroupait directement *m1-m2-m3-m5*. Il faudrait, pour optimiser les choix, disposer d'une information supplémentaire (chiffrage gain/perte) permettant de préférer en contexte une agglutination à une autre.

Voici deux exemples de problèmes soulevés par notre méthode actuelle, en supposant que l'on ait les règles d'agglutination proposées par le tableau 5.

règle	**T1**	**T2**	**agglutination (T1-T2)**
R1	m1-m2	m3-m4	m1-m2-m3-m4
R2	m1-m2	m3	m1-m2-m3
R3	m3	m5	m3-m5
R4	m2	m3	m2-m3
R5	m1	m2	m1-m2
R6	m3	m4	m3-m4

Tableau 5. *Exemples de règles d'agglutination de collocations, par exemple R6 signifie que si l'on rencontre les mots m3 et m4 côte à côte dans le texte on pourra les agglutiner pour former m3-m4.*

– problème 1 : on n'agglutine que deux termes à chaque passe, ainsi, selon l'ordre dans lequel les règles sont rencontrées, on ne pourra pas créer toutes les agglutinations, par exemple : on a la chaîne *m1 m2 m3 m4* : si, à la première passe, on a agglutiné m2 et m3 (R4), jamais on ne pourra appliquer la règle R1 ;

– problème 2 : cas de recoupement : si pour un même ensemble il est possible de créer plusieurs agglutinations différentes, comment savoir laquelle est la plus intéressante, par exemple : on a la chaîne *m1 m2 m3 m5* : comment choisir entre appliquer R3 *m1 m2 m3-m5* ou appliquer successivement R5 *m1-m2 m3 m5* et R2 *m1-m2-m3 m5* ?

Pratiquement, un exemple de problème qui pourrait se poser avec le corpus présenté serait celui provoqué par l'application des règles exposées dans le tableau 6.

règle	**T1**	**T2**	**agglutination (T1-T2)**
R7	très-utile	pour-mener-bien	très-utile-pour-mener-bien
R8	pour	mener	pour-mener
R9	très	utile	très-utile
R10	très-utile	pour	très-utile-pour
R11	pour	mener-bien	pour-mener-bien
R12	mener	bien	mener-bien

Tableau 6. *Exemples de règles d'agglutination de collocations du corpus jeuxvideo de DEFT 07.*

Ainsi, lors des extractions de collocations, le système a jugé que *pour mener* correspondait à une collocation possible, de même que *mener bien*, puis, au fil des itérations, la chaîne complète *très utile pour mener bien*. Cependant, selon si, au moment d'agglutiner ces collocations dans le corpus, on applique en premier la règle R8 ou le couple R12-R9, on n'obtiendra pas le même résultat. Dans le premier cas, on arrivera au mieux à segmenter ainsi : *très-utile pour-mener bien*. Dans le second cas, on pourra éventuellement arriver jusqu'à agglutiner la chaîne complète *très-utile-pour-mener-bien*. On voit ici les limites d'une méthode qui ne s'appuie pas sur une modélisation de la structure syntagmatique des phrases. Sachant les problèmes posés par une approche syntaxique (robustesse, manque de couverture, temps d'exécution), nous avons délibérément opté pour une analyse probabiliste.

5.2. *Autres perspectives*

En relation avec les problèmes exposés à la section 5.1, nous cherchons aussi une méthode pour calculer des collocations de plus de deux termes, de façon directe et non incrémentale comme nous le faisons actuellement. De plus, nous aimerions avoir un moyen de trouver des collocations "à trous" (termes non-obligatoirement consécutifs).

Par ailleurs, nous aimerions introduire les mêmes mécanismes qui amènent un être humain à lire un texte de façon différente suivant s'il le comprend d'une manière ou d'une autre. Regarder de bout en bout le texte traité sous un angle différent selon chacune des opinions envisagées, pourrait se traduire pour nous par l'emploi d'autant de jeux d'agglutinations qu'il y a de catégories. C'est-à-dire que, pour chaque texte, on effectuerait trois tests de catégorisation, en n'agglutinant pour chacun que les chaînes spécifiques à une des trois catégories. Puis, au regard de ces trois tests, on attribuerait à ce texte la catégorie ayant obtenu les meilleurs résultats. L'idée sous-jacente est que la manière dont les termes sont regroupés (et donc la phrase segmentée) peut faire varier totalement la catégorisation proposée par le système. Ainsi, la phrase *J'apprécie fort peu l'histoire*, fera probablement pencher la catégorisation vers une critique plutôt négative si elle est découpé comme suit : *J'apprécie-fort-peu l'histoire* et plutôt positive en suivant cet autre découpage : *J'apprécie-fort peu l'histoire*. Ces cas ne sont

pas improbables si on considère que nos corpus sont extraits de critiques laissées par des internautes (langage non-forcément académique), lemmatisés et sans ponctuation.

Pour comparaison, il nous semble intéressant, en utilisant la méthode du rapport de vraisemblance globale (collocations apprises sur les textes de toutes les classes confondues), d'extraire des collocations sur l'intégralité du corpus (apprentissage + test), ce qui augmenterait le nombre d'exemples sans toutefois introduire de biais, puisque l'on n'utilise pas la répartition dans les classes.

6. Conclusion

Si l'apport de la prise en compte des collocations dans la catégorisation des textes (orientée ici détection d'opinion) peut sembler avoir un impact faible au niveau du F-score (gain d'un peu plus de 1,5%), ces résultats sont à relativiser par rapport au score élevé qu'obtenait déjà le classifieur sur lequel nous nous sommes basés. De plus, il ne faut pas voir ici uniquement l'influence sur les résultats de la classification, mais aussi l'apport pratique pour l'utilisateur, c'est-à-dire la possibilité de lui montrer les chaînes spécifiques à chacune des classes. Si ces chaînes sont des agglutinations, le sens qu'elles sous-tendent peut être plus facilement perceptible que s'il s'agissait de termes simples (cas des chaînes telles que *rien-de-bien-transcendant* ou *exempt-de-défaut*). Ceci est particulièrement intéressant dans les cas de détection d'opinion, puisque cela nous permet par exemple de visualiser les reproches les plus souvent formulés. Enfin, il s'agit d'une méthode nouvelle, pour laquelle de nombreuses améliorations sont envisageables.

7. Bibliographie

Daille B., « Study and Implementation of Combined Techniques for Automatic Extraction of Terminology », *The Balancing Act : Combining Symbolic and Statistical Approaches to Language*, vol. 1, p. 49-66, 1996.

Dunning T., « Accurate methods for the statistics of surprise and coincidence », *Computational Linguistics*, vol. 19, p. 61-74, 1993.

Fan R.-E., Chang K.-W., Hsieh C.-J., Wang X.-R., Lin C.-J., « LIBLINEAR : A Library for Large Linear Classification », *The Journal of Machine Learning Research*, vol. 9, p. 1871-1874, 2008.

Ferret O., « Using collocations for topic segmentation and link detection », *Proceedings of the 19th international conference on Computational linguistics*, Taipei, Taiwan, p. 1-7, 2002.

Gaussier N. C. E., Goutte C., Renders J. M., « Word sequence kernels », *The Journal of Machine Learning Research*, vol. 3, p. 1059-1082, 2003.

Grouin C., Berthelin J.-B., Ayari S. E., Heitz T., Hurault-Plantet M., Jardino M., Khalis Z., Lastes M., « Présentation de DEFT 07 (DEfi Fouille de Textes) », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 1-8, 2007.

Macdonald C., Ounis I., Soboroff I., « Overview of the TREC 2007 Blog Track », *Proceedings of TREC 2007*, Gaithersburg, USA, 2007.

Manning C. D., Schütze H., *Foundations of statistical natural language processing*, MIT Press, p. 151-189, 2000.

Paroubek P., Berthelin J.-B., Ayari S. E., Grouin C., Heitz T., Hurault-Plantet M., Jardino M., Khalis Z., Lastes M., « Résultats de l'édition 2007 du DEfi Fouille de Textes », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 9-17, 2007.

Pearce D., « A Comparative Evaluation of Collocation Extraction Techniques », *Conference on language Resources and Evaluation*, p. 1530-1536, 2002.

Plantié M., Dray G., Roche M., « Défi DEFT07 : Comparaison d'approches pour la classification de textes d'opinion », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 57-69, 2007.

Roche M., « Acquisition de la terminologie et définition des tâches à effectuer, deux principes indissociables », *Actes des Journées de Rochebrune (Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels)*, p. 151-161, 2006.

Salton G., Buckley C., « Term weighting approaches in automatic text retrieval », *Information Processing and Management*, vol. 24(5), p. 513-523, 1988.

Seretan V., Nerima L., Wehrli E., « Using the Web as a corpus for the syntactic-based collocation identification », *Proceedings of International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, p. 1871-1874, May, 2004.

Smadja F. A., McKeown K. R., « Automatically extracting and representing collocations for language generation », *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, p. 252-259, 1990.

Thanopoulos A., Fakotakis N., Kokkinakis G., « Comparative Evaluation of Collocation Extraction Metrics », *The 3rd International Conference on Language Resource and Evaluation*, p. 620-625, 2002.

Torres-Moreno J.-M., El-Bèze M., Béchet F., Camelin N., « Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? Application au défi DEFT 2007 », *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, Grenoble, France, p. 119-133, 2007.

Wiebe J., Wilson T., Bell M., « Identifying Collocations for Recognizing Opinions », *Proceedings of the ACL/EACL Workshop on Collocation*, Toulouse, France, 2001.

Yang Y., Liu X., « A re-examination of text categorization methods », *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, Berkeley, California, USA, p. 42-49, 1999.

Yu J., Jin Z., Wen Z., « Automatic Detection of Collocation », *The 4th Chinese lexical semantics workshop*, Hong-Cong, 2003.

Rôle de la matrice d'information et pondération des composantes dans les noyaux de Fisher pour PLSI[†]

Jean-Cédric Chappelier — Emmanuel Eckard

Laboratoire d'Intelligence Artificielle
École Polytechnique Fédérale de Lausanne
CH–1015 Lausanne
{jean-cedric.chappelier,emmanuel.eckard}@epfl.ch

RÉSUMÉ. *Des similarités entre documents à base de catégories sémantiques latentes et de noyaux de Fisher ont été proposées pour la première fois il y a dix ans par T. Hofmann dans le contexte du "Probabilistic Latent Semantic Indexing", puis étendues par Nyffenegger et al. (2006). Le présent article présente une étude approfondie et une révision de ces modèles par (1) une description unifiée et simplifiée, (2) une étude du rôle de la matrice d'information de Fisher $G(\theta)$, et (3) une analyse de l'impact des paramètres associés aux catégories latentes. Il fournit de plus de nouveaux résultats expérimentaux sur une grande collection de document provenant du corpus d'évaluation TREC–AP.*

ABSTRACT. *An information-geometric approach for document similarities in the framework of "Probabilistic Latent Semantic Indexing" was first proposed by T. Hofmann (2000) and later extended ("revisited") by Nyffenegger et al. (2006). This paper presents an in-depth study and revision of these models by (1) providing a simpler unified description framework, (2) investigating the role of the Fisher Information Matrix $G(\theta)$, and (3) analyzing the impact of latent "topic" parameters in such models. It furthermore provides new experimental results on larger collections coming from the TREC–AP evaluation corpus.*

MOTS-CLÉS : *Recherche d'information, classification textuelle, représentation de documents, noyau de Fisher, PLSI.*

KEYWORDS: *Information Retrieval, Document Classification, document representation, Fisher kernel, PLSI.*

[†] Ce travail a été financé dans le cadre des projets 200021–111817 et 200020–119745 du Fond National Suisse.

1. Introduction

L'indexation automatique de grandes collections de documents en vue de leur analyse, de leur organisation ou de leur fouille demeure l'un des grands sujets de l'intelligence artificielle moderne. Dans ce contexte, la représentation des documents comme mélanges de catégories latentes s'est avérée prometteuse. Le modèle « Probabilistic Latent Semantic Indexing » (PLSI) (Hofmann, 1999, Hofmann, 2000, Hofmann, 2001) à base de représentations latentes probabilistes a conduit à diverses extensions et applications sur des données textuelles (Vinokourov *et al.,* 2002, Gaussier *et al.,* 2002, Steyvers *et al.,* 2004, Jin *et al.,* 2004, Mei *et al.,* 2006), sonores (Ahrendt *et al.,* 2005), ou graphiques (Monay *et al.,* 2004, Quelhas *et al.,* 2005, Bosch *et al.,* 2006, Monay *et al.,* 2007, Lienhart *et al.,* 2007).

Dans ce contexte, la « similarité cosinus » employée à l'origine, sans justification théorique, pour évaluer la proximité sémantique entre documents, a laissé la place à des similarités à base de noyaux de Fisher, mieux justifiée théoriquement (Hofmann, 2000). Cette approche a ensuite été étendue à plusieurs autres modèles de similarité résumés ci-dessous (Nyffenegger *et al.,* 2006).

Cet article propose une étude approfondie et une révision de ces différents modèles par (1) une description unifiée et simplifiée, (2) une étude du rôle de la matrice d'information de Fisher $G(\theta)$, et (3) une analyse de l'impact des paramètres associés aux catégories latentes. De plus, il fournit de nouveaux résultats expérimentaux sur une grande collection de document provenant du corpus d'évaluation TREC–AP.

La section suivante offre un rappel sur le modèle de documents et les mesures de similarité de PLSI. Le rôle de la matrice d'information de Fisher est ensuite examiné, suivi par une discussion sur les proportions entre les différentes composantes des noyaux. Pour finir, le cadre expérimental est présenté et les principaux résultats sont résumés, avant la conclusion.

2. Modèle de documents et mesures de similarité

2.1. *Le modèle PLSI de documents*

PLSI est un modèle à catégories latentes pour la classification de documents et la recherche d'information (Hofmann, 1999). Il modélise les documents comme des réalisations de tirages aléatoires successifs de couples document–terme (d, w) : itérativement, une catégorie sémantique $z \in Z$ est d'abord choisie, avec une probabilité $P(z)$, puis un terme w et un document d sont choisis avec respectivement des probabilités $P(w|z)$ et $P(d|z)$. Dans PLSI, w et d sont supposés indépendants pour z connu ; la probabilité d'une paire (d, w) s'écrit alors :

$$P(d,w) = \sum_{z \in Z} P(z) \underbrace{P(w|z)P(d|z)}_{d,w \text{ indép. à } z \text{ connu}} .$$

Les paramètres du modèle PLSI sont $\theta = (P(z), P(w|z), P(d|z))$, pour tous les z, w et d du modèle. Ces paramètres s'estiment par l'algorithme Expectation-Maximization (EM) pour une collection de documents C donnée (Hofmann, 1999, Hofmann, 2001).

2.2. *Variantes des noyaux de Fisher pour PLSI*

Rappelons que pour une famille $P(X|\theta)$ de modèles stochastiques paramétrée par θ, le noyau de Fisher fournit une mesure de similarité entre occurrences, lequel, pour deux occurrences X et Y de cette famille au point θ, est défini par

$$K(X,Y) = U_X(\theta)^T \; G(\theta)^{-1} \; U_Y(\theta), \qquad [1]$$

où $U_X(\theta)$ est le gradient par rapport aux paramètres de la log-vraisemblance du modèle : $U_X(\theta) = \nabla_\theta \log P(X|\theta)$, et où la matrice d'information de Fisher $G(\theta)$ est la covariance de $U_X(\theta)$: $G(\theta) = \mathbf{E}_X[U_X(\theta)\; U_X(\theta)^T]$.

Le noyaux de Fisher pour PLSI dérivé par Hofmann (Hofmann, 2000), mesurant la distance entre un document d et une requête q, s'écrit :

$$K^{\mathrm{H}}(d,q) = \sum_z \frac{P(z|d)P(z|q)}{P(z)} + \sum_w \widehat{P}(w|d)\widehat{P}(w|q) \sum_z \frac{P(z|d,w)P(z|q,w)}{P(w|z)},$$

où $\widehat{P}(w|d) = \frac{n(d,w)}{|d|}$, avec $n(d,w)$ le nombre d'occurrences du terme w dans le document d et $|d| = \sum_w n(d,w)$, sa taille (en nombre de termes).

Cette mesure améliore notablement les performances par rapport aux formulations originales qui utilisent la mesure cosinus. Toutefois, la dérivation de K^{H} néglige la contribution de la matrice d'information de Fisher $G(\theta)$,[1] et contient une normalisation par la taille du document $|d|$, dont la justification théorique n'est pas évidente[2] (Nyffenegger *et al.*, 2006). Différentes variantes pour le noyau de Fisher de PLSI sont ainsi introduites :

1) La remarque concernant la renormalisation par la longueur du document conduit au développement d'un noyau de Fisher K^F, non normalisé par $|d|$. Un noyau hybride nommé VS (« *vector space* ») est également proposé, dans lequel la composante K_z^H liée aux catégories sémantiques latentes n'est pas normalisé par $|d|$, mais où la composante K_w^H, liée aux termes, l'est.

2) D'autre part, l'observation concernant la matrice d'information de Fisher conduit au développement d'un noyau dit « *DFIM* » (*Diagonal Fisher Information*

1. Hofmann assimile $G(\theta)$ à la matrice identité par une reparamétrisation justifiée dans le cas des multinomiales ; cependant, PLSI n'est ni une multinomiale, ni dans une famille exponentielle, et $G(\theta)$ peut s'éloigner significativement de la matrice identité dans ce type de cas.
2. Elle pourrait cependant s'expliquer en envisageant le noyau sous l'angle d'un processus stochastique indépendant et identiquement distribué.

Matrix), où les composantes diagonales de la matrice d'information de Fisher $G(\theta)$ sont prises en compte en utilisant l'approximation

$$G(\theta)_{(ii)} \approx \sum_{d \in C} \left(U_d(\theta)_{(i)}\right)^2 . \qquad [2]$$

Par contraste et lorsque c'est nécessaire, on nommera « *IFIM* » (*Identity Fisher Information Matrix*) les noyaux qui ne tiennent pas compte de ces facteurs et font l'hypothèse d'identité pour $G(\theta)$.

Nyffenegger et al. (2006) proposent une étude comparative du noyau de Fisher K^{H}, du noyau VS K^{VS} et de sa version DFIM $K^{\text{DFIM-VS}}$. Le présent article va plus loin en introduisant un certain nombre d'autres variantes : le noyau de Fisher K^{F} ; sa version DFIM, $K^{\text{DFIM-F}}$, qui prend en compte les termes diagonaux de $G(\theta)$ dans le contexte de K^{F} ; et $K^{\text{DFIM-H}}$, la version construite de la même façon à partir de K^{H}. Il étudie aussi indépendamment les composantes liées aux termes et celles liées aux catégories latentes de chacun de ces noyaux.

3. Normalisations des différentes variantes du noyau

En appliquant l'équation [1] au modèle PLSI, on constate que tous les noyaux sont composés de deux termes additifs : l'un reflète la contribution des catégories sémantiques latentes $z \in Z$, et l'autre, la contribution des termes w :[3]

$$K(d,q) = K_z(d,q) + K_w(d,q) = \sum_z k_z(z,d,q) + \sum_w k_w(w,d,q) \qquad [3]$$
$$= \sum_z \frac{U_d(z)\,U_q(z)}{\alpha(z)} + \sum_w \sum_z \frac{U_d(w|z)\,U_q(w|z)}{\gamma(w,z)},$$

en notant $U_d(z)$ la composante de $U_d(\theta)$ qui correspond à $P(z)$, et de la même façon $U_d(w|z)$ pour $P(w|z)$, et où $\alpha(z)$ et $\gamma(w,z)$ sont soit égaux à 1 (cas IFIM), soit représentent les composantes diagonales de $G(\theta)$, estimées par l'équation [2] sur une collection de documents C :

$$\alpha(z) = \sum_{d \in C} U_d(z)^2, \qquad \gamma(w,z) = \sum_{d \in C} U_d(w|z)^2.$$

Le rôle normalisateur de $G(\theta)$ apparaît ici clairement : premièrement, tous les termes indépendants de d s'annulent dans k_w et k_z puisqu'ils se factorisent dans $\alpha(z)$ et $\gamma(w,z)$; deuxièmement, k_z et k_w ont une forme proche de $\frac{a \cdot b}{a^2+b^2+c^2}$, laquelle est majorée par 0.5. Cette forme est exacte lorsque d et q sont tous deux compris dans C ; on a alors, par exemple pour k_z, $a = U_d(z)$, $b = U_q(z)$ et $c = \sum_{\delta \neq d,q} U_\delta(z)$.

3. Les contributions des paramètres $P(d|z)$ se simplifient, puisque pour deux documents d'indices d_1 et d_2 différents, les vecteurs $U_{d_1}(\{P(d|z)\})$ et $U_{d_2}(\{P(d|z)\})$ sont orthogonaux.

Les expression détaillées de k_z et k_w pour les différents noyaux étudiés ici sont données dans le tableau 1.

$k_z(z,d,q)$	IFIM ($G(\theta)=I$)	DFIM
non normalisé (F)	$k_z^{\mathrm{F}}(z,d,q) = \frac{P(d,z)P(q,z)}{P(z)}$	$k_z^{\mathrm{F}}(z,d,q) \cdot \underbrace{\left(\sum_\delta \frac{P^2(\delta,z)}{P(z)}\right)^{-1}}_{\alpha^{\mathrm{F}}(z)^{-1}}$
Hofmann (H)	$k_z^{\mathrm{H}}(z,d,q) = \frac{P(z\|d)P(z\|q)}{P(z)} = \frac{k_z^{\mathrm{F}}(z,d,q)}{P(d)\,P(q)}$	$k_z^{\mathrm{H}}(z,d,q) \cdot \underbrace{\left(\sum_\delta \frac{P^2(z\|\delta)}{P(z)}\right)^{-1}}_{\alpha^{\mathrm{F}}(z)^{-1}}$
« *Vector space* » (VS)	$k_z^{\mathrm{VS}} = k_z^{\mathrm{F}}$	$k_z^{\text{DFIM-VS}} = k_z^{\text{DFIM-F}}$

$k_w(w,d,q)$ toutes versions

$$\widehat{P}(d,w)\widehat{P}(q,w)\sum_z \frac{P(d|z)P(q|z)}{P(d,w)P(q,w)} \begin{cases} P^2(z)P(w|z) & \leftarrow k_w^{\mathrm{F}} \\ \left[\sum_\delta \widehat{P}^2(\delta,w)\frac{P^2(\delta|z)}{P^2(\delta,w)}\right]^{-1} & \leftarrow k_w^{\text{DFIM-F}} \end{cases}$$

$$\widehat{P}(w|d)\widehat{P}(w|q)\sum_z \frac{P(d|z)P(q|z)}{P(d,w)P(q,w)} \begin{cases} P^2(z)P(w|z) & \leftarrow k_w^{\mathrm{H}} \\ \left[\sum_\delta \widehat{P}^2(w|\delta)\frac{P^2(\delta|z)}{P^2(\delta,w)}\right]^{-1} & \leftarrow k_w^{\text{DFIM-H}} \end{cases}$$

$k_w^{\mathrm{VS}} = k_w^{\mathrm{H}}$; $k_w^{\text{DFIM-VS}} = k_w^{\text{DFIM-H}}$

Tableau 1. *Composante catégories $k_z(z,d,q)$ et composante termes $k_w(w,d,q)$ des différents noyaux de Fisher pour PLSI. $\widehat{P}(d,w) = \frac{n(d,w)}{|C|}$. Noter que $k_w^H = \frac{|C|^2}{|d||q|}k_w^F$.*

4. Rapports et proportions entre composantes des noyaux

La renormalisation par $|d|$ dans l'équation d'Hofmann entraîne les proportions suivantes entre K^{H} et K^{VS}, exprimées en fonction de K^{F} :

	K_z	K_w
non normalisé : K^{F}	K^{F}_z	K^{F}_w
Hofmann (2000) : K^{H}	$\frac{1}{P(d)P(q)} K^{\mathrm{F}}_z$	$\frac{\lvert C\rvert^2}{\lvert d\rvert\lvert q\rvert} K^{\mathrm{F}}_w$
« Vector Space » : K^{VS}	K^{F}_z	$\frac{\lvert C\rvert^2}{\lvert d\rvert\lvert q\rvert} K^{\mathrm{F}}_w$

Compte tenu de l'ordre de grandeur typique de la taille de $|C|$ (nombre total d'occurrences de tous les termes dans la collection de documents) par rapport aux valeurs typiques pour $|d|$ et $|q|$ (i.e. $|C|^2 \gg |d||q|$), il est évident que le noyau VS est fortement dominé par sa composante K_w. Ceci est confirmé expérimentalement, comme montré en section suivante.

En ce qui concerne les noyaux K^{F} et K^{H}, on peut remarquer que la log-vraisemblance non normalisée

$$l^{\mathrm{F}}(d) = \sum_w n(d,w) \log \sum_z P(z)P(w|z)P(d|z)$$

et celle utilisée par Hofmann (2000), l^{H}, sont reliées par $l^{\mathrm{H}}(d) = \frac{1}{|d|} l^{\mathrm{F}}(d)$. Lorsque la log-vraisemblance est multipliée par une constante, le noyau de Fisher résultant reste inchangé (cf Eq. 1). Toutefois, si la log-vraisemblance est multipliée par une fonction de d indépendante des paramètres, c'est-à-dire si $l'_d(\theta) = \lambda(d)\, l_d(\theta)$, avec $\nabla_\theta \lambda = 0$, alors

$$G'(\theta) = \mathbf{E}_d \left[(\nabla_\rho l'_d(\rho)) \, (\nabla_\rho l'_d(\rho))^T \right] = \mathbf{E}_d \left[\lambda(d)^2 (\nabla_\rho l_d(\rho)) \, (\nabla_\rho l_d(\rho))^T \right],$$

et

$$K'_\theta(d,q) = \lambda(d)\, \lambda(q) \left(\nabla_\theta l_d(\theta))^T \, G'(\theta)^{-1} \, (\nabla_\theta l_d(\theta) \right),$$

qui, dans le cas général, ne peut pas s'écrire en termes de $K_\theta(d,q)$.

Cependant, si $G(\theta)$ n'est pas prise en compte (IFIM), les deux noyaux sont alors bien en relation directe :

$$K'_\theta(d,q) = \lambda(d)\, \lambda(q)\, K_\theta(d,q).$$

Ainsi, entre les versions IFIM des noyaux K^{F} et K^{H}, on doit avoir : $K^{\mathrm{H}}(d,q) = \frac{1}{|d|\,|q|} K^{\mathrm{F}}(d,q)$. La raison pour laquelle cette relation n'est pas exactement vérifiée dans les formules du tableau 1 vient du fait que ces formules utilisent des hypothèses différentes dans leurs dérivations : Hofmann (2000) postule que $\sum_w \frac{\widehat{P}(w|d)}{P(w|d)} P(w|z) \approx 1$, là où Nyffenegger et al. (2006) postulent que $\sum_w \frac{\widehat{P}(d,w)}{P(d,w)} P(w|z) \approx 1$.

Remarquons toutefois que

$$\sum_w \frac{\widehat{P}(w|d)}{P(w|d)} P(w|z) = P(d)\,\frac{|C|}{|d|}\,\sum_w \frac{\widehat{P}(d,w)}{P(d,w)} P(w|z)\,.$$

Le passage de la première approximation à la seconde transforme donc $|d|/|C|$ en $P(d)$, ce qui explique les formules obtenues (tableau 1).

Il y aurait ainsi trois versions similaires possibles pour les noyaux d'Hofmann : K^{H}, le noyau initialement calculé et décrit plus haut, mais aussi $K^{\mathrm{H}_1} = \frac{|C|^2}{|d|\,|q|} K^{\mathrm{F}}$, qui a le même k_z que K^{H} mais un k_w différent, et $K^{\mathrm{H}_2} = \frac{1}{P(d)\,P(q)} K^{\mathrm{F}}$, qui a le même k_w que K^{H} mais un autre k_z.

Ces trois noyaux ont été comparés expérimentalement : on ne constate aucune différence dans les résultats finaux (le classement des documents), les différences dans les valeurs des scores de similarité étant de l'ordre de 10^{-4}%. Ceci vient de ce que $|d|/|C|$ est de fait un très bon estimateur de $P(d) = \sum_z P(d|z)P(z)$. Pour des raisons de cohérence avec la littérature existante, nous garderons ici K^{H} plutôt que K^{H_1} ou K^{H_2}.

Nous avons donc en tout 14 noyaux différents qui s'expriment tous sous la forme de l'équation [3] et sont résumés dans le tableau 1 : les trois modèles IFIM K^{F}, K^{H}, et K^{VS} ; leurs versions DFIM ; et les (huit) versions de K_w et K_z réparties entre $K^{(\text{DFIM-})\mathrm{F}}$ et $K^{(\text{DFIM-})\mathrm{H}}$. À noter qu'il n'y a pas de K_w ni K_z spécifiques pour $K^{(\text{DFIM-})\mathrm{VS}}$ puisque par construction $K_z^{(\text{DFIM-})\mathrm{VS}} = K_z^{(\text{DFIM-})\mathrm{F}}$ et $K_w^{(\text{DFIM-})\mathrm{VS}} = K_w^{(\text{DFIM-})\mathrm{H}}$.

5. Expériences

Nous avons évalué ces 14 noyaux sur les bases d'évaluation standard de la recherche d'information CACM, CISI, MED, CRAN et TIME provenant de la collection SMART[4]. Nous les avons de plus évalués sur un corpus nettement plus grand, constitué d'une partie du corpus TREC–AP 89 (Harman, 1995), une collection de nouvelles d'agence de l'Associated Press collectées sur l'année 1989. Pour des raisons pratiques (temps de calcul et taille mémoire), seuls les 7466 premiers documents de la collection et les 50 premières requêtes ont été conservés[5] ; en ce qui concerne les occurrences de termes, cela constitue une base plus de 10 fois plus grande que la plus grosse base de SMART, et près de 5 fois plus grande en terme de nombre de documents. Les caractéristiques principales de ces corpus d'évaluation sont présentées dans le tableau 2.

4. `ftp://ftp.cs.cornell.edu/pub/smart/`
5. Documents AP890101-0001 à AP890131-0311.

	CACM	CRAN	TIME	CISI	MED	AP89_01XX
Nb de termes	4 911	4 063	13 367	5 545	7 688	13 379
$\|C\|$	90 927	120 973	114 850	87 067	76 571	1 321 482
Documents						
Nb	1 587	1 398	425	1 460	1 033	7 466
$\|d\|$ moyen	56.8	85.1	268.6	56.7	73.8	177.2
Requêtes						
Nb	64	225	83	112	30	50
$\|q\|$ moyen	12.7	8.9	8.2	37.7	11.4	79.3

Tableau 2. *Caractéristiques principales des corpus d'évaluation.*

		CACM	CRAN	TIME	CISI	MED	AP89
Résultats	MAP de BM25	**31.4**	**42.4**	**69.2**	12.3	52.3	19.7
	K_w^{H} MAP	30.0	33.6	55.6	**20.2**	49.8	16.5
	$K_w^{\text{DFIM-H}}$ MAP	23.2	37.0	60.8	15.6	45.5	**21.6**
	Meilleure MAP des noyaux PLSI	30.7	37.6	60.8	**20.3**	**53.8**	**21.6**
	Meilleur noyau PLSI, pour $\|Z\| =$	K_w^{F} 16	$K^{\text{DFIM-H}}$ 64	$K_w^{\text{DFIM-H}}$ 8	K^{VS} 8	K^{H} 32	$K_w^{\text{DFIM-H}}$ 48
Conclusions	noyau PLSI $>$ BM25 ?	Non	Non	Non	**OUI**	**oui**	**oui**
	K_z contribue ?	Non	Non	Non	Non	peu	Non
	DFIM $G(\theta)$ contribue ?	Non	**Oui** (K_w)	**Oui** (K_w)	Non	peu	**Oui** (K_w)

Tableau 3. *Principaux résultats et conclusions des 3024 expériences sur les 14 modèles et 6 corpus.*

Pour les expériences sur les collections SMART, six expériences avec des conditions initiales d'apprentissage différentes ont été effectuées pour chacun des modèles, et pour différents nombres de catégories latentes : $|Z| \in \{1, 2, 8, 16, 32, 64, 128\}$, soit un total de 2940 expériences[6]. Pour le corpus TREC-AP, les expériences n'ont été faites que sur la base d'une seule condition initiale mais pour différents $|Z| \in \{1, 32, 48, 64, 80, 128\}$, soit 84 expériences en tout.

Pour toutes ces expériences, le stemming a été effectué à l'aide du stemmer de Porter de Xapian[7]. Les résultats ont été obtenus grâce à l'outil standard `trec_eval`[8]. Nous utilisons ici la mesure *Mean Average Precision* (MAP) pour les présenter, mais les conclusions se sont avérées être exactement les mêmes avec la précision à 5 points (P5) ou la R-précision.
Toutes les figures (excepté la 5) représentent la MAP en fonction de $|Z|$, avec des barres d'erreur verticales correspondant à un écart type.

Les résultats les plus importants de ces 3024 expériences, résumés dans tableau 3, sont :

1) Comme l'illustre par exemple la figure 1, $\{K^{\text{DFIM-VS}}, K_w^{\text{DFIM-H}}, K^{\text{H}}\}$ (resp. $\{K^{\text{DFIM-F}}, K_w^{\text{DFIM-F}}\}$) se comportent de façon semblable. La raison est que le rôle normalisateur de $G(\theta)$ rend $K_z \ll K_w$ pour les noyaux DFIM.

De plus, $K^{\text{VS}} \simeq K_w^{\text{H}}$ puisque $\frac{|C|^2}{|d||q|} \gg 1$, comme mentionné en section 4. Le modèle VS ne vaut donc pas la peine d'être considéré, puisqu'il imite de très près le comportement de la composante $K_w^{\text{(DFIM-)H}}$.

2) Comme l'illustre par exemple la figure 2, K_z détériore les performances, de façon générale : seul, il donne de mauvais résultats ; de plus, au fur et à mesure que son rôle devient plus important dans K^{H} et K^{F}, quand $|Z|$ s'accroît, les performances de ces noyaux se détériorent : partant de K_w pour une valeur faible de $|Z|$, les performances de K^{H} et K^{F} chutent jusqu'à celles de K_z pour $|Z|$ élevé.

3) K_w pris seul offre toujours de bons résultats, si ce n'est les meilleurs résultats.

4) On observe les mêmes effets sur le corpus TREC-AP corpus, de plus grande taille, comme illustré au bas de la figure 4.

5) Comparés au modèle BM25 (Robertson *et al.*, 1994), qui est l'état de l'art en la matière, les meilleurs noyaux basés sur PLSI donnent de meilleurs résultats sur les corpus les plus difficiles sémantiquement : CISI, où les documents et les requêtes partagent peu de termes, ce qui en fait un échantillon de choix pour évaluer les modèles de recherche robustes à la synonymie ou qui utilisent des catégories latentes[9], MED (vocabulaire spécialisé) et TREC–AP.

---

6. 2940 : 5 corpus, 6 expériences, 7 nombres de catégories latentes et 14 noyaux.
7. `http://xapian.org/`
8. `http://trec.nist.gov/trec_eval/`
9. CISI est remarquable en ceci que certaines requêtes sont supposées extraire des documents avec lesquelles elles ne partagent *aucun* terme significatif.

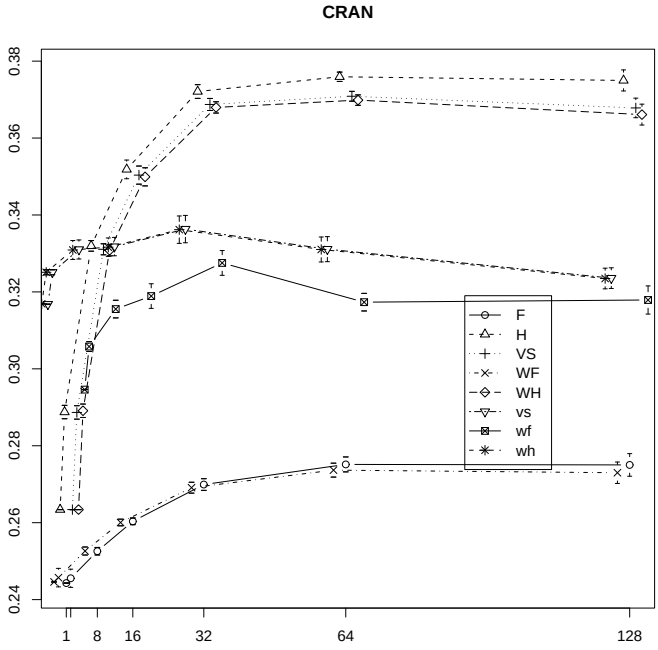


Figure 1. *Résultats typiques illustrés ici par la base CRAN, montrant que* $K^{DFIM\text{-}VS}(VS) \simeq K_w^{DFIM\text{-}H}(WH)$, $K^{VS}(vs) \simeq K_w^{H}(wh)$ *et* $K^{DFIM\text{-}F}(F) \simeq K_w^{DFIM\text{-}F}(WF)$.

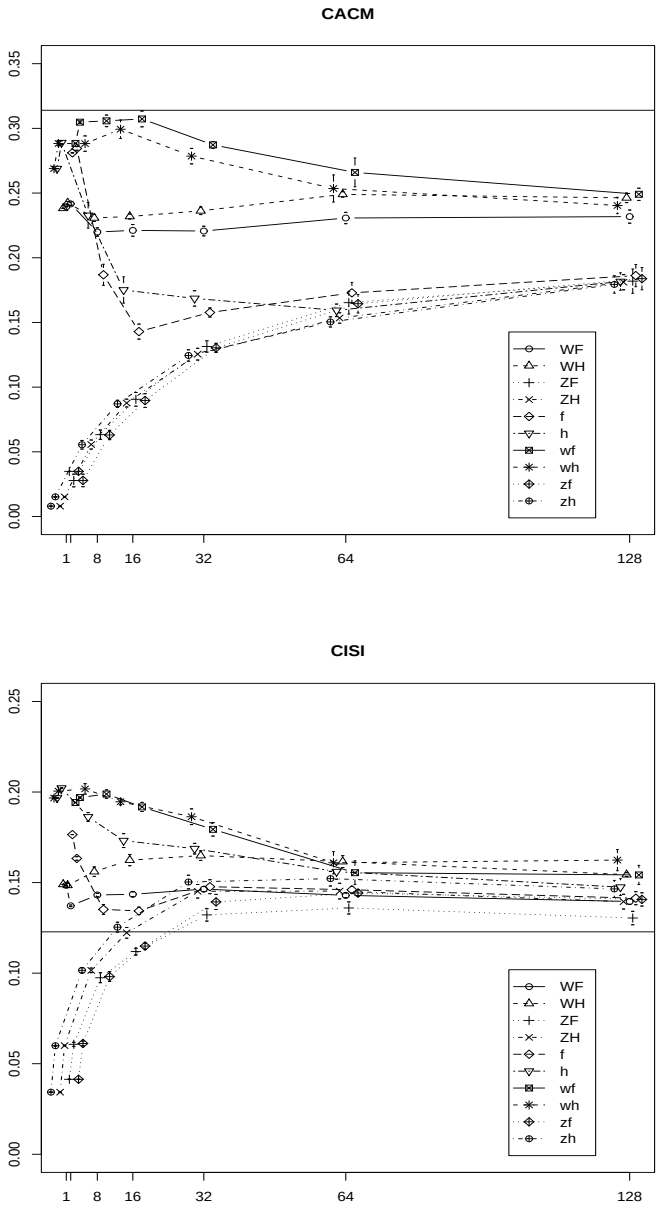


Figure 2. *Résultats obtenus sur les corpus CACM et CISI pour différents modèles :* $K_w^{DFIM\text{-}F}$ *(WF),* $K_w^{DFIM\text{-}H}$ *(WH),* $K_z^{DFIM\text{-}F}$ *(ZF),* $K_z^{DFIM\text{-}H}$ *(ZH),* K^H *(h),* K^F *(f),* K_w^F *(wf),* K_w^H *(wh),* K_z^F *(zf), et* K_z^H *(zh). La barre horizontale représente la performance du modèle BM25, indépendant de* $|Z|$*, qui constitue l'état de l'art. Autres corpus en figures 3 et 4.*

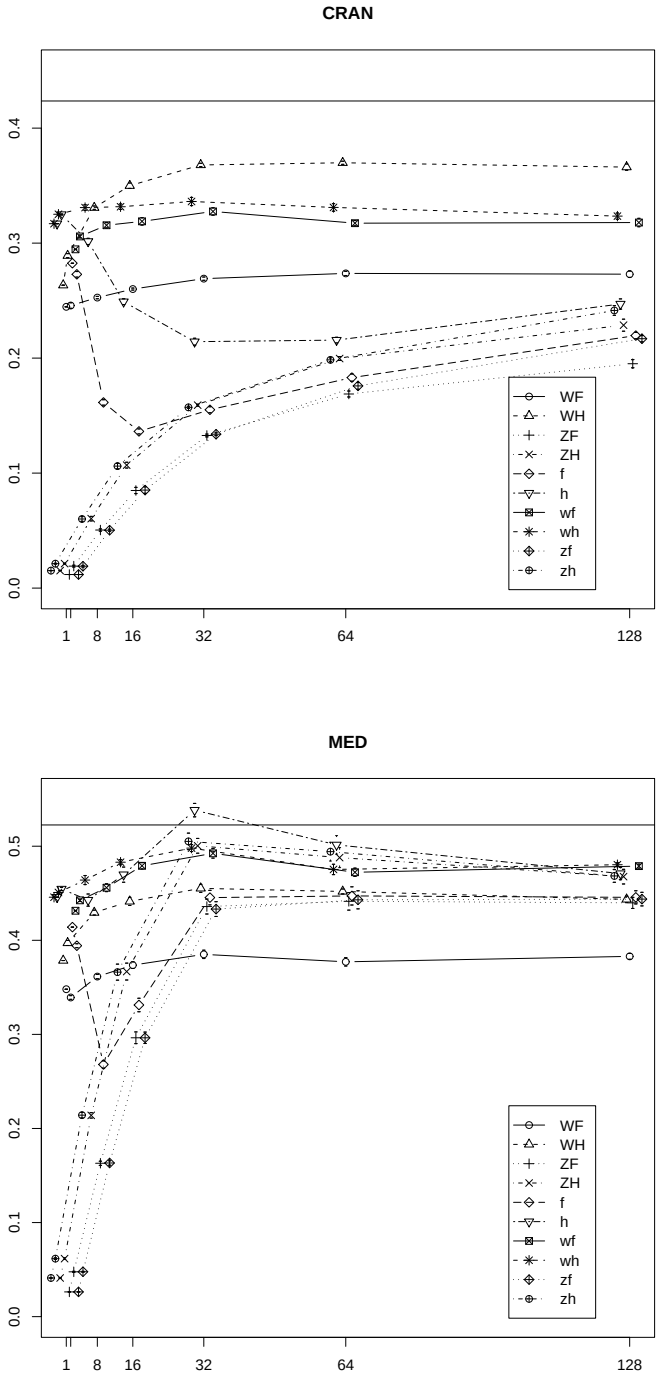


Figure 3. *Résultats obtenus sur les corpus CRAN et MED (légende figure 2).*

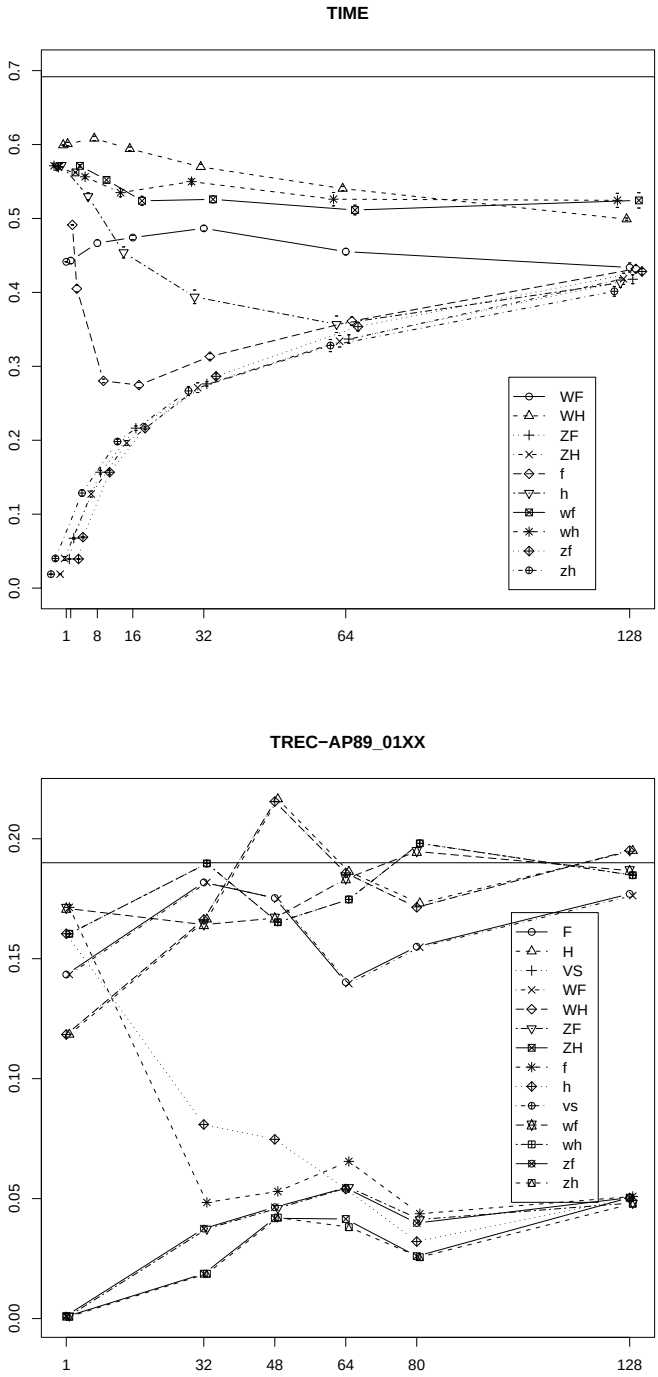


Figure 4. *Résultats obtenus sur les corpus TIME et TREC–AP (légende figure 2).*

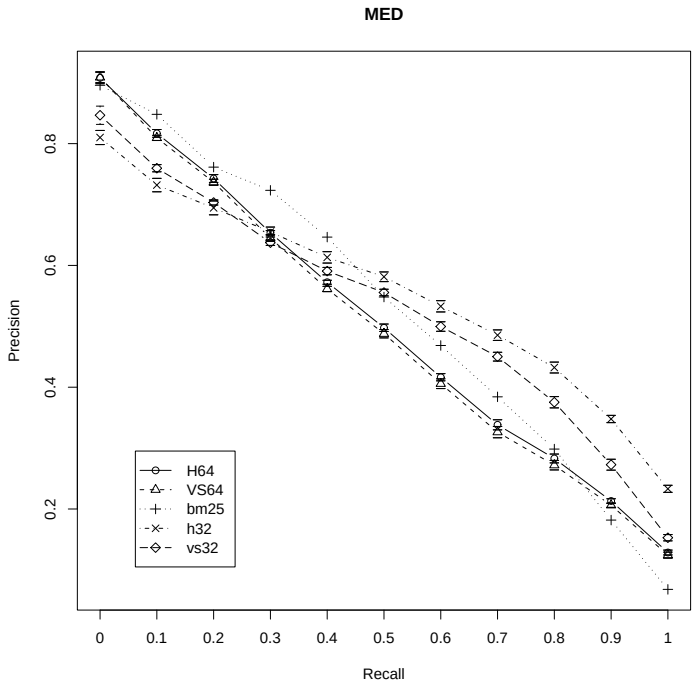


Figure 5. *Courbes précision-rappel sur le corpus MED pour BM25, $K^{DFIM\text{-}H}$, avec $|Z|$=64 (H64), $K^{DFIM\text{-}VS}$, avec $|Z|$=64 (VS64), K^{H}, avec $|Z|$=32 (h32), et K^{VS}, avec $|Z|$=32 (vs32).*

Le seul cas où les conclusions précédentes doivent être nuancées est le corpus MED. Sur ce corpus, les différents noyaux se comportent plus ou moins bien en fonction de la valeur de rappel considérée (Figure 5) : certains sont meilleurs à bas rappel et d'autres à haut rappel. Des mesures globales comme MAP ou R-Prec ne permettent pas de rendre compte de ce genre de nuances.

6. Conclusion

Le présent article analyse différentes dérivations de noyaux de Fisher pour le modèle PLSI. Il se concentre particulièrement sur le rôle de la matrice d'information de Fisher $G(\theta)$ et sur l'importance relative des composantes représentant les contributions des catégories sémantiques latentes et des termes, respectivement.

Nous avons pu confirmer expérimentalement que les modèles à sémantique latente comme PLSI peuvent se révéler intéressants dans des collections de documents sémantiquement difficiles, où documents et requêtes ne partagent pas nécessairement de termes significatifs, si tant est que des capacités de calcul suffisantes sont disponibles.

En ce qui concerne le rôle des composantes « catégories sémantiques latentes » et « termes », K_z peut clairement être négligé — tous du moins pour les nombres de catégories latentes qui permettent une utilisation pratique de tels modèles ($|Z|$ petit). Il est fort possible qu'un nombre beaucoup plus grand de catégories sémantiques latentes ($|Z|$ grand) améliore les performances de K_z, particulièrement pour les plus grandes collections de documents ; mais en pratique, il n'est pas possible d'entraîner de telles configurations, parce que PLSI ne passe pas à l'échelle, justement sur les grandes collections pour lesquelles l'idée serait prometteuse[10].

En ce qui concerne le rôle de la matrice d'information de Fisher, son rôle normalisateur améliore les résultats sur les collections les plus grandes (TIME, CRAN, TREC–AP89).

Globalement, pour les corpus où PLSI pourrait être avantageux par rapport aux modèles standards, il est recommandé d'utiliser $K_w^{\text{DFIM-H}}$ comme mesure de similarité.

7. Bibliographie

Ahrendt P., Goutte C., Larsen J., « Co-occurrence Models in Music Genre Classification », IEEE *Int. Workshop on Machine Learning for Signal Processing*, Sep, 2005.

Bosch A., Zisserman A., Munoz X., « Scene Classification via pLSA », *Proc. of the European Conf. on Computer Vision*, 2006.

10. Il a par exemple fallu 45 heures de temps de processeur et 6.7 Gb de RAM pour exécuter l'apprentissage EM pour la base TREC–AP sur un cœur d'un ordinateur octo-core Intel Xenon à 2 GHz.

Gaussier E., Goutte C., Popat K., Chen F., « A Hierarchical Model for Clustering and Categorising Documents », *Proc. of 24th BCS-IRSG European Colloquium on IR Research*, p. 229-247, 2002.

Harman D., « Overview of the Fourth Text REtrieval Conference (TREC-4) », *Proc. of Forth Text REtrieval Conf. (TREC–4)*, p. 1-23, 1995.

Hofmann T., « Probabilistic Latent Semantic Indexing », *Proc. of 22th Int. Conf. on Research and Development in Information Retrieval*, p. 50-57, 1999.

Hofmann T., « Learning the Similarity of Documents : An Information-Geometric Approach to Document Retrieval and Categorization », *Advances in Neural Information Processing Systems*, vol. 12, p. 914-920, 2000.

Hofmann T., « Unsupervised learning by probabilistic latent semantic analysis », *Machine Learning*, vol. 42, n° 1, p. 177-196, 2001.

Jin X., Zhou Y., Mobasher B., « Web usage mining based on probabilistic latent semantic analysis », *Proc. of 10th Int. Conf. on Knowledge Discovery and Data Mining*, p. 197-205, 2004.

Lienhart R., Slaney M., « PLSA on Large-scale Image Databases », *Proc. of the 2007 Int. Conf. on Acoustics, Speech and Signal Processing, IEEE, (ICASSP'2007)*, vol. 4, p. 1217-1220, 2007.

Mei Q., Zhai C., « A mixture model for contextual text mining », *Proc. of 12th Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, p. 649-655, 2006.

Monay F., Gatica-Perez D., « PLSA-based Image Auto-Annotation : Constraining the Latent Space », *Proc. ACM Int. Conf. on Multimedia (ACM MM)*, 2004.

Monay F., Gatica-Perez D., « Modeling semantic aspects for cross-media image indexing », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

Nyffenegger M., Chappelier J.-C., Gaussier E., « Revisiting Fisher Kernels for Document Similarities », *Proc. of 17th European Conference on Machine Learning*, vol. 4212 of *Lecture Notes in Computer Science*, Springer, p. 727-734, 2006.

Quelhas P., Monay F., Odobez J.-M., Gatica-Perez D., Tuytelaars T., Gool L. V., « Modeling scenes with local descriptors and latent aspects », *Proc. of ICCV 2005*, vol. 1, p. 883-890, 2005.

Robertson S. E., Walker S., Jones S., Hancock-Beaulieu M., Gatford M., « Okapi at TREC–3 », *Proc. of the Third Text REtrieval Conf. (TREC–3)*, 1994.

Steyvers M., Smyth P., Rosen-Zvi M., Griffiths T., « Probabilistic author-topic models for information discovery », *Proc. of 10th Int. Conf. on Knowledge Discovery and Data Mining*, p. 306-315, 2004.

Vinokourov A., Girolami M., « A Probabilistic Framework for the Hierarchic Organisation and Classification of Document Collections », *Journal of Intelligent Information Systems*, vol. 18, n° 2/3, p. 153-172, 2002.

Chapitre 6

Recherche d'Information dans les Documents Structurés

Identification et structuration hiérarchique des titres dans les documents HTML

Structuration hiérarchique des titres

Thierry Waszak*, — Claude de Loupy*,*** — Patrice Bellot****

* *Syllabs*
2, rue de Fontarabie, F-75020 Paris
{waszak, loupy}@syllabs.com

** *LIA / Université d'Avignon*
339, chemin des Mainajaries, Agroparc BP 1228, F-84911 Avignon

*** *MoDyCo / Université de Paris 10*
UMR 7114, 200, avenue de la République, F-92001 Nanterre

RÉSUMÉ. *Dans cet article, nous présentons une méthode pour automatiquement identifier et structurer hiérarchiquement les titres dans les documents HTML. Bien que la syntaxe HTML propose des balises de titres, l'usage de ces balises dans beaucoup de documents n'est pas correct ou ces balises ne sont pas utilisées. Notre méthode se base sur les propriétés visuelles, telles la taille ou la couleur de la police, obtenues grâce aux feuilles de style (CSS). L'hypothèse est que plus un élément est visible, plus son niveau dans la hiérarchie des titres est élevé. Nous avons extrait du Web un corpus de CSS que nous utilisons dans l'apprentissage d'un modèle de Markov caché. Les premiers résultats donnent une F-Mesure de 0,70 pour la structuration des titres et de 0,86 pour l'identification.*

ABSTRACT. *In this paper, we describe a method to automatically identify titles within Web pages. Although HTML syntax provides specific tags for titles, they are not always correctly used, and sometimes they do not even appear. We use visual clues like font size or colour provided by Cascading Style Sheets in order to retrieve the title hierarchy. The assumption is that the level of an element in the title hierarchy increases with its visibility. We automatically built a CSS corpus by crawling the Web and used it to learn a Hidden Markov Model which identifies titles and their hierarchy. Primary results give a F-Measure of 0.70 for titles structuring and 0.86 for titles identification.*

MOTS-CLÉS : *Hiérarchie des titres, Modèle de Markov Caché, Balises de visibilité, document HTML, Corpus Web.*

KEYWORDS: *Titles structuring, Hidden Markov Model, Visibility tags, HTML document, Web corpus.*

1. Introduction[1]

Avec la grande quantité d'information en constante augmentation disponible sur le Web, la nécessité de développer des méthodes permettant l'extraction de l'information pertinente à l'intérieur même des documents HTML est évidente. Il s'agit de rechercher le contenu d'un paragraphe spécifique en faisant abstraction du bruit que représente certaine partie du document (publicité, pop-up…). Cet article traite du problème de l'identification et de la structuration hiérarchique des titres dans les pages Web. En effet, les titres sont de bons indicateurs des sujets développés dans les paragraphes qu'ils introduisent. Ils peuvent donc aider à repérer l'information pertinente. Non seulement ils organisent sémantiquement le texte en en permettant une meilleure compréhension, mais ils sont aussi révélateurs de l'organisation spatiale du texte (Ho-Dac *et al.*, 2004 ; Jacques *et al.*, 2006). Ainsi, les titres peuvent être utilisés dans l'optique d'améliorer les systèmes de résumé automatique (en se basant sur les titres – résumé thématique) (Edmundson, 1969 ; Marcu, 1997). En recherche d'information, Hu *et al.* prend en considération ces propriétés des titres en donnant un poids plus élevés aux mots faisant partie du titre principal des documents ; cela a pour effet d'améliorer les performances de leur système (Hu *et al.*, 2005).

Il n'existe pas, à notre connaissance, de précédents travaux portant sur l'identification et la structuration hiérarchique des titres dans les pages Web : Hu *et al.* recherche uniquement le titre principal des pages. D'autres études se sont intéressées à la segmentation des documents HTML : il s'agit de structurer les pages Web en identifiant les différentes zones des pages et en leurs affectant un label. Ainsi la structuration des titres peut s'apparenter à une segmentation de document. En effet, les principaux titres d'une page, peuvent être utilisés pour identifier les différentes zones de cette page. En segmentation de document HTML, Song *et al.* ainsi que Xue *et al.* utilisent la représentation DOM des documents HTML afin d'en extraire les différentes propriétés qui permettront une classification automatique des zones des pages Web à l'aide de machines à vecteur support (SVN) et de réseaux de neurones (Song *et al.*, 2004 ; Xue *et al.*, 2007). Ces propriétés sont des balises HTML (celles définissant la taille de police, la couleur, *etc.*), la position dans l'arbre DOM représentant le document HTML et des propriétés linguistiques comme le nombre de mots. Afin d'avoir une méthode plus portable et d'être moins dépendant du format spécifique aux différents sites Web, Hattori *et al.* ne considère que le nombre d'apparition des balises dans l'arbre DOM ainsi que leurs profondeurs relatives (Hattori *et al.*, 2007) alors que Mukherjee *et al.* essaye de rassembler les différentes zones des pages Web en considérant leurs similarités dans l'arbre DOM (Mukherjee *et al.*, 2003).

1. Ce travail a été effectué dans le cadre du projet ANR-RNTL TextCoop (www.textcoop.org).

Notre méthode se base sur les feuilles de style (Cascading Style Sheets – CSS) généralement associées aux documents HTML. En effet, ce sont les CSS qui contiennent les informations sur la mise en page des documents HTML (taille de police, couleur, *etc.*). Or, ces propriétés visuelles sont de première importance dans l'identification des titres (Hu *et al.*, 2005 ; Song *et al.*, 2004 ; Xue *et al.*, 2007 ; Yang *et al.*, 2001). L'idée est de classer les différents styles (qui sont appliqués a un ou plusieurs éléments dans l'arbre DOM) en fonction de leurs relatives visibilités et de décider si un style est lié à un titre (avec un certain niveau hiérarchique – plus le niveau est élevé plus la visibilité est importante) en utilisant un modèle de Markov caché appris sur un corpus extrait du Web. Contrairement à Hu *et al.*, 2005 ; Mukherjee *et al.*, 2003 ; Song *et al.*, 2004 ; Xue *et al.*, 2007 ; Yang *et al.*, 2001, nous ne considérons pas les balises HTML elles-mêmes, mais plutôt les propriétés visuelles associées à une ou plusieurs balises (*i.e.* une suite de balise) HTML. Dans un modèle étendu, nous utilisons (comme Hattori *et al.*, 2007) les statistiques sur les balises du document HTML.

Remarquons enfin que la syntaxe HTML possède des balises spécifiques pour les titres *(<H1>, <H2>, etc.*). Ces balises permettent une identification ainsi que la structuration hiérarchique des titres. Le problème est que l'utilisation de ces balises n'est pas toujours correcte. En effet, beaucoup de pages Web utilisent ces balises dans un ordre incohérent : on trouvera par exemple *<H5>*, *<H2>*, *<H3>* au lieu de *<H1>*, *<H2>*, *<H3>*. Notre étude de corpus montre qu'au moins 25% des pages Web sont ainsi mal structurées. De plus, il existe des cas ou ces balises n'apparaissent même pas. Par conséquent, nous ne considérerons pas ces balises, mais seulement les balises (ou suite de balises) pouvant être associées à un style (CSS) et portant donc une certaine information visuelle.

Dans la section 2, nous présentons notre méthode pour la structuration hiérarchique de titres ; dans la section 3 nous expliquons comment nous avons automatiquement constitué un corpus extrait du Web. Finalement, notre approche est évaluée dans la section 4.

2. Méthode

La principale hypothèse est de considérer qu'un titre à un niveau *n+1* (*<Hi>*) est plus visible qu'un titre à un niveau *n* (*<Hi+1>*). Le titre (*<H1>*) est généralement écrit d'une manière plus « visible » (en gras, avec une taille de police plus grande, *etc.*) qu'un sous-titre (*<H2>*). Notre système se base donc sur ces propriétés visuelles des documents (Hu *et al.*, 2005 ; Song *et al.*, 2004 ; Xue *et al.*, 2007 ; Yang *et al.*, 2001). Ces propriétés sont retrouvées en considérant les feuilles de style (CSS) dont l'utilisation est maintenant largement répandue sur le Web. Ces feuilles de style servent en effet à définir le formatage du texte (l'aspect visuel) alors que le document HTML en lui-même ne doit contenir que le texte. Afin de pouvoir utiliser les informations visuelles présentes dans les CSS, la première étape de notre

approche consiste à analyser ces CSS associées aux documents HTML et de créer ce qu'on défini comme des « *pseudo documents* ». Ces pseudo documents se trouvent au cœur de notre approche et sont notamment utilisés lors de la phase d'apprentissage et d'analyse.

2.1. *Pseudo document*

Les CSS sont composées de balises HTML T_i (ou de séquences de balises – le texte dans le document HTML délimité par ces balises sera formaté avec un certain style défini dans la CSS) associées à la définition d'un style défini à l'aide de propriétés p_i associées à des valeurs v_i. Ainsi une ligne dans une CSS aura la syntaxe suivante : $T_i :\{(p_{ij} :v_{ij} ;)\}^+$. Par exemple : *DIV :{font-color :red ;font-weight :bold ;}* signifie que le texte délimité par une balise DIV sera en rouge et en gras. Il est à noter que pour cet exemple, la balise <*DIV*> dans le document HTML pourrait être remplacée par la suite de balise suivante : <*FONT COLOR="red"*><*B*>. Nous appellerons par la suite ce type de balises des *balises de visibilité*. On a donc une balise *clé* T_i qui peut être associée avec une ou plusieurs balises de visibilité V_{ij}. La ligne CSS peut donc être convertie en une *association* : (T_i,V_{ij}). Le pseudo document est alors simplement constitué de ces associations. Il est à noter que tous les couples (p_{ij},v_{ij}) ne sont pas considérés. En effet, la fonction de transformation [1] utilisée, permet une réduction du nombre de propriétés de deux points de vue.

$$Tr_v(p_{ij}, v_{ij}) = V_{ij} \qquad [1]$$

Tout d'abord, seuls les p_j pouvant être mis en relation avec des balises de visibilité sont conservés. Cette réduction est faite grâce à l'écriture de règles. Par exemple, des règles définissent qu'une indentation de texte sera transformée en * *, une marge ou un bord en <*BR/*> et toutes les propriétés sur les polices en différentes balises de visibilités : <*B*>, <*FONT SIZE="couleur"*>, *etc*. Deuxièmement, les tailles et couleurs sont normalisées. En effet, ces propriétés peuvent être numériques. Nous arrondissons donc ces valeurs afin de ne pas avoir plus de 19 tailles de police différentes et plus de 256 couleurs.

Remarquons que dans le cas où un document HTML n'est associé à aucune CSS, un pseudo document peut tout de même être généré. Dans ce cas, les balises de visibilités présentes dans le document HTML sont remplacées par des balises clés arbitraires. Par exemple, la suite de balises suivante : <*B*><*FONT SIZE="+2"*> sera remplacée par la balise clé <*T1*>.

Le tableau 1 est une illustration d'un pseudo document. Les niveaux de titre sont ce que nous cherchons et ce qui doit être étiqueté pour un apprentissage. On peut remarquer que le niveau de titre augmente alors que les <*Hi*> diminuent.

Puisque notre modèle de structuration hiérarchique des titres se veut statistique, un corpus annoté de pseudo documents est nécessaire. Ce corpus a été créé

automatiquement en récupérant des documents HTML ainsi que leur CSS en parcourant le Web. Ce corpus, le parcours du Web et l'étiquetage automatique des niveaux de titre sont présentés dans la section 3.

	Balise clé (T_i)	Balise de visibilité (V_{ij})	Niveau de titre
avec CSS	<P>	<B>	0
	<P>	<I>	0
	<H3>	<FONT SIZE=+2>	1
	<H2>	<FONT SIZE=+2><I>	1
	<H1>	<B><FONT SIZE=+3>	2
sans CSS	<T1>	<B>	0
	<T2>	<I>	0
	<T3>	<B><FONT SIZE=+2><I>	1
	<T4>	<B><FONT SIZE=+2>	2

Tableau 1. *Représentation de pseudo documents*

2.2. *Structuration hiérarchique des titres*

Afin de structurer les titres, il faut tout d'abord structurer le pseudo document. En effet, il s'agit d'ordonner le pseudo document de façon à avoir les balises les plus visibles en fin de document. Ainsi la dernière association du pseudo document représentera le titre de plus haut niveau. Il faut ensuite un modèle capable de reconnaitre un changement de niveau afin de savoir quand nous passons d'un titre de niveau *n* à un titre de niveau *n+1*. Pour cela nous utilisons un modèle de Markov caché (Hidden Markov Model – HMM) (Rabiner, 1990). Dans les prochaines sous-sections, nous introduisons le *score de visibilité* qui est utilisé pour ordonner les balises de visibilité dans le pseudo document puis le modèle HMM.

2.2.1. *Score de visibilité*

Cette fonction doit être capable de classer efficacement les balises en fonction de leur visibilité. On définit $S_i = V_{i1}\ldots V_{ij}$ (S_i est une suite de balises de visibilité). Soit la fonction S_v qui associe un score de visibilité à une suite de balises. S_v est défini comme la somme des probabilités qu'une balise de visibilité corresponde à un changement de niveau :

$$S_v(S_i) = \sum_{b_j \in S_i} P\left(changement | b_j\right) \quad [2]$$

où $P(changement|b_j)$ représente la probabilité que b_j (b_j est une balise de visibilité) corresponde à un changement de niveau. Ce score est calculé pour chaque b_j du corpus. Pour la phase d'analyse, on ne tient pas compte des b_j qui n'ont jamais été vu dans le corpus d'apprentissage.

2.2.2. *Modèle de Markov caché*

Les modèles de Markov cachés modélisent des séquences de données. Les HMM peuvent être vus comme une généralisation stochastique d'automates à état finis, où à la fois les transitions entre états et la génération des symboles sont gouvernés par des distributions de probabilité (Stolcke *et al.*, 1992). Un HMM peut être caractérisé comme suit : $HMM = \{W,V,A,B,\pi\}$ où W sont les différents états du modèle, V l'alphabet, A les probabilités de transition entre états, B les probabilités d'observation des symboles pour chaque état et π les distributions de probabilité initiales.

Pour notre problème de structuration des titres, nous voulons maximiser $P(L|D)$ où L représente la séquence des niveaux de titre dans le document D. (D fait référence au document HTML **et** à ses CSS associées.) D'après le théorème de Bayes, cela revient à maximiser :

$$P(L|D) = P(D|L).P(L) \quad [3]$$

Dans ce modèle, on décide de ne prendre en considération que la mise en forme (*i.e.* les suites de balises de visibilité S_j). On considère donc C (représentant cette mise en forme) comme une approximation de D. Ainsi [3] devient [4] :

$$P(L|D) = P(C|L).P(L) \quad [4]$$

Dans le modèle HMM, W représente les différents niveaux de titre : $W = \{0,\ldots,N\}$ où N est le plus haut niveau rencontré dans le corpus d'apprentissage ; $V = \{b_j\}_{(1\leq j\leq M)}$ avec M le nombre maximum de balises de visibilité b_j rencontrées dans le corpus ; les probabilités A et B sont obtenues grâce au corpus d'apprentissage. Remarquons qu'on doit avoir $\pi_0 = 1$ et $\pi_i = 0_{(1\leq i\leq N)}$. En effet, tous les pseudo documents, une fois ordonnés par leur score de visibilité, commencent par de balises de visibilité ne représentant pas un titre. La figure 1 illustre le HMM associé au pseudo document (avec CSS) présenté dans la table 1.

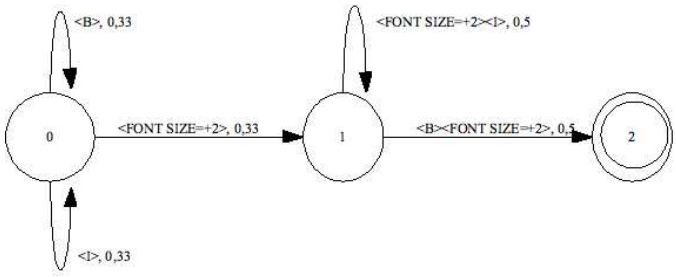


Figure 1. *Automate à états finis représentant un HMM*

Etant donnée une séquence d'observation de balises de visibilité, en utilisant l'algorithme de Viterbi, le modèle HMM nous donne la séquence optimale de niveaux de titre associée à cette observation.

2.3. *Modèle étendu*

Jusqu'ici uniquement les propriétés visuelles de mise en forme des documents HTML ont été utilisées. L'hypothèse du modèle étendu est que l'ajout d'autres propriétés issues non plus des CSS mais du document HTML lui-même peuvent améliorer les résultats. Comme évoqué dans (Hattori *et al.*, 2007), utiliser la grammaire des balises HTML peut s'avérer restrictif à un domaine (au corpus d'apprentissage). Nous décidons donc d'utiliser comme propriétés la fréquence d'apparition d'une balise dans le document HTML, ou encore la profondeur relative dans l'arbre DOM de la balise. L'hypothèse étant que les différents niveaux de titre sont moins fréquents que les paragraphes et qu'un titre ne peut pas avoir une profondeur plus importante qu'un sous-titre. Dans les sous-sections suivantes, nous présentons le modèle HMM étendu, puis ce même modèle auquel on applique certaines règles linguistiques.

2.3.1. *Description du modèle étendu*

D représentant le document HTML et ses CSS associées, on fait maintenant l'hypothèse que $D = H \cap C$ où *H* représente les propriétés extraites du document HTML et *C* celles extraites des CSS. En effet, le texte affiché dans un navigateur Web provient de la combinaison entre le texte lui-même et la mise en forme. On fait l'hypothèse que *H* et *C* sont indépendants et on a donc l'approximation suivante : $P(D|L) = P(C|L).P(H|L)$. L'équation [3] devient alors :

$$P(L|D) = P(C|L).P(H|L).P(L) \quad [5]$$

A l'aide des propriétés provenant du document HTML, on modifie également le score de visibilité S_v en pondérant la première formule [2] avec la fréquence d'apparition f_i des balises clés (T_i) et leurs profondeurs relatives d_i dans l'arbre DOM représentant le document HTML. En effet, pour une balise T_i, le niveau de titre diminue lorsque la profondeur augmente et lorsque la fréquence d'apparition augmente. On obtient le score de visibilité suivant :

$$S_v(S_i) = \frac{1}{f_i.d_i}.\sum_{b_j \in S_i} P\left(changement|b_j\right) \quad [6]$$

2.3.2. *Modèle étendu avec règles linguistiques*

A partir de l'observation du corpus, il apparaît que certaines règles linguistiques simples peuvent être ajoutées au précédent modèle afin d'améliorer le score de visibilité. Par exemple, si une balise clé apparaît moins de *x* fois (on fixe

empiriquement $x = 10$) dans le document HTML, elle a de forte chance de représenter un titre (dans la formule [7], on pose $\alpha = 1$ si $f_i \leq x$ et $\alpha = 0{,}9$ sinon). De plus, si la profondeur d'une balise dans l'arbre DOM est plus grande que celle de la balise précédente, cette précédente balise a de fortes chances d'être un titre de plus haut niveau ($\beta = 1$ sinon $\beta = 0{,}9$). Enfin, si une balise se trouve à la racine de l'arbre DOM ou à la profondeur maximale, elle ne représentera pas un titre ($\delta = 0$ sinon $\delta = 1$). On obtient pour le score de visibilité la formule suivante :

$$S_v(S_i) = \frac{\delta\alpha\beta}{f_i.d_i} . \sum_{b_j \in S_i} P\left(changement | b_j\right) \quad [7]$$

2.4. *Du pseudo document au document HTML restructuré*

A ce stade, seule la structuration des pseudo documents est faite. La dernière étape consiste à réaffecter aux balises clés les balises de titre identifiées dans les pseudo documents. Il suffit simplement de remplacer dans les documents HTML les balises clés en titres ou paragraphes. Le plus haut niveau de titre dans le pseudo document deviendra une balise *<H1>*, puis *<H2>*, *etc.* On peut encore rajouter ici quelques règles linguistiques pour corriger certaines erreurs que le modèle peut commettre. Ainsi, on remarquera que les titres sont souvent composés de peu de mots. On décide donc d'invalider la reconnaissance d'un titre de plus de *y* caractères (on fixe empiriquement $y = 80$). On fait aussi la correction dans le cas où un titre est identifié en fin de section : en effet, un titre est forcément suivit d'un paragraphe. Remarquons que ces règles portent uniquement sur les propriétés des documents HTML (*H*).

3. Corpus

Les méthodes d'apprentissage supervisées nécessitent un corpus d'apprentissage. Ce corpus doit être suffisamment important pour que l'apprentissage soit efficace ; ce qui demande du temps pour l'annotation. Afin de nous affranchir de ce problème, nous utilisons les propriétés de la syntaxe HTML qui prévoient des balises pour l'identification (et la structuration hiérarchique) des titres. Toute la difficulté est de parcourir le Web afin d'en récupérer seulement les documents *bien formés.*

3.1. *Documents bien formés*

Le principal problème des documents HTML est que l'utilisation des balises de titre, lorsqu'elles sont présentes, peut être faite dans un ordre incohérent. (Par exemple, *<H5>,<H2>,<H3>* au lieu de *<H1>,<H2>,<H3>*.) Ainsi, on considérera un document HTML comme correct, seulement si sa hiérarchie de titres est correctement suivit *i.e.* qu'il existe *<Hi>* dans le document HTML tel qu'il

existe j et n avec $1 \leq j,n \leq 6$ et $1 \leq j+n \leq 6$ tels que pour chaque i : $j \leq i \leq j+n$. (Le document HTML avec les balises <*H5*>,<*H2*>,<*H3*> ne sera donc pas considéré comme bien formé.) Cela ne garantit pas pour autant que l'ordre des titres dans les documents HTML restants soit correct. En effet, cette restriction ne nous assure pas que <*H2*> ne soit pas utilisée à la place de <*H1*> comme balise de titre de plus haut niveau. Nous décidons donc de ne conserver que les documents XHTML. En effet, il est raisonnable de faire l'hypothèse que l'ordre des balises de titre est respecté dans ces documents ; la syntaxe XHTML étant plus stricte que la syntaxe HTML, on peut donc penser qu'une attention particulière est donnée à la rédaction de ces documents XHTML.

Ainsi, afin d'obtenir un corpus exploitable, il nous suffit de parcourir automatiquement le Web à la recherche de ces documents bien formés (documents XHTML et leurs CSS associées) et d'en extraire les pseudo documents.

3.2. *Extraction du Web des documents bien formés*

La première étape consiste à poser une requête à un moteur de recherche[2] afin de retrouver des URL à partir desquelles on récupère les documents. Ainsi, on constitue des corpus en relation avec des requêtes : il s'agit en quelque sorte de corpus thématiques. La deuxième étape est le parcours du Web lui-même. Comme illustré dans la figure 2, on suit un algorithme de parcours en largeur.

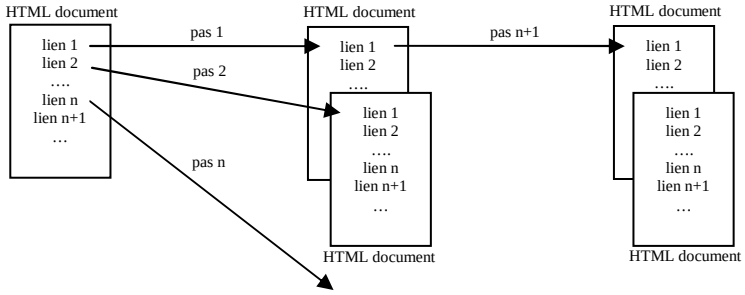


Figure 2. *Algorithme de parcours du Web*

En effet, comme le premier lien rapporté par un moteur de recherche doit être le lien le plus en relation avec la requête, ce lien doit d'abord être exploré. On explore ensuite le deuxième lien rapporté, *etc*. Ensuite, les liens des liens de la première page sont explorés et ainsi de suite. (On se limite au $n = 50$ premiers liens de chaque page. De plus, un maximum de $m = 7$ pages avec la même URL de base sont rapportées par soucis de couverture.) Cette stratégie de parcours du Web permet la

2. Nous avons utilisé Google (http://www.google.fr/)

création d'un corpus thématique dont la provenance n'est pas restreinte à un site particulier et donc à une certaine mise en forme. Il s'agit d'avoir la meilleure couverture de la grammaire HTML et CSS possible.

3.3. *Statistiques de corpus*

Dans le tableau 2 sont présentées les statistiques de corpus obtenues à la suite du processus présenté dans la section précédente. Ces statistiques sont calculées pour 5 différentes requêtes. Ces requêtes sont R1 : « CSS » ; R2 : « Histoire de France » ; R3 : « Politique » ; R4 : « Rugby » et R5 : « Afghanistan ». On récupère 200 documents pour chaque requête.

	R1	R2	R3	R4	R5	Moyenne
nombre de pages explorées	408	555	504	511	668	529
pourcentage de pages bien formées (pbf)	**0,49**	**0,36**	**0,40**	**0,39**	**0,30**	**0,39**
nombre de CSS par pbf	2,0	1,7	2,2	2,0	2,3	2,0
nombre de niveaux de titre par pages	2,7	2,7	2,3	2,7	2,6	2,6
pourcentage de pbf avec un ordre de titre cohérent	**0,78**	**0,77**	**0,84**	**0,70**	**0,55**	0,73
pourcentage de pbf contenant H1	0,88	0,90	0,87	0,81	0,81	**0,85**
pourcentage de pbf contenant H2	0,85	0,82	0,80	0,83	0,84	0,83
pourcentage de pbf contenant H3	0,69	0,57	0,51	0,60	0,56	0,59
pourcentage de pbf contenant H4	0,22	0,25	0,06	0,33	0,26	0,22
pourcentage de pbf contenant H5	0,06	0,07	0,02	0,11	0,13	0,08
pourcentage de pbf contenant H6	0,03	0,06	0,01	0,05	0,07	0,04

Tableau 2. *Statistiques des corpus*

On peut voir que seulement 39% des pages Web sont bien formées (de la façon dont nous l'avons défini). En moyenne, il y 2 CSS associées à une page bien formée et 2,6 niveaux de titre différents. De plus, seulement 73% des pages bien formées utilisent une hiérarchie de titre cohérente (comme présenté en section 3.1). On remarque aussi que 85% des pages utilisent la balise de plus au niveau <*H1*>. Cela sous entend que 15% des pages n'utilisent pas <*H1*> comme balise de titre de plus haut niveau ! Ces pages pourront tout de même nous servir dans l'apprentissage : on ne considère pas les balises elles-mêmes mais le style qui leur est associé ainsi que leur hiérarchie. En effet, dans le pseudo document, les balises de visibilité sont ordonnées selon leur hiérarchie et on pourra identifier le titre de plus haut niveau pour ce qu'il aurait du être : <*H1*>. En ordonnant les balises de visibilité suivant la hiérarchie suggérée par les balises de titre on peut ainsi obtenir un corpus d'apprentissage. Dans le souci d'avoir un corpus avec une meilleure couverture de la grammaire CSS et HTML, nous avons posé les mêmes requête et récupéré des corpus constitués de 1000 documents chacun. Les statistiques de corpus sont alors plus homogènes.

On peut aussi remarquer qu'il y a moins de pages bien formées parlant de l'histoire de France qu'il y en a propos des CSS. Cela pourrait s'expliquer par le fait que les pages parlant de CSS sont écrites par des informaticiens qui seront plus stricts dans l'utilisation de la syntaxe HTML.

Finalement, on voit qu'environ 27% de pages bien formées devraient être réordonnées. De plus, comme ces pages bien formées ne représentent que 39% des pages du Web, on voit bien qu'un outil de structuration hiérarchique des titres a tout son intérêt.

4. Evaluation

Dans cette section, nous présentons l'évaluation des différents modèles décrits précédemment. A cette fin, il nous faut tout d'abord définir les métriques que l'on va utiliser pour cette évaluation. Il faut noter que la première évaluation porte sur les pseudo documents ; on ne peut juger des performances de reconnaissance et de structuration des titres véritablement que sur la deuxième évaluation où les titres identifiés sont insérés dans les documents HTML.

4.1. *Métriques*

Nous effectuons une évaluation de nos modèles pour la tâche de structuration hiérarchique des titres mais aussi sur la tâche plus « simple » d'identification des titres. Dans ce cas, on cherche juste a savoir si un titre a été correctement reconnu. Pour cette tâche, on utilise les traditionnelles mesures de précision et de rappel. Dans le cas de la structuration hiérarchique des titres, on note l_i et l'_i respectivement le niveau de titre de référence et le niveau de titre supposé associés à une suite de balise de visibilité S_i. Précision et rappel sont définis de la manière suivante :

$$Précision = \sum_{l_i \neq 0} score(l_i) / nb(l'_i \neq 0) \quad [8]$$

$$Rappel = \sum_{l_i \neq 0} score(l_i) / nb(l_i \neq 0) \quad [9]$$

avec

$$score(l_i) = \sum_{l_j \neq l_i} ordre(l_j, l_i) / nb(l_j \neq l_i) \quad [10]$$

$$ordre(l_j, l_i) = \begin{cases} 1 \; si \; (l_j - l_i) > 0 \; et \; (l'_j - l'_i) > 0 \\ 1 \; si \; (l_j - l_i) < 0 \; et \; (l'_j - l'_i) < 0 \\ 0 \; sinon \end{cases} \quad [11]$$

où *ordre*(l_j,l_i) est égal à 1 si le niveau de titre l'_j est dans le même ordre par rapport à l'_i que l_j par rapport à l_i : il s'agit ici de donner plus de poids à un titre qui suit la hiérarchie de référence en vérifiant que quelque soit le niveau de titre l_j dans la hiérarchie de référence, le niveau de titre l'_j dans la hiérarchie supposée est dans le même ordre (plus grand ou plus petit) que le niveau du titre considéré (respectivement l_i et l'_i). *nb*($l_i \neq 0$) représente le nombre de titres dans la hiérarchie de référence et *nb*($l'_i \neq 0$) le nombre de titres identifiés par le modèle.

4.2. *Evaluation sur les pseudo documents*

A partir des cinq corpus extraits du Web (comme présenté en section 3.3), on produit cinq évaluations, une par corpus. A chaque fois, 80% du corpus est utilisé pour l'apprentissage et 20% pour l'évaluation. On procède à une évaluation croisée. Chaque corpus est constitué de 1000 documents. Les résultats obtenus, pour la structuration hiérarchique des titres puis pour la « simple » tâche d'identification des titres, sont présentés dans les tableaux 3 et 4. Dans ces tableaux, Modèle1 est le premier modèle « simple » que nous avons évoqué en section 2.2, Modèle2 est le modèle étendu présenté en section 2.3.1 et Modèle3 est le modèle étendu auquel on ajoute des règles linguistiques (section 2.3.2). (On note P pour la Précision et R pour le Rappel ; Ri sont les différentes requêtes *cf.* section 3.3.)

	R1		R2		R3		R4		R5		Moyenne	
	P	R	P	R	P	R	P	R	P	R	P	R
Modèle1	0,25	0,26	0,30	0,31	0,25	0,26	0,35	0,39	0,27	0,29	**0,28**	**0,32**
Modèle2	0,26	0,37	0,42	0,47	0,30	0,33	0,39	0,39	0,31	0,34	**0,34**	**0,38**
Modèle3	0,38	0,49	0,53	0,60	0,39	0,43	0,48	0,51	0,38	0,45	**0,43**	**0,50**

Tableau 3. *Structuration hiérarchique des titres dans les pseudo documents*

	R1		R2		R3		R4		R5		Moyenne	
	P	R	P	R	P	R	P	R	P	R	P	R
Modèle1	0,28	0,29	0,30	0,32	0,29	0,31	0,35	0,40	0,30	0,39	**0,30**	**0,34**
Modèle2	0,34	0,53	0,48	0,58	0,35	0,41	0,42	0,43	0,34	0,38	**0,39**	**0,47**
Modèle3	0,47	0,66	0,60	0,74	0,46	0,55	0,53	0,58	0,44	0,55	**0,50**	**0,62**

Tableau 4. *Identification des titres dans les pseudo documents*

On obtient de meilleurs résultats pour la tâche d'identification des titres. Cela n'est pas surprenant. La F-Mesure du Modèle1 au Modèle2 puis au Modèle3 passe de 0,30 à 0,36 et à **0,47** pour la structuration hiérarchique des titres et de 0,32 à 0,43 et à **0,56** pour l'identification de titres. On prouve ainsi l'importance des statistiques sur les balises HTML (*H*), qui peut s'apparenter à l'information structurelle des titres (Ho-Dac *et al.*, 2004). De plus, l'ajout de règles linguistiques permet un gain

de performance significatif par rapport au seul modèle statistique. Ces règles linguistiques modifiant uniquement le score de visibilité, cela montre également l'importance de la fonction de score de visibilité.

En différenciant les différents corpus, on remarque que celui traitant de l'histoire de France est celui qui obtient les meilleurs résultats. En effet, bien que les historiens écrivent moins de document bien formés, ils semblent utiliser moins de niveaux de titre (le plus souvent seulement <*H1*> et <*H2*>).

Il faut remarquer enfin que cette évaluation porte uniquement sur les pseudo documents. Or on peut s'apercevoir que tous les styles référencés dans une CSS, ne sont pas utilisés dans le document HTML. Ainsi, pour évaluer la structuration et l'identification des titres, il est préférable de le faire sur les documents HTML restructurés (comme décrit en section 2.4).

4.3. *Evaluation sur les documents HTML restructurés*

Cette évaluation à été menée sur une sélection aléatoire de 10 documents. Pour évaluer le système, il a fallu annoter ces 10 documents en ne considérant que la mise en forme de la page Web, ainsi que son contenu (on ne se base pas sur les balises HTML). Dans un premier temps ces documents ont été d'abord nettoyés : les parties non informatives (publicité, menus, *etc.*) ont été supprimées. Pour la tâche de structuration hiérarchique des titres, on obtient une précision de **0,66** et un rappel de **0,74** (soit une F-Mesure de 0,70) alors que pour la tâche d'identification, on obtient une précision de **0,92** et un rappel de **0,81** (soit une F-mesure de 0,86). Cette évaluation finale de notre système est bien meilleure que ce que laissait suggérer l'évaluation sur les pseudo documents. De plus ces résultats sont bien meilleurs que ce obtenus par Hu et al. sur l'identification du seul titre principal des pages Web (dans notre cas, on cherche à identifier tous les titres des pages).

5. Conclusion

Dans cet article, nous avons décrit une méthode pour automatiquement identifier et structurer hiérarchiquement les titres dans les documents HTML. Dans un premier temps, il s'agit d'utiliser un score de visibilité pouvant être attribué à une balise clé HTML en retrouvant dans la ou les feuilles de style associées le style à appliquer au texte encapsulé par cette balise clé (couleur du texte, taille de la police, *etc.*) Tout comme (Hu *et al.*, 2005 ; Song *et al.*, 2004 ; Xue *et al.*, 2007 ; Yang *et al.*, 2001), on se base sur les propriétés visuelles des documents. Dans un second temps, nous avons rajouté à ces propriétés des propriétés de la structure linéaire du texte que l'on peut extraire directement des documents HTML. Il s'agit de considérer seulement les statistiques des balises (fréquence, profondeur dans l'arbre DOM) et non la grammaire HTML afin de ne pas être dépendant d'un domaine (Hattori *et al.*, 2007).

L'évaluation a clairement montré l'importance de ces propriétés. En effet, les titres participent à cette organisation linéaire (Ho-Dac *et al.*, 2004 ; Jacques *et al.*, 2006). Notre modèle est basé sur un modèle de Markov caché qui identifie les changements de niveau de titre et leur hiérarchie. La principale difficulté a été de définir une fonction de score de visibilité la plus efficace possible. Enfin, l'ajout de certaines règles linguistiques simples (comme rejeter par exemple les titres de plus de 80 caractères) permettent d'améliorer sensiblement les résultats. On obtient une F-Mesure finale de 0,70 pour la structuration hiérarchique des titres et de 0,86 pour l'identification des titres.

Dans de futurs travaux, il nous faudra augmenter le nombre de documents pour l'évaluation sur les documents HTML restructurés (qui n'a été faite dans cet article que sur 10 documents). Nous pensons également intégrer une méthode de segmentation de document HTML comme présentée dans (Mukherjee *et al.*, 2003). En effet, cette technique tente d'identifier des séquences récursives de balises dans les arbres DOM. Identifier ces séquences nous permettra de reconnaître les sous-sections des documents et donc les sous-titres et titres plus facilement. On pourrait également étudier plus précisément les différences entre les différents domaines (issus des différentes requêtes posées au moteur de recherche) et évaluer les performances d'un modèle appris sur un domaine et appliqué à un autre. Un autre piste d'amélioration serait de combiner deux modèles de Markov caché appris d'une part sur les balises de visibilité et d'autre part sur la structure spatiale du texte.

6. Bibliographie

Edmundson H. P., « New Methods in Automatic Extracting », *Journal of the ACM*, vol. 16, 1969, p. 264-285.

Hattori G., Hoashi K., Matsumoto K., Sugaya F., « Robust web page segmentation for mobile terminal using content-distances and page layout information », *Proceedings of the 16th international conference on World Wide Web WWW'07*, 2007, p. 361-370.

Ho-Dac L., Jacques M., Rebeyrolle J., « Sur la fonction discursive des titres », *L'unité texte*, 2004, p. 125-152.

Hu Y., Xin G., Song R., Hu G., Shi S., Cao Y., Li H., « Title extraction from bodies of HTML documents and its application to web page retrieval », *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR'05*, 2005, p. 250-257.

Jacques M., Rebeyrolle J., « Titres et structuration des documents », *Actes International Symposium: Discourse and Document ISDD'06*, 2006, p. 1-12.

Marcu D., « From Local to Global Coherence: A Bottom-Up Approach to Text Planning », *Proceedings of the 14th National Conference on Artificial Intelligence AAAI'97*, 1997, p. 629-636.

Mukherjee S., Yang G., Tan W., Ramakrishnan I., « Automatic Discovery of Semantic Structures in HTML documents », *Proceedings of the Seventh International Conference on Document Analysis and Recognition ICDAR'03*, 2003, p. 669-679.

Rabiner L. R., « A tutorial on hidden Markov models and selected applications in speech recognition », *Readings in speech recognition*, 1990, p. 267-296.

Song R., Liu H., Wen J., Ma W., « Learning block importance models for web pages », *Proceedings of the 13th international conference on World Wide Web WWW'04*, 2004, p. 203-211.

Stolcke A., Omohundro S. M., « Hidden Markov Model Induction by Bayesian Model Merging », *Advances in Neural Information Processing Systems 5*, 1992, p. 11-18.

Xue Y., Hu Y., Xin G., Song R., Shi S., Cao Y., Lin C., Li H., « Web page title extraction and its application », *Information Processing and Management: an International Journal*, vol. 43, 2007, p. 1332-1347.

Yang Y., Zhang H., « HTML Page Analysis Based on Visual Cues », *Proceedings of the Sixth International Conference on Document Analysis and Recognition ICDAR'01*, 2001, p. 859-864.

Zou J., Le D., Thoma G. R., « Structure and content analysis for html medical articles: a hidden markov model approach », *Proceedings of the 2007 ACM symposium on Document engineering*, 2007, p. 199-201.

Classification de Structures Arborescentes : Cas de Documents XML

Ali Aïtelhadj*, **, Mohamed Mezghiche** et **Fatiha Souam*****

* *Université Mouloud Mammeri de Tizi-Ouzou (UMMTO) Algérie*
aaitelhadj@ymail.com

** *LIFAB (Labo d'Informatique Fondamentale et Appliquée de Boumerdes) Université M'hamed Bouguera de Boumerdes (UMBB) Algérie*
mohamed.mezghiche@yahoo.fr

*** *Université Mouloud Mammeri de Tizi-Ouzou (UMMTO) Algérie*
souam_fat@yahoo.fr

Chercheur

RÉSUMÉ. *Cet article présente une méthode de classification structurelle de documents XML. Notre approche consiste d'abord à extraire automatiquement la structure arborescente de chaque document XML à classer, et ensuite à utiliser cette structure comme modèle de représentation pour la classification du document XML correspondant. L'appariement de ces structures est fondé sur un calcul de leurs similarités. Pour l'expérimentation nous avons utilisé un corpus INEX.*

ABSTRACT. *In this paper we present a clustering method for XML documents. Our step is two-phase based: we first automatically extract the structure from the document; we then use it as model of representation to classify the document that it represents. The matching of the documents' structures is based on the calculation of their similarities. For the experimentation we used the INEX.*

MOTS-CLÉS: *Clustering, classification structurelle, structure, arbre, similarité, contexte, nœud, index, seuil.*

KEYWORDS: *Clustering, structural classification, structure, tree, similarity, context, node, index, threshold.*

1. Introduction

En très peu de temps, XML est devenu un standard incontournable pour la représentation et l'échange de données sur le Web. De plus, non seulement les collections de documents XML sont réutilisées, mais le volume de leurs échanges s'accroît continuellement. Cependant, avec les outils actuellement disponibles, la recherche d'informations dans ce type de documents est une tâche non triviale, notamment si elle concerne plusieurs domaines. Aujourd'hui les techniques d'accès aux données XML sont pour la plupart issues des travaux de la communauté des BD (Bases de Données) sur des données semi structurées.

Un document XML est caractérisé par un contenu (du texte) et une structure (balises). Ce type de documents ne peut cependant être exploité efficacement par les méthodes classiques de RI. En effet, ces dernières ne traitent un document que du point de vue de son contenu, alors que le format XML permet d'ajouter des contraintes structurelles (balises). Ceci impose alors d'adapter les méthodes classiques de RI ou d'introduire de nouveaux mécanismes afin d'exploiter au mieux les informations disponibles sous format XML. Notre travail rentre dans cette perspective et a pour motivation d'étendre les méthodes actuelles (classiques) de RI par un système de classification. Pour cela nous proposons une approche qui consiste d'abord à représenter chaque document XML par sa structure générique (ou résumé d'arbre), et ensuite à utiliser cette structure comme modèle de représentation pour la classification structurelle du document XML correspondant. Par conséquent, la classification d'une telle structure équivaut à la classification structurelle du document XML qu'elle représente.

La classification structurelle a pour rôle de regrouper des documents XML structurellement similaires dans des clusters (ou classes), afin de réduire le temps de réponse et augmenter la précision des moteurs de recherche. L'idée est que, si les documents partagent des structures similaires, ils sont plus à même de correspondre à la partie structurelle d'une requête. Notons que dans le cas d'XML les requêtes peuvent avoir une partie contenu et une partie structure.

Cet article est organisé de la manière suivante: outre l'introduction, la section 2 met en valeur la problématique et donne un état de l'art sur la classification structurelle de documents XML, la section 3 présente l'approche de classification structurelle que nous proposons, la section 4 est consacrée à une expérimentation de notre approche de classification, enfin la section 5, situe notre proposition de classification par rapport aux méthodes évoquées dans l'état de l'art.

2. Problématique et état de l'art

2.1. *Problématique*

La question primordiale est : *Comment doit-on indexer les informations des documents XML afin de pouvoir y accéder efficacement ultérieurement?*

En se focalisant sur la structure, on peut identifier certaines problématiques spécifiques liées au format XML. On peut citer *l'hétérogénéité sémantique* de la structure des documents XML (Denoyer, 2004). En effet, deux documents XML aux *contenus identiques* ou très proches thématiquement, peuvent avoir des *structures complètement différentes*. Le problème est : *Comment dans ce cas définir le formalisme que l'on va pouvoir utiliser pour formuler les requêtes utilisateur (besoin en information) composées de contenus et de structures?* Mais la problématique centrale qui nous incombe ici est *Comment peut-on mesurer la similarité structurelle de documents XML ?* Cette similarité soulève d'autres questions liées notamment au degré de similarité des structures, en particulier : *A partir de quel seuil de similarité doit-on considérer que deux documents peuvent être structurellement similaires?* C'est à ces questions que nous tentons de répondre dans cet article.

2.2. *Aperçu sur l'état de l'art*

Les approches de classification, se déclinent en deux variantes : la classification supervisée et la classification non supervisée (ou clustering). Dans la première, on dispose d'un ensemble d'objets classés, servant d'échantillon d'apprentissage. Le problème est alors d'associer à tout nouvel objet sa classe la plus appropriée, en se servant des exemples déjà étiquetés. Dans la seconde par contre (classification non supervisée), les classes possibles ne sont pas connues à priori, et les exemples disponibles sont non étiquetés. Le but est donc de regrouper dans un même cluster (ou classe) les objets considérés comme similaires, pour constituer les classes.

Les travaux sur la classification de documents XML existants se distinguent par la manière de représenter les documents, mais aussi par les méthodes de classification utilisées. Dans la *classification structurelle*, on distingue deux courants principaux. Dans le premier, on classe directement des documents XML. Dans le second par contre, au lieu de classer directement des documents XML, on classe leurs DTDs (Document Type Definition).

2.2.1. *Classification structurelle de documents XML*

Nous nous intéressons ici qu'aux approches qui représentent les documents XML par des arbres *étiquetés (ordonnés ou non)*. Les étiquettes correspondent à des tags XML. Un arbre est dit *ordonné* si les fils de chacun de ses nœuds (non feuilles), sont liés par une relation d'ordre totale de gauche à droite.

–Les approches de (Nierman *et al.*, 2002) et (Francesca *et al.*, 2003) utilisent des arbres *étiquetés* et *ordonnés*. La similarité entre deux arbres est basée sur leur *distance d'édition*. Une *distance d'édition* mesure le nombre d'opérations élémentaires (insertions et suppressions de nœuds) à effectuer pour transformer un

arbre en un autre. La *distance d'édition* permet d'effectuer un clustering de ces arbres en appliquant une méthode de classification hiérarchique ascendante.

–(Dalamagas *et al.*, 2004) utilisent des "résumés d'arbres" *étiquetés* et *ordonnés*. Comme illustré sur la figure 1, un "résumé d'arbre" est obtenu par deux transformations : la première réduit la *profondeur* de l'arbre de sorte que tout nœud (non feuille) ayant la même étiquette que l'un de ses ancêtres devienne un descendant direct (fils) de cet ancêtre ; la seconde élimine les *répétitions* des nœuds frères. L'arbre obtenu, peut ne pas être *exact*, mais il est *enraciné* (sa racine est conservée). Cette approche effectue aussi un clustering hiérarchique ascendant basé sur un calcul de *distance d'édition*.

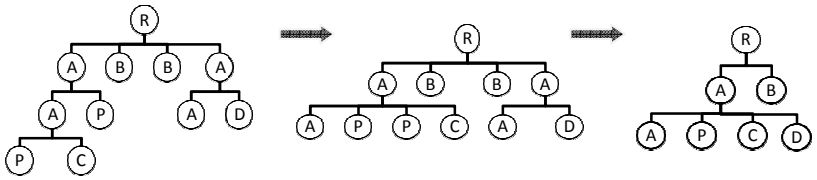


Figure 1. *Extraction de "résumés d'arbres"(Dalamagas et al, 2004)*

–(Termier *et al.*, 2002), (Costa *et al.*, 2004) et (Del Razo Lopez *et al.*, 2006), utilisent le formalisme des arbres fréquents. Ces approches sont fondées sur la découverte de sous-arbres qui apparaissent suffisamment fréquemment dans des collections d'arbres étiquetés. Dans (Termier *et al.*, 2002), les sous-arbres fréquents, peuvent être *ordonnés ou non*, ils peuvent aussi être des sous-arbres *exacts* ou bien conserver uniquement la notion d'*ancêtre*. Les sous-arbres de droite sur la figure 2 sont tous fréquents selon (Termier *et al.*, 2002). Dans les deux autres approches, les sous-arbres sont *ordonnés* et *enracinés*, comme les deux sous-arbres les plus à droite de la figure 2. (Costa *et al.*, 2004) s'appuient sur la *distance d'édition* pour effectuer un clustering hiérarchique. (Termier *et al.*, 2002) caractérisent un cluster par le sous arbre fréquent maximal commun aux sous arbres du cluster. Quant à (Del Razo López *et al.*, 2006), leur travail consiste à intégrer des schémas XML en vue de construire un schéma médiateur pour l'interrogation de documents XML.

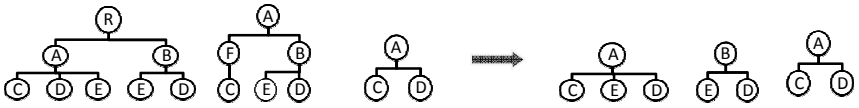


Figure 2. *Extraction de sous-arbres fréquents*

2.2.2. *Classification structurelle basée sur des DTDs*

Par définition, une DTD est considérée comme une grammaire à contexte libre qui engendre un nombre potentiellement infini de documents XML. De ce fait, au

lieu de classer directement des documents XML, l'approche préconise d'établir la classification de leurs DTDs. Ainsi chaque cluster devient le représentant d'un ensemble de documents XML structurellement similaires.

Parmi les travaux qui ont été rapportés sur la classification de DTDs, nous citons ci-après quelques uns des plus connus :

–Dans XClust (Mong *et al.*, 2002) les auteurs proposent un modèle de clustering de DTDs. Chaque DTD est représentée par un arbre. La similarité entre deux nœuds de deux arbres est calculée en exploitant plusieurs niveaux : la similarité ontologique des nœuds en utilisant un dictionnaire, la similarité de leurs ensembles de descendants immédiats (fils), la similarité de leurs ancêtres et enfin, la similarité des ensembles de feuilles des sous arbres dont ils sont respectivement les racines.

– Les auteurs de SPL (Schema Probabilistic Learning) (Su *et al.*, 2001) proposent un mécanisme qui identifie syntaxiquement la similarité entre DTDs en adoptant une stratégie de classification ascendante. Comparativement à XClust de (Mong *et al.*, 2002), SPL exploite uniquement les contextes des descendants immédiats (fils) pour deux nœuds.

–LSD (Learning Schema Document) (Doan *et al.*, 2001) est une approche basée sur l'apprentissage et l'inférence, combinés avec une instance de DTD. C'est une classification supervisée, c'est-à-dire, qu'on connaît les classes à l'avance.

– Enfin, dans Cupid (Madhavan *et al.*, 2001), les auteurs proposent un algorithme qui s'appuie sur un schéma générique de matching de DTDs. Le but du matching est de cibler un schéma médian vers lequel doivent converger toutes les DTDs ayant des structures similaires. Tout comme XClust (Mong *et al.*, 2002), pour calculer la similarité des arbres DTDs, l'approche Cupid de (Madhavan *et al.*, 2001), s'appuie sur un dictionnaire, mais elle exploite uniquement les contextes feuilles.

3. Approche pour la classification structurelle de documents XML

3.1. *Présentation générale de l'approche*

Notre approche se situe dans le premier courant, à savoir, dans les approches qui consistent à classer directement des documents XML. Plus précisément, dans notre cas, on classe directement des documents XML selon leurs structures arborescentes (en ignorant leurs contenus). Comme illustré sur la figure 3, la structure arborescente (résumé d'arbre) est extraite par un *extracteur*, elle est ensuite utilisée par un *classifieur* pour classer le document XML correspondant.

Figure 3. *Approche de classification structurelle de documents XML*

3.2. *Extraction de la structure générique*

Cette structure est générique car dans notre approche ce n'est pas toute l'information structurelle qui est utilisée pour représenter un document XML. En effet, lorsqu'une balise est dupliquée, il est inutile de répercuter cette duplication dans la structure que nous voudrions extraire. En outre, tout autre type de balises (commentaires, instructions.)„ sera ignoré. Cependant, les attributs seront considérés comme les fils (sous balises) des balises dans lesquelles ils apparaissent.

L'algorithme d'extraction de cette structure est composé de deux parseurs. Le premier s'appuie sur SAX (Simple API for XML), API (Application Programming Interface), qui renvoie tous les tags (balises et attributs), rencontrés sur un document XML. Ces derniers sont interceptés, filtrés puis transformés par un second parseur en le "résumé d'arbre" correspondant, conformément aux prévisions de notre approche. L'essentiel de la tâche d'extraction est accompli par ce deuxième parseur, car c'est lui qui permet de passer de la forme linéaire (source) du document à sa représentation hiérarchique (résumé d'arbre). Une illustration de cette extraction est schématiquement présentée par la figure 4.

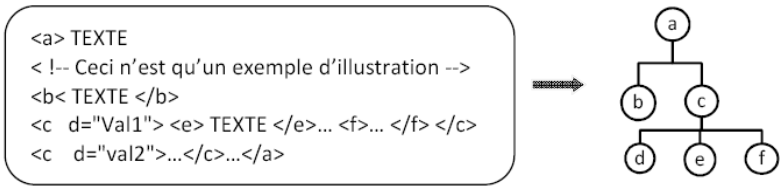


Figure 4. *Document XML et son "résumé d'arbre" généré par l'extracteur*

On remarque que le "résume d'arbre" est constitué uniquement de balises et d'attributs (les contenus du document ont disparu). On remarque sur le "résumé d'arbre", une seule occurrence du nœud c, alors qu'il y'en avait deux, sur le document XML en entrée. En outre, l'attribut d de la balise c apparait comme le fils du nœud c sur le "résumé d'arbre".

3.3. *Classification structurelle de documents XML*

3.3.1. *Rôles du cluster*

Dans notre modèle, un cluster remplit plusieurs rôles :

– Il peut être consulté à chaque fois lors du processus de classification pour accepter ou non de loger un document XML nouvellement traité.

–Il permet à un utilisateur d'interroger les documents XML qui s'y trouvent.

– Il peut servir ultérieurement pour créer un index basé conjointement sur les contenus et la structure des documents XML qui s'y trouvent.

3.3.2. *Structure d'un cluster*

On définit un cluster C_i par un ensemble d'arbres comme suit (formule [1]) :

$$C_i = \{T_{ik} / k = 1..p\} \quad \text{tel que } \lambda \leq Sim_{k=2..p}(T_{i1}, T_{ik}) \leq 1 \quad \text{et } 0 < \lambda \leq 1 \quad [1]$$

– $\{T_{ik}\}$ est l'ensemble des arbres du cluster C_i , et p son cardinal ;

– T_{i1} est le "résumé d'arbre" le plus pertinent à représenter l'ensemble des arbres du cluster C_i, on le nomme centroïde de C_i ;

– Sim est la fonction de similarité structurelle. Par définition, sa valeur est un nombre réel de l'intervalle [0,1].

– λ est le seuil à partir duquel les autres arbres $\{T_{ik} / k = 2..p\}$ de C_i sont similaires au centroïde T_{i1}.

REMARQUE. — NOUS VERRONS DANS L'ALGORITHME DE CLUSTERING QUI SUIT QUE LE CENTROÏDE EST MOBILE. CETTE « MOBILITE » EST FONDEE SUR LE PRINCIPE D'AUGMENTER AU MAXIMUM LA SIMILARITE ENTRE MEMBRES D'UN MEME CLUSTER, TOUT EN REDUISANT LA SIMILARITE ENTRE CLUSTERS.

3.3.3. *Algorithme de clustering*

```
/*  n étant le nombre de clusters et p = #Ci la taille du cluster Ci          */
Pour chaque arbre T /*  Représentant d'un document XML à classer             */
Faire Si Ω = ϕ   /*  L'ensemble des clusters Ω est vide                       */
     Alors      /*  Création du premier cluster                              */
          C1 ← T ; /* Placer T comme centroïde dans le cluster C1             */
          Ω = Ω ∪ {C1} / * Ajouter ce premier cluster C1 dans Ω               */
     Sinon  /*  Cas où il existe déjà au moins un cluster                     */
          /* On parcourt Ω pour chercher le cluster à associer à T           */
          Si arg max_{i=1..n} (λ ≤{Sim(Ti1, T)} ≤ 1) existe
          Alors  /* On a trouvé le cluster Ci le plus approprié pour T       */
               Ci ← {T} / * Affecter l'arbre T au cluster Ci                  */
               /*  Calcul du nouveau centroïde pour le cluster Ci            */
               Pour k=1..p /* Parcourir le cluster Ci en considérant          */
                          /* tous ses arbres, y compris le centroïde         */
               Faire Importance(Tik) = Σ_{Tij ∈ Ci} Sim(Tik, Tij) ;
               Ti1= arg max_{k=1..p}(Importance(Tik))
               /* Ti1 est le nouveau centroïde calculé pour Ci               */
          Sinon /*  On crée un nouveau cluster C pour l'arbre T              */
               C ← {T} ; /* T est le centroïde du cluster créé C             */
               Ω = Ω ∪ {C}
Fait
```

REMARQUE. — COMME ON PEUT LE CONSTATER, NOTRE METHODE DE CLASSIFICATION N'EST PAS SUPERVISEE. DE PLUS, LE NOMBRE DE CLUSTERS S'INCREMENTE AU FUR ET A MESURE QUE SE DEROULE LE PROCESSUS DE CLUSTERING.

Pour classer un arbre T (représentant un document XML à classer), l'algorithme précédent doit fonctionner comme suit :

– S'il n'existe aucun cluster dans la base Ω, il faut tout simplement en créer un en y mettant le premier arbre T qui se présente. On fixe initialement ce dernier comme centroïde de C_1.

– S'il existe déjà au moins un cluster, on parcourt la base Ω pour chercher s'il existe un cluster C_i qui permet de loger l'arbre T. Là aussi, deux cas peuvent être envisagés :

- Dans le premier, on est supposé avoir trouvé le cluster C_i le plus approprié pour l'arbre T. En l'occurrence, C_i est celui dont le centroïde T_{i1} donne le meilleur seuil de similarité avec l'arbre T. Cette question est réglée par la formule $argmax_{i=1..n}(\lambda \leq \{Sim(T_{i1}, T)\} \leq 1)$. Cependant, pour avoir le centroïde le plus représentatif pour C_i, il est primordial de calculer le poids de chacun de ses arbres. Pour cela, on utilise la fonction $Importance(T_{ik})$ qui nous permet d'identifier quel est l'arbre le plus important, et à ce titre, il sera désigné comme nouveau centroïde de C_i.

- Dans le second, il va falloir créer un nouveau cluster et l'intégrer dans la base Ω, ceci est réalisé par les deux dernières instructions de l'algorithme précédent.

3.4 *Calcul de la similarité structurelle de deux arbres*

De manière générale pour comparer deux termes on utilise un dictionnaire. Mais dans un contexte hiérarchique, il est nécessaire de considérer chaque terme (nœud) dans son étendue contextuelle. Ainsi, la similarité de deux nœuds va dépendre non seulement de leur similarité ontologique (en utilisant un dictionnaire), mais aussi, de celle de leurs contextes hiérarchiques. La figure 5 met en relief trois contextes : le *contexte ancêtres*, le *contexte des descendants immédiats* (fils) et le *contexte feuilles.*

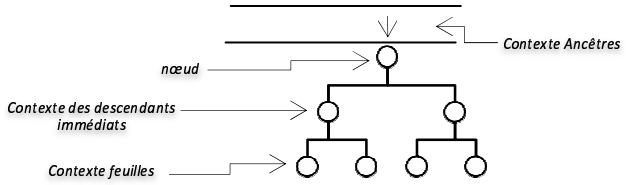


Figure 5. *Contextes utilisés pour un nœud*

A l'origine l'approche, utilisant ces trois context es, a été appliquée pour comparer des arbres représentant des DTDs dans (Mong *et al.,* 2002). Nous l'adaptons pour comparer des arbres représentant de s documents XML. Nous expliquons ci-après le formalisme utilisé.

La similarité de deux arbres T_1 et T_2 est calculée par la formule [2] suivante :

$$Similarité(T_1, T_2) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} sim(e_{1i}, e_{2j})}{Max(|T_1|,|T_2|)} \text{ où } \lambda \leq sim(e_{1i}, e_{2j}) \leq 1 \qquad [2]$$

$sim(e_{1i}, e_{2j})$ représente la similarité des nœuds e_{1i} et e_{2j}, qui est égale, au moins à λ et au plus à 1. Les nœuds e_{1i} et e_{2j} appartiennent respectivement aux arbres T_1 et T_2. $|T_1|$ et $|T_2|$ étant respectivement les tailles (nombres de nœuds) des arbres T_1 et T_2. La division par $Max\ (|T_1|, |T_2|)$ permet de normaliser le résultat de la sommation. $|T_1| = n$ et $|T_2| = m$. Le résultat sera la similarité des arbres T_1 et T_2.

Mais le calcul proprement dit, de la similarité de deux nœuds e_1 et e_2, tel qu'il est préconisé dans l'approche adoptée, nécessite d' introduire de nouvelles considérations qui tiennent compte du contexte hiérarchique de chaque nœud. Ainsi, pour synthétiser le calcul de la similarité de deux nœuds e_1 et e_2, la formule [3] agrège trois types de similarités comme suit :

$$sim(e_1, e_2) = w_1 * S_1(e_1, e_2) + w_2 * S_2(e_1, e_2) + w_3 * S_3(e_1, e_2) \qquad [3]$$

$$o\grave{u}\ \ w_1 \geq 0, w_2 \geq 0 \text{ et } w_3 \geq 0 \text{ sont des poids tels que } \ w_1 + w_2 + w_3 = 1$$

– $S_1(e_1,\ e_2)$ représente la *similarité sémantique* de e_1 et e_2. Cette dernière dépend de la similarité ontologique de e_1 et e_2 (déduite à partir d'un dictionnaire), et de la similarité de leurs ancêtres respectifs (calculée par la formule [5]).

La *similarité sémantique* de e_1 et e_2 est calculée par la formule [4] suivante :

$$S_1\ (e_1, e_2) = PCC(r_1.e_1,\ r_2.e_2) * S_0(e_1,\ e_2) \qquad [4]$$

$S_0\ (e_1, e_2)$ étant la similarité ontologique des nœuds e_1 et e_2. Un simple algorithme permet de calculer cette dernière. En effet, si les nœuds sont représentés par un même terme, leur similarité est évidemment égale à 1; s'ils sont synonymes, elle peut être fixée par exemple à 0.8 ou 0.9, sinon elle est égale à 0. Sur le tableau 1 on a un exemple de similarité ontologique de deux arbres. Les lignes sont représentées par les nœuds du premier arbre, les colonnes par ceux du second.

	facture	article	q	pu	design
facture	1	0	0	0	0
produit	0	0.9	0	0	0
code	0	0	0	0	0
qte	0	0	0.9	0	0
prixunit	0	0	0	0.9	0
num	0	0	0	0	0

Tableau 1. *Matrice de similarité ontologique de deux arbres*

$PCC(r_1.e_1,\ r_2.e_2)$ (Path Context Coefficient) représente la similarité des chemins $r_1.e_1$ et $r_2.e_2$ allant respectivement des racines r_1 et r_2 aux nœuds e_1 et e_2. En fait, c'est la similarité des ancêtres respectivement des nœuds e_1 et e_2. Les contextes ancêtres jouent un rôle prépondérant dans le calcul de la similarité. L'idée est que même si deux nœuds sont ontologiquement identiques ou similaires, ils ne le resteront pas forcément dans les contextes de leurs ancêtres respectifs qui peuvent être complètement différents. Pour calculer $PCC(r_1.e_1,\ r_2.e_2)$, on applique la formule [5] suivante :

$$PCC(r_1.e_1,\ r_2.e_2) = \frac{\sum_{i=1}^{p}\sum_{j=1}^{q} sim\_path(n_{1i}, n_{2j})}{Max(|r_1.e_1|,\ |r_2.e_2|)}$$

$$avec\ \lambda \leq sim\_path(n_{i1}, n_{2j}) \leq 1 \qquad [5]$$

$sim\_path(n_{1i},\ n_{2j})$ représente la similarité ontologique des nœuds n_{1i} et n_{2j}, qui est au moins égale à λ et au plus égale à 1. n_{1i} et n_{2j} appartiennent respectivement aux chemins (paths) $r_1.e_1$ et $r_2.e_2$. On normalise la sommation par le nombre de nœuds du plus long des chemins $r_1.e_1$ et $r_2.e_2$, c'est-à-dire, $Max(|r_1.e_1|,\ |r_2.e_2|)$. $|r_1.e_1| = p$ et $|r_2.e_2| = q$.

– $S_2(e_1, e_2)$ correspond à la *similarité des descendants immédiats* (fils) respectivement de e_1 et e_2. Autant que les contextes ancêtres, les contextes des descendants jouent également un rôle capital dans le calcul de la similarité. En effet, deux nœuds similaires ou identiques peuvent devenir complètement dissimilaires sous la contrainte de leurs descendants. La formule [6] permet de calculer la *similarité des descendants immédiats* (fils) respectivement de e_1 et e_2.

$$S_2(e_1,\ e_2) = \frac{\sum_{i=1}^{r}\sum_{j=1}^{s} sim\_desc(n_{1i},\ n_{2j})}{Max(\ |desc_1|,\ |desc_2|)}$$

$$avec\ \lambda \leq sim\_desc(n_{1i}, n_{2j}) \leq 1 \qquad [6]$$

$sim\_desc(n_{1i},\ n_{2j})$ représente la similarité ontologique, des nœuds n_{1i} et n_{2j}, qui est au minimum égale à λ et au maximum égale à 1. n_{1i} et n_{2j} appartiennent respectivement aux ensembles des fils de e_1 et e_2. On normalise la sommation par le plus grand des cardinaux des deux ensembles des descendants immédiats $desc_1$ et $desc_2$, c'est-à-dire, $Max(\ |desc_1|,\ |desc_2|)$. $|desc_1| = r$ et $|desc_2| = s$.

– $S_3(e_1, e_2)$ correspond à la *similarité des contextes feuilles* des sous arbres respectivement de racines e_1 et e_2

REMARQUE. — POUR LEVER TOUTE EQUIVOQUE, LE TERME *feuilles* ICI DESIGNE NON PAS DES CONTENUS, MAIS TOUT SIMPLEMENT DES NŒUDS TERMINAUX D'UN ARBRE.

Le contexte feuilles a lui aussi son importance dans le calcul de la similarité. En effet, intuitivement deux nœuds similaires dans les contextes de leurs ancêtres et fils respectifs, ont plus de chance de le rester s'ils présentent des feuillages similaires que ceux qui n'en présentent pas. La formule [7] permet de calculer la *similarité des contextes feuilles* des sous arbres respectivement de racines e_1 et e_2.

$$S_3(e_1,\ e_2) = \frac{\sum_{i=1}^{t}\sum_{j=1}^{u} sim\_leaf(n_{1i},\ n_{2j})}{Max(\ |leave(e_1)|,\ \ |leave(e_2)|)}$$

$$avec\ \lambda\ \leq sim\_leaf(n_{1i}, n_{2j}) \leq 1 \qquad [7]$$

$sim\_leaf(n_{1i},\ n_{2j})$ représente la similarité des feuilles n_{1i} et n_{2j} qui est au minimum égale à λ et au maximum égale 1. n_{1i} et n_{2j} appartiennent respectivement aux ensembles des feuilles des sous-arbres de racines e_1 et e_2. On note ces derniers respectivement par $leave(e_1)$ et $leave(e_2)$. On normalise la sommation par le plus grand des cardinaux de ces deux ensembles, c'est-à-dire, $Max(\ |leave(e_1)|, |leave(e_2)|)$. $|leave(e_1)| = t$ et $|leave(e_2)| = u$.

Mais pour calculer la similarité $sim\_leaf(n_{1i},\ n_{2j})$ proprement dite, on reconduit la logique de la formule [4] pour chaque paire de nœuds feuilles (n_{1i}, n_{2j}), comme indiqué par la formule [8] suivante :

$$sim\_leaf\ (n_{1i},\ n_{2j}) = PCC\big(e_1.n_{1i},\ e_2.n_{2j}\big) * S_0\big(n_{1i},\ n_{2j}\big) \qquad [8]$$

Tout comme avec la formule [4], la formule [8] cherche à calculer la similarité des nœuds n_{1i} et n_{2j}, qui sont des feuilles appartenant respectivement aux sous arbres de racines e_1 et e_2. Ce calcul implique la similarité ontologique $S_0\big(n_{1i},\ n_{2j}\big)$ (dictionnaire), des nœuds feuilles n_{1i} et n_{2j} et la similarité $PCC\big(e_1.n_{1i},\ e_2.n_{2j}\big)$ des chemins (similarité des ancêtres), allant de e_1 et e_2 respectivement à n_{1i} et $n_{2j.}$

Ce système de formules est applicable de manière générale à des nœuds internes ou racines. Cependant, si au moins un des nœuds e_1 ou e_2 est une feuille, il est évident qu'il(s) ne possède(nt) pas de descendants (ni fils, ni feuilles), et par conséquent, le calcul de leur similarité s'en trouve biaisé, car $S_2(e_1, e_2) = 0$ et $S_3(e_1, e_2) = 0$, alors qu'ils peuvent être ontologiquement identiques ou synonymes. Alors pour y remédier, le calcul des *similarités des descendants immédiats* et des *contextes feuilles* de ces nœuds sera remplacé par celui de leur *similarité sémantique*, comme indiqué ci-dessous par [9] et [10] :

$$S_2(e_1, e_2) = S_1(e_1, e_2) \qquad [9]$$

$$S_3(e_1, e_2) = S_1(e_1, e_2) \qquad [10]$$

4. Expérimentation

4.1 *Implémentation du système de classification*

Nous avons développé un premier programme en Java sous l'environnement JCreator. Le programme développé consiste en deux modules : le premier s'appuie sur SAX (Simple API for XML), API (Application Programming Interface), pour effectuer un premier parsing comme annoncé en section 3 (sous section 3.1). Ce module fournit un fichier intermédiaire intercepté par un deuxième module pour finaliser l'extraction du "résumé d'arbre".

Nous avons ensuite écrit un deuxième programme en C++ qui utilise les fichiers (résumés d'arbres) générés par le premier programme pour les clustériser.

Ces deux programmes correspondent respectivement à l'extracteur et au classifieur évoqués dans la sous section 3.1 à travers la figure 3.

4.2 *Evaluation des performances et Résultats*

Nous avons utilisé 2 collections de documents XML :

– Un corpus réel de documents XML ACM SIGMOD téléchargé à partir de www.acm.org/sigmod/record/xml. Ce corpus concerne les articles publiés par ACM SIGMOD.

– Une collection de documents XML fabriqués à partir d'un échantillon issu d'un corpus INEX 2007 (Initiative for the Evaluation of XML Retrieval). *INEX est une campagne d'évaluation des Systèmes de Recherche d'Information sur des documents XML. Son objectif principal est de promouvoir l'évaluation de la recherche sur des documents XML*.

Pour cette expérimentation nous n'avons pas eu besoin d'utiliser un dictionnaire (les vocabulaires utilisés pour les structures dans ces corpus sont standards). Nous avons alors inhibé la procédure de recherche des synonymes dans notre programme de classification (classifieur).

Le corpus ACM SIGMOD choisi est composé de 48 documents XML et de 4 DTDs qu'on peut considérer comme des classes cibles par rapport auxquelles on peut évaluer notre clustering. Par contre, les documents du corpus INEX sont hétérogènes (leurs DTDs ne sont pas connues à priori). Mais pour pouvoir effectuer notre test de clustering avec un tel corpus, nous avons fabriqué une collection comportant 250 documents XML avec l'outil Altova XMLSpy v2009. L'outil en question a été téléchargé à partir du site www.altova.com. Il permet de générer et d'associer une DTD pour un document XML et vice versa.

Pour préparer cette collection, nous avons choisi 10 documents XML INEX très hétérogènes pour lesquelles nous avons générés 10 DTDs par Altova XMLspy. Mais pour s'assurer que ces DTDs sont effectivement très différentes, nous avons utilisé un autre outil Altova DiffDog (téléchargé aussi de www.Altova.com), qui permet de

comparer deux documents et visualiser leurs différences. Ensuite, pour chacune de ces 10 DTDs ainsi fabriquées, nous avons construit un ensemble de 25 documents XML validés par Altova XMLspy. Chacun de ces ensembles contient des documents tirés du corpus INEX et des documents XML artificiels générés automatiquement par Altova XMLspy. Notons que pour confectionner un document XML à partir d'une DTD, Altova XMLspy offre plusieurs options. C es options consistent en des choix multiples sur le nombre de nœuds à garder, les répétitions de ces nœuds, les attributs associés à ces nœuds, les contenus, et même les tailles des tags.

Dans la phase suivante nous avons extrait d'abord des deux collections précédentes, les "résumés d'arbres" correspondants, ensuite nous avons procédé respectivement à leur clustering. Pour évaluer la pertinence de notre approche, nous avons utilisé les mesures *P* (Précision) et *R* (Rappel), exprimées respectivement par les formules [11] et [12] suivantes :

$$P = \frac{Nombre\ d'arbres\ \ bien\ classés}{Nombre\ d'arbres\ du\ cluster} \quad [11]$$

$$R = \frac{Nombre\ d'arbres\ \ bien\ classés}{Nombre\ de\ documents\ de\ la\ DTD} \quad [12]$$

Nous avons effectué deux tests, en fixant les poids d'agrégation aux valeurs $w_1 = \frac{1}{3}$, $w_2 = \frac{1}{3}$ et $w_3 = \frac{1}{3}$ et le seuil de similarité respectivement à λ =0.5 et λ =0.7. Les résultats expérimentaux sont donnés par les tableaux 2 et 3. Les abréviations utilisées dans ces tableaux sont : Doc désigne le nombre de documents XML ; CL : Cluster ; S : similarité ; N : désigne le nombre d'arbres d'un cluster; et enfin X-Y dans la colonne S, signifie X ≤ valeur(S) ≤ Y.

Corpus réel							
		λ =0.5			λ =0.7		
Classes cibles	Doc	CL	N	S	CL	N	S
DTD1	30	C1	30	0.57-1	C1	19	0.94-1
DTD2	16	C2	16	0.86-1	C2	16	0.86-1
DTD3	1	C3	1	-	C3	1	-
DTD4	1	C4	1	-	C4	1	-
					C5	11	0.94-1
		P =1, *R* =1			*P* =1, *R* =0.76		

Tableau 2. *Résultats de classification avec le corpus réel*

Corpus synthétique							
		λ =0.5			λ =0.7		
Classes cibles	Doc	CL	N	S	CL	N	S
DTD1	25	C1	25	0.67-1	C1	20	0.83-1
DTD2	25	C2	25	0.86-0.94	C2	25	0.86-0.94
DTD3	25	C3	25	0.81-0.96	C3	25	0.81-0.96
DTD4	25	C4	25	0.60-0.91	C4	19	0.78-0.91
DTD5	25	C5	25	0.65-1	C5	23	0.81-1
DTD6	25	C6	25	0.83-1	C6	25	0.83-1
DTD7	25	C7	25	0.86-1	C7	25	0.86-1
DTD8	25	C8	25	0.78-0.94	C8	25	0.78-0.94
DTD9	25	C9	25	0.91-1	C9	25	0.91-1
DTD10	25	C10	25	0.94-0.98	C10	25	0.94-0.98
					C11	5	0.96-1
					C12	6	0.88-1
					C13	2	1
		P =1, *R* =1			*P* =1, *R* =0.948		

Tableau 3. *Résultats de classification avec le corpus synthétique*

REMARQUE. ~~S~~ ACHANT QUE LES SIMILARITES OBTENUES AVEC λ =0.5, SONT TOUTES SITUEES ENTRE 0.57 ET 1, NOUS N'AVONS REFAIT LE TEST (λ =0.7) QUE POUR CERTAINS ARBRES, A SAVOIR, CEUX DU CLUSTER5 (1ER CORPUS) ET CEUX DES CLUSTERS (CLUSTER11, CLUSTER12 ET CLUSTER13 DU 2EMECORPUS). LES SIMILARITES DANS CES CLUSTERS SONT TRES ELEVEES (ENTRE 0.88 ET 1).

Les valeurs des précisions *P* pour les deux corpus sont toutes à 1. Ceci s'explique par notre approche de représentation des documents XML. En effet, dans notre cas, un document XML est représenté par un "résumé d'arbre" très proche d'un arbre qu'on aurait dérivé à partir de la DTD a ssociée au document. Mais avec λ =0.7, les valeurs moyennes des rappels *R* sont de 0.76 et 0.948 respectivement pour les deux collections. En effet, dans certains cas, certains arbres présentent des similarités (par rapport au centroïde) insuffisantes (0.57 et 0.67) respectivement pour les deux corpus, qui ne leur permettent pas de rester regroupés dans les mêmes clusters comme dans le cas de λ =0.5. Ce décalage est dû à une perte d'information

provoquée par l'élimination de sous-arbres dont les racines sont des répétitions de nœuds frères. En outre, pour calculer la similarité de deux arbres, le système de formules ([4], [5], ..,.[10]), tient compte de l'age ncement hiérarchique de leurs nœuds, si bien qu'on ne trouve dans un même cluster que les documents ayant des structures hiérarchiquement très proches.

Nous avons vérifié manuellement les arbres ayant donné les plus petites similarités (0.57, 0.60, 0.65 et 067), et nous avons constaté qu'ils ne présentaient en fait, que de petites différences avec leurs centroïdes respectifs. Nous avons trouvé que ces différences sont en moyenne de l'ordre de 35%. Donc un taux moyen de 65% de nœuds en commun a suffi pour obtenir des similarités nettement supérieures à $\lambda = 0.5$. De ce fait, et conformément aux résultats obtenus avec $\lambda = 0.7$, il semble que $\lambda = 0.5$ est une valeur moyenne satisfaisante pour notre approche avec le deux corpus testés.

5. Caractéristiques de la méthode proposée vis-à-vis de l'état de l'art

Par rapport à certaines méthodes évoquées dans l'état de l'art, notre proposition se démarque par un certain nombre de points importants.

– En effet, tout comme dans (Nierman *et al.*, 2002), (Francesca *et al.*, 2003) et (Dalamagas et al., 2004), dans notre approche un document XML est représenté par un "résumé d'arbre" *exact ou non* et *ordonné*, alors que dans (Termier *et al.*, 2002), un document XML peut être représenté par plusieurs sous-arbres fréquents (un document peut apparaitre dans plusieurs clusters), qui peuvent être *exacts ou non* et *ordonnés ou non*.

– Le clustering de (Termier *et al*, 2002) est basé sur le calcul du sous-arbre fréquent maximal commun aux documents XML de chaque cluster. (Nierman *et al.*, 2002), (Francesca *et al.*, 2003), (Costa *et al.*, 2004) et (Dalamagas et al., 2004), effectuent un clustering *hiérarchique*, en s'appuyant sur une mesure de *distance* (voir sous section 2.2.1). Notre approche effectue un clustering *incrémental* (voir algorithme 3.3.3), fondé sur un calcul de *similarité* (voir sous section 3.4).

–Nous avons adapté, le système de formules dévelop pé dans (Mong *et al*, 2002), mais dans notre cas, les calculs sont relativement allégés du fait que nous manipulons des arbres étiquetés simples. L'approche de (Mong *et al.*, 2002) établit un *clustering hiérarchique ascendant* sur des arbres représentant des DTDs. Les nœuds d'un arbre DTD dénotent souvent des expressio ns régulières dont le traitement n'est pas toujours trivial.

Mais, le caractère novateur de cette nouvelle approche par rapport à (Aïtelhadj *et al.*, 2006), concerne la « mobilité » des centroïdes. En effet, même si initialement à la création d'un cluster, c'est le p remier arbre qui se présente qui possède le statut de centroïde, il doit être remplacé, si c'est nécessaire, par un autre arbre plus représentatif. L'algorithme de clusterin g de la sous section 3.3.3 met bien l'accent sur ce point. En effet, la fonction *importance* qui mesure le *poids* de chaque

arbre du cluster dans cet algorithme, garantit la pertinence du nouveau centroïde calculé.

6. Conclusion et perspectives

Nous avons proposé une méthode de classification structurelle de documents XML. Pour cela, nous avons proposé une manière de représenter structurellement un document XML et un algorithme de clustering basé sur une mesure de similarité structurelle entre documents XML. La classification de telles structures équivaut à la classification structurelle des documents XML qu'elles représentent. Les clusters sont créés au fur et à mesure que les documents se présentent. Notre modèle touche à deux aspects fondamentaux intéressants de la RI. En effet, d'une part, la classification permet de réduire le nombre de documents traités et du coup, augmenter le nombre de documents pertinents qui peuvent être retournés par un moteur de recherche. D'autre part, les clusters constituent un index permettant aux utilisateurs d'accéder aux groupes de documents XML qu'ils souhaitent interroger et atteindre les "unités d'information" spécifiques qui les intéressent.

L'expérimentation menée n'est qu'une petite esquisse qui permet de tester la faisabilité et quelque peu la fiabilité de l'approche proposée. Cependant, pour un passage à l'échelle, il est judicieux d'établir des tests sur de plus grandes collections de documents avec des formules de calcul de la *précision* plus appropriées telles que la pureté, l'entropie et autres.

En perspective, on peut rebâtir ou étendre cette classification en y adjoignant la composante correspondant aux contenus des documents, afin que le système de recherche puisse traiter ultérieurement des requêtes constituées de combinaisons de mots appartenant aux vocabulaires de la structure et/ou du contenu.

7. Bibliographie

Aïtelhadj A., Boughanem M., « CSDX: Classification Structurelle de documents XML », *VSST'2006*, Télécom Lille1, 16-17 Janvier 2006.

Costa G., Manco G., Ortale R., Tagarelli. A., « A Tree-Based Approach to Clustering XML Documents by Structure », *In PKDD*, 2004, pp 137-148.

Dalamagas T., Cheng T., Winkel K.-J. Sellis T. K., « Clustering XML Documents Using Structural Summaries », *In EDBT Workshops*, 2004, pp 547-556.

Del Razo Lopez F., Laurent A., Poncelet P., Teisseire M., « Recherche de sous-structures fréquentes pour l'intégration de schémas XML », *In Conférence Extraction et Gestion des Connaissances (EGC 2006)*, Lille, Janvier 2006, volume II, p 487-498.

Denoyer L., Apprentissage et Inférence statistique dans les bases de documents structurés: Application aux corpus de documents textuels, Thèse de Doctorat Paris 6, 2004.

Doan A., Domingos P., Halevy A Y., « Reconciling schemas of disparate data source : a machine-Learning approach », *In proceeding of ACM SIGMOD international conference on Management data*, ACM Press 2001, p 509-520.

Francesca F. D., Gordano G., Ortale R., Tagarelli A « Distance-based Clustering of XML Documents », *In Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, 2003, pp 75-78.

Madhavan J., Bernstein P A., Rahm E., « Generic Schema Matching with Cupid », *VLDB* 2001.

Mong L., Huai L., Hsu W. et Yang X., « XClust: Clustering XML Schemas for Effective Integration », *School of Computing, National University of Singapore, In proceedings of 11th CIKM*, 2002, p 292-299.

Nierman A., Jagadish H. V., « Evaluating Structural Similarity in XML Documents », *In Proceedings of the Fifth International Workshop on the Web and Databases*, WebDB, 2002, Madison,Wisconsin, USA.

Su H., Padmanabhan S., Lo M., « Identification of Syntactically Similar DTD Elements in Schema Matching across DTDs », *WAIM*, 2001.

Termier A., Rousset M.C. et Sebag., « Treefinder : a First Step towards XML Data Mining », *In Proceedings of ICMD* 2002, p 450-457.

Utilisation des liens entre documents structurés pour la recherche d'information

Philippe Mulhem, Delphine Verbyst

Laboratoire LIG-CNRS
Équipe MRIM - Bât B
Domaine Universitaire
385 rue de la Bibliothèque
F-38400 Saint Martin d'Hères
Philippe.Mulhem@imag.fr, Delphine.Verbyst@imag.fr

RÉSUMÉ. *Nous proposons dans cet article une approche pour rechercher des documents structurés qui intègre les liens existants entre les parties de documents ainsi que la composition structurelle des documents. Les liens entre les parties de documents sont caractérisés par des notions d'exhaustivité et de spécificité relatives, utilisées pour définir la valeur de pertinence des parties de documents. Nous proposons une approche par fonction de correspondance stratifiée pour utiliser ces éléments lors de la recherche de documents. Les expérimentations reportées ici portent sur le corpus de la compétition INEX 2008. Nos résultats sur la campagne d'évaluation nous placent en cinquième position sur 61 résultats officiels pour la tâche de recherche focalisée (Focused).*

ABSTRACT. *We present in this paper an approach to retrieve structured documents that uses non structural relations between document elements in conjunction with document/doxel structural relationships. We characterize the non structural relations by relative exhaustivity and specificity scores. We propose to express stratified matching functions to use these elements during document retrieval. Results of experiments on the INEX 2008 test collection are presented. Our best run is in the top 5 (among 61) official results for the Focused Task at INEX 2008.*

MOTS-CLÉS: *documents structurés, liens non compositionnels, INEX.*

KEYWORDS: *structured documents, non compositional links, INEX.*

1. Introduction

Ce document décrit une proposition de modélisation pour la recherche de documents structurés, et des expérimentations sur le corpus de la compétition INEX 2008[1].

Notre objectif ici est de montrer que l'utilisation de liens structurels (la composition initiale des parties de documents) et de liens non structurels conduit à des résultats de bonne qualité pour un système de recherche de documents XML. Dans la suite, une partie d'un document structuré sera appelée *doxel*, et les liens non structurels sortant d'un doxel suivant la relation r sont appelés environnement non structurel du doxel selon r. Une hypothèse forte de notre approche est que les doxels ne sont pas seulement pertinents en raison de leur contenu, mais aussi parce qu'ils sont liés à d'autres doxels pertinents. D'une certaine manière, nous revenons dans ce travail sur l'hypothèse de regroupement (*Cluster Hypothesis*) de van Rijsbergen [van Rijsbergen 1979], en considérant que la pertinence d'un doxel est affectée par la pertinence de ses doxels connexes.
Afin de tirer profit des relations entre doxels, nous les caractérisons en utilisant des mesures d'exhaustivité et de spécificité relatives, calculées à l'indexation.

Nous considérons que le traitement des liens entre doxels d'une manière appropriée peut aider un système de recherche d'information à fournir de meilleurs résultats. Pour utiliser les différentes informations associées aux documents structurés, nous proposons une correspondance entre documents et requêtes en plusieurs étapes, appelées strates. L'intérêt de ces strates est de permettre explicitement une prise en compte de ces informations de manière séquentielle, permettant d'accorder davantage d'importance à certaines informations sur les documents structurés.

Cet article est organisé de la manière suivante. Nous décrivons un état de l'art sur la recherche de documents structurés en partie 2, en nous intéressant à l'utilisation de la structure et des relations non structurelles. Cette partie nous permet également de poser les bases de la correspondance stratifiée entre documents et requêtes. Nous explicitons le modèle de documents structurés proposé dans la partie 3. La section 4 décrit la caractérisation des relations entre doxels. La section 5 présente le processus de correspondance stratifiée que nous utilisons. Les résultats obtenus sur le corpus INEX 2008 pour la piste (*track*) « ad hoc » [Kamp et al. 2008] sont présentés dans la section 6, et nous concluons en partie 7.

2. Etat de l'art

A la suite des travaux précurseurs de ceux de Wilkinson en 1994 [Wilkinson 1994], le début des années 2000, avec l'avènement du format XML a vu de nombreux

[1] http://www.inex.otago.ac.nz/

travaux de recherche se portent sur ce sujet. Conjointement, l'émergence de campagnes d'évaluations sur les documents structurés, en particulier INEX depuis 2002 a favorisé ce mouvement.

Si l'on se concentre sur des travaux basés sur les liens structurels des documents, on distingue les approches qui les utilisent à l'indexation de celles qui les utilisent lors du traitement des requêtes. Parmi les approches qui s'appliquent à l'indexation, considérons par exemple les travaux de Cui et Wen [Cui & Wen 2003] : ils propagent les termes d'indexation des feuilles des documents vers les racines, en élaguant les termes dans toutes les feuilles d'un doxel, pour les termes qui représentent chaque composant de ce doxel. L'avantage d'une telle approche est de réduire la taille de l'index, car tous les termes décrivant un doxel et chacun de ses composés ne sont plus stockés, mais que les doxels sont caractérisés transitivement.
La difficulté de tels travaux est de réaliser des élagages à la fois nombreux et de qualité pour la recherche. Les travaux de Lalmas [Lalmas &Vannoorenberghe 2004], dans le cadre de l'utilisation de la théorie de Dempster-Shafer pour l'indexation et la recherche de documents structurés, indexent des doxels en fonction des termes qui indexent leurs composants, il s'agit donc ici également d'une propagation de termes à l'indexation.
Les recherches qui ont trait à la prise en compte de la structure des documents lors du traitement de requêtes sont nombreux, on peut en particulier citer les travaux de Zargayouna [Zargayouna 2004] qui propagent les valeurs de pertinence, ou bien les travaux de [Huang et al. 2007] qui utilisent des probabilités *a priori* des doxels en fonction de leur position et de leur profondeur dans le document. L'approche de Wilkinson entre dans cette catégorie. Les systèmes à base de réseaux probabilistes comme [Myaeng et al . 1998] ou [Piwowarski &Gallinari 2005] utilisent également les relations structurelles entre les doxels à l'indexation.
Il est important de signaler que propager au moment du traitement de requête est moins complexe à réaliser que d'indexer en prenant en compte la structure des documents, car on peut facilement réutiliser des traitements ou des systèmes existants. Notre approche se situe dans la lignée de l'utilisation de la structure lors de la recherche des doxels.

Si nous nous intéressons maintenant à des travaux qui utilisent des liens non structurels entre les documents comme des liens de navigation, il existe également de nombreux travaux. Pour des pages HTML du Web, les approches basées sur des valeurs de popularité comme Pagerank [Brin et Page 1998] sont reconnues comme étant efficaces. Ces approches traitent indépendamment la pertinence des pages et la popularité. Les travaux de Smucker et Allan [Smucker &Allan 2008] sur les pages web, et ceux de Savoy [Savoy 1996] sur des articles scientifiques ont montré par ailleurs que les liens non structurels entre les documents peuvent être utiles pour la recherche d'information. Nous étudions ici l'utilisation de tels liens dans le cas de documents structurés, en nous intéressant à leur caractérisation pour la propagation de valeurs de pertinence.

Lors du traitement de requêtes, on se trouve face à des questions sur l'ordre des opérations à effectuer pour retrouver les documents. En RI classique par exemple, une requête posée à la fois sur le contenu et sur les attributs externes du document est traitée en effectuant la recherche d'information sur tous les documents puis en filtrant les réponses d'après les critères externes. Le même principe est utilisé dans la recherche de documents structurés quand la requête porte sur le contenu et sur la structure des résultats ; dans ce cas on peut rechercher sur tous les doxels d'après le critère de contenu puis filtrer sur le type recherché [Krumpholz & Hawking 2007]. Nous caractérisons ces démarches par le terme « recherche stratifiée », pour laquelle différentes étapes se succèdent afin d'obtenir le résultat escompté. Nous proposons ici un premier pas pour expliciter de telles stratifications pour la recherche de documents structurés en tenant compte de la structure des documents ainsi que des relations entre les doxels.

3. Modèle de documents structurés

Pour des raisons de place, nous décrivons succinctement dans cette partie le modèle de documents structurés proposé. Pour favoriser la compréhension, nous nous intéressons ici davantage aux relations entre les doxels, et moins à la description des doxels eux-mêmes, tout en posant que ces doxels sont associés à une représentation de leur contenu.

Nous définissons l'ensemble des doxels du corpus de documents structurés par *Dox*. La relation de composition *comp*, représentée par un ensemble C_{comp} entre deux doxels d1 et d2 (d1 composé directement par d2) est incluse dans $Dox \times Dox$. Elle est non réflexive et non transitive. La relation de composition découle de la structure logique du document. Toute autre relation, que nous qualifions de non compositionnelle, *r* est représentée par un ensemble de couples C_r, inclus dans $Dox \times Dox$, les contraintes éventuelles sur r étant spécifiques à leur sémantique. Par exemple, des relations de navigation entre doxels ne possèdent pas de contrainte particulière. Pour un doxel d_1 quelconque, on appelle environnement de d_1 pour la relation r, noté $Env_r(d_1)$, l'ensemble de doxels de la collection vers lesquels pointe d_1 suivant r. Les relations non compositionnelles peuvent provenir du corpus initial ou bien être générées *a posteriori* par le système de recherche d'information.

Nous décrivons un exemple de ces éléments sur la figure 1. Dans cette figure, l'ensemble Dox = {d1, d2, d4, d5, d6, d7, d8, d9, d10, d11, d12}, C_{comp} est égal à {(d1, d2), (d1, d3), (d1, d4), (d2, d5), (d2, d6), (d3, d7), (d3, d8), (d9, d10), (d9, d11), (d10, d12), (d11, d13), (d11, d14)}. Nous y présentons également deux relations, r1 et r2, la première représentant un lien de navigation tiré de la structure des documents initiaux avec C_{r1} = {(d3, d10), (d11, d1)}, et la seconde étant par exemple une relation générée à partir de la similarité entre les doxels avec C_{r2} = {(d12, d7)}, indiquant que le contenu du doxel source est très similaire à celui du doxel cible.

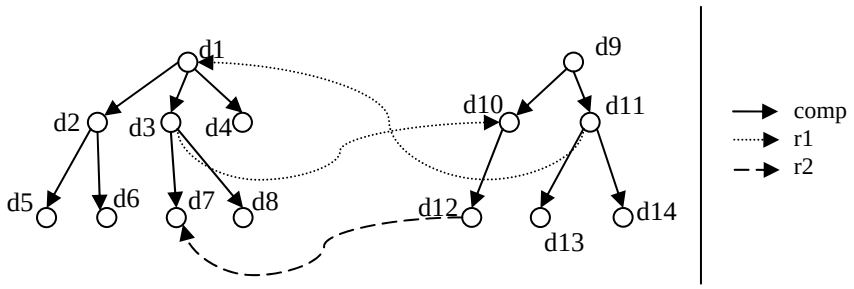


Figure 1 –Un exemple de corpus avec une relation de composition *comp* et deux relations non compositionnelles *r1* et *r2*.

La modélisation proposée permet de bien représenter les différentes relations entre doxels, afin de les caractériser et de les utiliser lors du traitement de requêtes.

4. Caractérisation des relations entre doxels

Plutôt que de se limiter au fait que des relations entre doxels existent, il nous a semblé préférable de se poser la question de décrire finement l'intérêt potentiel de ces liens en terme d'exploration de l'es pace résultat en supposant que l'utilisateur les suive. En s'inspirant des travaux sur l'évaluation de systèmes de recherche de documents structurés [Piwowarski & Lalmas 2004] et plus anciennement sur [Chiaramella et al. 1996], nous décidons d'étudier les caractérisations liées au fait que le doxel cible d'un lien est plus exhaus tif ou plus spécifique que le doxel source de ce lien. De manière plus précise, s'il existe une relation r du doxel d1 vers le doxel d2 :
- l'exhaustivité relative du lien (d1, d2), notée Exh(d1, d2), dénote le fait que d2 traite de tous les sujets de d1. Nous fixons cette valeur dans l'intervalle [0, 1], avec une valeur proche de 1 si d2 traite de tous les sujets de d1, et une valeur proche de 0 si d2 ne traite que de peu de sujets de d1 ;
- la spécificité relative du lien (d1, d2), notée Spe(d1, d2), dénote le fait que d2 ne traite que des sujets de d1. Nous fixons cette valeur dans l'intervalle [0, 1], avec une valeur proche de 1 si d2 ne traite que des sujets de d1, et une valeur proche de 0 si d2 traite de nombreux autres sujets que ceux de d1.
Ces mesures ne sont pas nécessairement corrélées, car un document d2 peut traiter de tous les sujets de d1 et d'aucun autre sujet, et donc être caractérisé par une exhaustivité et une spécificité proches de 1 ; le document d2 peut également traiter de tous les sujets de d1 et de beaucoup d'autre sujets, et dans ce cas l'exhaustivité du lien est proche de 1 et sa spécificité proche de 0.

Une fois les propriétés de ces mesures définies, nous décrivons maintenant des formules qui permettent de les calculer. Pour cela, nous nous inspirons de la fonction de recouvrement (*overlap*) définie dans [Salton & McGill 1983] sur des

ensembles. Cette fonction de recouvrement, symétrique, a pour valeur la taille de l'intersection de deux ensembles divisée par la taille du plus petit des ensembles. Nous devons l'adapter pour les raisons suivantes :

- nous prenons en compte les représentations de doxels qui ne sont pas des ensembles de termes mais des vecteurs pondérés, ces représentations étant plus adaptées à la recherche d'information,
- les fonctions que nous définissons ne doivent pas être symétriques : l'exhaustivité de d1 vers d2 n'a pas la même valeur que l'exhaustivité de d2 vers d1, et de même pour la spécificité relative.

On peut cependant remarquer que, d'après la définition que nous avons faite des exhaustivités et spécificités relatives, il est raisonnable de proposer que Exh(d1, d2) soit égal à Spe(d2, d1).

Supposons que la représentation du contenu d'un doxel d_i est exprimée par un vecteur $(w_{i,1}, ..., w_{i,n})$ de poids correspondant à des pondérations *tf.idf*, avec n la taille du vocabulaire. Nous définissons les formules suivantes pour calculer les valeurs d'exhaustivité et de spécificité relatives pour un lien d'un doxel d1 vers un doxel d2 :

$$Exh(d_1, d_2) = \frac{\sum_{i \in [1,n] | w_{2,i} \neq 0} w_{1,i}^2}{\sum_{i \in [1,n]} w_{1,i}^2} \quad \text{et} \quad Spe(d_1, d_2) = \frac{\sum_{i \in [1,n] | w_{1,i} \neq 0} w_{2,i}^2}{\sum_{i \in [1,n]} w_{2,i}^2}$$

L'utilisation de carrés dans ces formules a pour rôle d'accorder davantage d'importance aux grandes valeurs qu'aux faibles. Ces mesures sont bien comprises dans l'intervalle [0, 1] si les doxels sont décrits par des vecteurs non nuls, ce qui est une hypothèse valide car un doxel sans contenu est habituellement considéré comme non pertinent quelque soit la requête.

5.Recherche stratifiée de doxels

Quand nous calculons la valeur de pertinence d'un doxel, nous voulons prendre en compte le contenu propre du doxel ainsi que son environnement. De plus, il est possible que la fonction de correspondance doive être stratifiée pour privilégier certaines relations lors de la recherche : nous proposons cette stratification à la suite de résultats expérimentaux obtenus pour la compétition d'Inex 2007 [Fuhr et al. 2008] pour lesquels il a été montré que traiter d'abord les documents complets et ensuite leurs doxels donne de bons résultats. Si un processus de correspondance est composé de deux strates s1 puis s2, les doxels résultats sont ordonnés tout d'abord sur les valeurs de pertinence obtenues par la strate s1, puis ensuite pour les résultats de même valeur pour la strate s1 par les valeurs de la strate s2. Il en résulte que la strate s1 possède une importance primordiale pour le processus de recherche. Nous nous limitons ici à des strates séquentielles s'enchaînant linéairement, chacune décrite par une fonction de correspondance. La première fonction de la liste est la première strate, etc. L'intérêt de l'utilisation de telles strates est qu'il est possible de

prendre en compte des fonctions de correspondance différentes à chaque strate, ce que nous avons testé lors de nos expérimentations.
Si le processus de recherche est décrit par une seule strate, on se retrouve dans le cas de la recherche d'information sur le conte nu classique. Par exemple, si on ne calcule qu'un cosinus entre une requête q et un doxel d et que le résultat est trié sur ce cosinus, la description du processus de correspondance est [$RSV_{\cos}$] avec $RSV_{\cos}(d,q) = \cos(\vec{d},\vec{q})$, avec la flèche dénotant le vecteur représentant le contenu de l'élément considéré.

Si nous prenons le cas d'un corpus de doxels sur lequel on calcule la correspondance sur le contenu des doxels et ensuite sur leur environnement suivant une relation r (comme utilisé dans [Savoy 1993] avec des liens de référence par exemple), la fonction de correspondance stratifiée peut être définie par la liste [$RSV_{\cos}$, RSV_{env_savoy}], avec $RSV_{\cos}$ décrite ci-dessus, et

$$RSV_{env\_savoy}(d,q) = \frac{1}{|Env_r(d)|} \sum_{d' \in Env_r(d)} \cos(d',q)$$

Dans les expérimentations que nous avons menées (cf. partie 6.2), nous avons utilisé une stratification avec une première strate sur les documents complets des doxels, et une seconde intégrant le contenu des doxels et leur environnement non structurel par un calcul avec la fonction de correspondance RSV_{env_link}, qui utilise les exhaustivité et spécificité relatives sur une relation *link* (cf. section 6.1) entre doxels, définie par :

$$RSV_{env-link}(d,q) = \alpha . RSV_{\cos}(d,q) + (1-\alpha) . \frac{1}{|env_{link}(d)|} . \sum_{d' \in env_{link}(d)} (\beta\, Exh(d,d') + (1-\beta) Spe(d,d')) RSV_{\cos}(d',q)$$

Cette fonction de correspondance, tirée de travaux précédents [Verbyst & Mulhem 2008], utilise une combinaison linéaire de la correspondance entre le contenu du doxel propre et de la requête et la correspondance des doxels ciblés par le doxel source, ce calcul intégrant les valeurs d'exhaustivité et de spécificité relatives entre les doxels. La valeur α dénote l'importandce relative du contenu propre par rapport à l'environnement du doxel, et la valeur β dénote l'importance de l'exhaustivité par rapport à la spécificité pour la prise en compte de l'environnement.

6. Expérimentations

Les expérimentations que nous présentons ont été menées dans le cadre de la campagne INEX 2008. La collection de test contient 658 000 documents en anglais extraits de wikipedia, et un total de 285 requêtes. De cet ensemble de requêtes, 70

ont été choisies *a posteriori* par les organisateurs comme base d'évaluation des systèmes. Les évaluations officielles de la tâche Ad hoc d'INEX sont fortement inspirées des courbes de rappel/précision [Kamps et al. 2008], mais au lieu de se baser sur des nombres de documents pertinents et/ou retrouvés elles utilisent des proportions basées sur la taille des parties de documents retrouvés et/ou petinents : le ratio de rappel devient le nombre de caractères pertinents dans les doxels renvoyés divisé par le nombre total de caractères pertinents, et la précision est le ratio de caractères pertinents renvoyés sur le nombre de caractères renvoyés (i.e., le nombre de caractères dans les doxels renvoyés par le système).

6.1. Représentation du corpus

Par rapport aux 110 millions de doxels de la collection INEX 2008, nous nous sommes limités à ceux correspondant aux marqueurs *article*, *title*, *section*, *paragraph*, *item* et *collectionlink*, ce qui donne un nombre de 29 millions de doxels. Sur ces doxels, nous avons défini une relation *doc_comp* qui permet pour un doxel quelconque (non document) de pointer sur le document qu'il compose. Cette relation est en fait créée à partir de la fermeture transitive de la composition structurelle des doxels. La relation *doc_comp* contient 28,5 millions de liens, et elle est non réflexive et non transitive, et pour chaque doxel non document elle ne relie qu'un et un seul document. La figure 2 donne un exemple de document tiré de la collection Inex 2008, on voit dans cette figure que 3 doxels (marqueurs soulignés) de documents sont indexées : l'*article* complet et deux *collectionlink*s. Le premier des *collectionlink*s considérés, correspondant à *French*, pointe sur un autre document de la collection, 10581.xml, dont l'identifiant est 10581.

```
<article>
<name id="288042">Croquembouche</name>
...
<body>A
<emph3>croquembouche</emph3>is a
<collectionlink ... xlink:href="10581.xml">French</collectionlink>
<collectionlink ... xlink:href="57572.xml">cake</collectionlink>
consisting of a conical heap of cream-filled
...
</body>
</article>
```

Figure 2 – Un exemple de document tiré de la collection Inex 2008.

Les 17 millions de marqueurs *collectionlink* dénotent des liens entre des doxels et des documents complets. Afin d'avoir des relations plus précises entre doxels, nous avons choisi de définir la relation *link* comme étant composée des liens collectionlinks étendus par les doxels des documents cibles les plus similaires au doxel contenant la source du lien. La relation *link* obtenue contient 115 millions de liens entre doxels, soit en moyenne 4 liens par doxel. La relation *link* n'a pas de

caractéristique particulière définie a priori. Pour chacun de ces liens, nous avons calculé les valeurs d'exhaustivité et de spécificité relatives en nous basant sur une représentation des doxels par des vecteurs pondérés à base de tf.idf, avec l'idf défini sur les documents complets.

6.2. Fonctions de correspondance

Dans cette partie, nous décrivons les fonctions de correspondance stratifiées entre documents et requêtes que nous avons expérimentées.

Tout d'abord, nous proposons de définir deux fonctions de correspondance stratifiées, qui en première étape se basent sur une recherche sur des documents complets des doxels par l'environnement *doc_comp*, et dans une seconde étape utilise une fonction de correspondance sur le contenu des doxels avec leur environnement *link*.
La première, appelée $C_{[doc\text{-}cos,env]}$, est définie par la liste $[RSV_{Doc\text{-}cos}, RSV_{env\text{-}link}]$:

- $RSV_{Doc-\cos}(d,q) = \cos(\vec{D},\vec{q}), \text{avec } D = \text{elem}(\text{env}_{doc\_comp}(d))$
  En posant que la fonction *elem* renvoie l'élément d'un singleton.
- Et $RSV_{env\text{-}link}$ définie précédemment en partie 5.

La seconde correspondance stratifiée, appelée $C_{[Doc\text{-}lm,env]}$, utilise quant à elle une valeur de pertinence pour le contenu des documents complets basée sur un modèle de langue unigramme utilisant un lissage de Dirichlet. Nous avons choisi d'utiliser cette correspondance car il a été montré lors de la campagne INEX 2007 [Kamps et al 2007] que ce modèle donne de bons résultats pour les documents complets. Elle est décrite par $[RSV_{Doc\text{-}LM}, RSV_{env\text{-}link}]$, et dans ce cas la première étape trie les documents en utilisant le modèle de langue, puis trie les doxels pertinents par les fonctions décrites précédemment.

Nous comparons les résultats ci-dessus à trois correspondances à une strate afin de déterminer les apports de chacune des strates utilisées au dessus :
- $C_{[doc\text{-}lm]}$ décrite par $[RSV_{Doc\text{-}LM}]$. Cette fonction est la première de $C_{[Doc\text{-}lm,env]}$. La correspondance $C_{[doc\text{-}lm]}$ renvoie en fait la même valeur de pertinence pour tous les doxels d'un même document, nous choisissons dans la réponse de ne fournir que les documents complets en réponse.
- $C_{[env]}$ avec $[RSV_{env\text{-}link}]$. Cette fonction de correspondance est en fait la seconde utilisée dans $C_{[doc\text{-}cos,env]}$ et $C_{[doc\text{-}lm,env]}$. Elle nous permet de déterminer comment se comporte la recherche en utilisant uniquement le contexte des doxels comme leur environnement de liens *link*.
- et enfin $C_{[no\text{-}env]}$ décrite par $[RSV_{cos}]$. Cette fonction de correspondance permet de déterminer dans quelle mesure les liens apportent un plus par rapport à une correspondance basée uniquement sur les contenus propres des doxels.

Le tableau 1 résume les différentes correspondances utilisées lors des expérimentations.

Correpondance	Strate 1	Strate 2 (si elle existe)
$C_{[doc\text{-}cos,env]}$	$RSV_{Doc\text{-}cos}$	$RSV_{env\text{-}link}$
$C_{[Doc\text{-}lm,env]}$	$RSV_{Doc\text{-}LM}$	$RSV_{env\text{-}link}$
$C_{[doc\text{-}lm]}$	$RSV_{Doc\text{-}LM}$	
$C_{[env]}$	$RSV_{env\text{-}link}$	
$C_{[no\text{-}env]}$	RSV_{cos}	

Tableau 1. Les différentes correspondances expérimentées.

6.3. Résultats expérimentaux

Nous décrivons ici les résultats obtenus par notre approche. Ces résultats sont pour une part des résultats officiels soumis à la compétition INEX 2008, et pour une part des résultats obtenus postérieurement afin de bien estimer l'impact des différents paramètres utilisés.

Nous présentons dans le tableau 2 les résultats obtenus avec les 5 correspondances issues de la l'utilisation de la relation *doc_comp* et de la relation *link*, utilisée avec 4 voisins, une valeur $\alpha = 0.5$ et une valeur $\beta = 0$. Ces valeurs ont été choisies après des tests effectués sur le corpus d'INEX 2007. Le tableau 2 présente les résultats en terme de précision à un taux de rappel de 0,00, 0,01, 0,05, 0,1 ainsi que la valeur de précision moyenne pour les correspondances $C_{[doc\text{-}cos,env]}$, $C_{[doc\text{-}lm,env]}$, $C_{[doc\text{-}lm]}$ et $C_{[env]}$. Rappelons que la mesure officielle d'INEX 2008 pour comparer les systèmes est la précision moyenne obtenue au taux de rappel 0.01 ; cette mesure favorise les systèmes qui fournissent de bons résultats initiaux.

Intéressons-nous aux deux dernières approches monostrates de la deuxième partie du tableau 2, $C_{[env]}$ et $C_{[no\text{-}env]}$. Nous constatons que l'approche sans environnement fournit des résultats comparables à l'approche avec environnement *link* pour iP[0,00] avec +2,2%, mais ensuite pour les valeurs iP[0,01], iP[0,05] et iP[0,10] l'utilisation de l'environnement renvoie da vantage de documents pertinents, avec respectivement +17,0%, +21,5% et +21.7% . Ces différences sont très importantes et ne nécessitent par de test de significativité statistique. Par ailleurs, le taux de précision moyenne entre ces deux configurations est de +10% en faveur de la recherche des doxels avec leur environnement *link*. On en conclut qu'utiliser l'environnement *link* tel que nous l'avons défini est profitable sur ce corpus.
Si l'on regarde les deux premières lignes du tableau 2 (résultats avec deux strates, obtenus officiellement à la compétition INEX 2008), on remarque que les résultats de l'approche ayant pour première strate le modèle langue obtiennent des résultats très supérieurs à ceux utilisant le cosinus pour les documents, et ceci pour iP[0,00], iP[0,01], iP[0,05] et iP[0,10], avec respectivement +28,1%, +28.5%, +18,4% et +12,1% . Le taux de précision moyenne entre ces deux configurations est de +7,6% en faveur de $C_{[doc\text{-}lm,env]}$. Les seules différences entre ces deux configurations étant la

première strate, on conclut qu'utiliser les correspondances sur les documents en se basant sur un modèle de langue est pertinent.
Nous comparons enfin notre meilleur résultat étudié jusqu'à présent, $C_{[doc\text{-}lm,env]}$, avec une configuration monostrate $C_{[doc\text{-}lm]}$, afin de vérifier que l'utilisation des deux strates est profitable. L'approche à deux strates donne des résultats meilleurs pour iP[0.00] et iP[0.01], avec respectivement +17,0% et +9,7%. Par contre ensuite, l'approche $C_{[doc\text{-}lm,env]}$ donne de plus mauvais résultats que $C_{[doc\text{-}lm]}$, avec -7,3% pour Ip[0.05] et -19,1% pour Ip[0.10]. Comme l'approche monostrate $C_{[doc\text{-}lm]}$ retourne des documents complets, la valeur de précision moyenne de $C_{[doc\text{-}lm,env]}$ lui est très inférieure : -51,7%. On constate donc que l'approche avec deux strates est donc meilleure pour les premiers résultats, mais que, pour favoriser le rappel, des approches avec documents complets sont meilleures, sur cette collection.

Notons par ailleurs que la configuration $C_{[doc\text{-}lm,env]}$ (deuxième ligne du tableau 1), est arrivée en 5ème position (et troisième site) lors de la compétition INEX 2008 [Kamps et al. 2008], et premier résultat français sur 61 résultats officiels validés.

Run	iP[0,00]	iP[0,01]	iP[0,05]	iP[0,10]	MaP
$C_{[doc\text{-}cos,env]}$	**0.5555**	**0.5187**	**0.4402**	**0.3762**	**0.1441**
$C_{[doc\text{-}lm,env]}$	**0.7114**	**0.6665**	**0.5210**	**0.4216**	**0.1339**
$C_{[doc\text{-}lm]}$	0.6078	0.6077	0.5623	0.5211	0.2771
$C_{[env]}$	0.4595	0.4035	0.3592	0.2942	0.0906
$C_{[no\text{-}env]}$	0.4495	0.3449	0.2957	0.2417	0.0824

Tableau 2. Résultats sur le corpus INEX 2008 (résultats officiels en gras)

7. Conclusion

Nous avons présenté dans cet article une approche pour indexer et retrouver des documents structurés, en prenant en compte la structure des documents ainsi que des relations non-structurelles entre les doxels. Au niveau de l'indexation, nous proposons de caractériser les relations non structurelles entre les doxels en définissant des valeurs de spécificité et d'exhaustivité relatives. Le traitement de requêtes que nous proposons repose sur une stratification, qui permet d'enchaîner des traitements partiels pour obtenir un ordonnancement des réponses dépendant des strates définies. Nous avons mené des expérimentations sur la collection de test d'INEX 2008. Les résultats que nous obtenons montrent que l'utilisation d'informations sur les documents complets qui contiennent des doxels, ainsi que sur les relations non structurelles qui existent entre les doxels, donnent de meilleurs résultats que l'utilisation indépendante de l'une ou l'autre de ces approches. En particulier, un de nos résultats a permis d'obtenir de très bons résultats officiels à la compétition INEX 2008.

Les travaux futurs que nous allons effectuer sur ce sujet ont trait à une étude plus détaillée de la notion de strates multiples pour effectuer la recherche de documents structurés inter-reliés. En particulier, nous allons formaliser davantage cette notion de strates afin de contrôler davantage ce principe. La définition des formules proposées pour le calcul des exhaustivité et spécificité relatives est également un sujet qui doit être approfondi, en particulier si le modèle utilisé pour représenter les contenus de document n'est plus un modèle vectoriel. A un niveau expérimental, nous allons étudier l'utilisation d'autres modèles de recherche d'information que le vectoriel pour les doxels, car il a été montré lors de la compétition INEX 2008 que l'utilisation de la formule du BM25 proposée par Robertson [Robertson et al. 1994] donne de très bons résultats.

Bibliographie

S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998.

Y. Chiaramella, F. Fourel and P. Mulhem. Modelling Multimedia Structured Documents. Technical report, FERMI ESPRIT BRA 8134, University of Glasgow, 1996.

H. Cui and J.-R. Wen. Hierarchical indexing and flexible element retrieval for structured documents. In 25th European Conference on Information Retrieval Research (ECIR'03), pp. 9-36, 2003.

N. Fuhr, J. Kamps, M. Lalmas, S. Malik, A. Trotman. Overview of the INEX 2007 Ad Hoc Track. Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2007), pp. 1-23, 2008.

F. Huang, S. Watt, D. Harper, M. Clark. Compact Representation in XML Retrieval. In Comparative Evaluation of XML Information Retrieval Systems (INEX 2006), pp. 65-72, 2007.

J. Kamps, S. Geva, A. Trotman, A. Woodley, M. Koolen. Overview of the INEX 2008 Ad Hoc Track. INEX 2008 Workshop Preproceedings. http://www.inex.otago.ac.nz/data/proceedings/INEX2008-preproceedings.pdf

A. Krumpholz and D. Hawking, CSRIRO's Participation in INEX 2006. In Comparative Evaluation of XML Information Retrieval Systems (INEX 2006), pp. 73-81, 2007.

M. Lalmas and P. Vannoorenberghe. Modelling XML retrieval with belief functions. CORIA 04. pp. 143-160, 2004.

S.-H. Myaeng, D.-H. Jang, M.-S. Kim, Z-C. Zhoo. A Flexible Model for Retrieval of SGML Documents. SIGIR 1998. pp. 138-145, 1998.

B. Piwowarski and P.Gallinari. A Bayesian Framework for XML Information Retrieval: Searching and Learning with the INEX Collection. Information Retrieval, Vol. 8, N. 4, Springer Netherlands, décembre 2005.

B. Piwowarski and M. Lalmas. Interface pour l'évaluation de systèmes de recherche sur des documents XML. CORIA 2004, pp. 109-120, 2004.

S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford. Okapi at TREC-3. TREC 1994, 1994.

G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, New York, NY, 1983.

J. Savoy, An extended vector-processing scheme for searching information in hypertext systems. Information Processing and Management. Vol. 32, N. 2, pp. 155-170, 1996.

M. D. Smucker and J. Allan. Using Similarity Links as Shortcuts to Relevant Web Pages, SIGIR '07, Amsterdam, The Netherlands, pp. 863-864, 2007.

C. van Rijsbergen. Information Retrieval. Burreworth 1979.

D. Verbyst and P. Mulhem. Doxels in context for retrieval: from structure to neighbours. ACM SAC 2008, pp. 1122-1126, 2008.

R. Wilkinson. Effective Retrieval of Structured Documents. Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, pp. 311-317, 1994.

X. Yin , W. S. Lee, Using link analysis to improve layout on mobile devices, Proceedings of the 13th international conference on World Wide Web, pp. 338-344, 2004.

H. Zargayouna. Contexte et sémantique pour une indexation de documents semi-structurés. CORIA 04. pp. 161-177, 2004.

Impact précoce du poids des balises pour la recherche d'information ciblée

Mathias Géry, Christine Largeron, Franck Thollard

Université de Lyon, F-69003, Lyon, France
Université de Saint-Étienne, F-42000, Saint-Étienne, France
CNRS UMR5516, Laboratoire Hubert Curien
{mathias.gery, christine.largeron, franck.thollard}@univ-st-etienne.fr

RÉSUMÉ. *Cet article traite de l'intégration des balises XML dans la fonction de pondération des termes, pour la recherche d'information (RI) XML ciblée. Notre modèle permet de considérer un certain type d'information structurelle : les balises qui représentent la structure logique des documents (titre, section, paragraphe, etc.), ainsi que les balises liées à la mise en forme (gras, italique, centré, etc.). Nous prenons en compte l'influence des balises sous forme d'un poids en estimant la probabilité pour une balise de mettre en évidence les termes pertinents. Ensuite, ces poids sont intégrés à la fonction de pondération des termes. Des expérimentations sur une collection de grande taille dans le cadre de la compétition de RI XML, INEX 2008, ont montré une amélioration de la qualité des résultats en RI ciblée.*

ABSTRACT. *This paper addresses the integration of XML tags in terms weighting function for focused XML Information Retrieval (IR). Our model allows to consider a certain kind of structural information: tags that represent logical structure (title, section, paragraph, etc.) as well as tags related to formatting (bold, italic, center, etc.). We take into account the tags influence by estimating the probability that tags distinguishe relevant terms. Then, these weights are integrated in terms weighting function. Experiments on a large collection during INEX 2008 XML IR evaluation campaign showed improvements on focused retrieval.*

MOTS-CLÉS : *Modèle probabiliste de document, Recherche d'information structurée, XML, Balises, Pondération*

KEYWORDS: *Probabilistc IR model, Structured IR, XML, Tags, Weighting*

1. Introduction

La plupart des documents disponibles dans des bases textuelles ou sur Internet sont fortement structurés. C'est le cas par exemple pour les articles scientifiques ou pour les documents écrits à l'aide de langages de balises (HTML, XML). L'information fournie par la structure peut être utilisée pour mettre en exergue certains mots : un mot ne revêt pas la même importance s'il apparaît dans une fonte particulière (gras, italique, etc.). De la même manière, un mot est plus important s'il apparaît dans certaines parties de document (un titre, la légende d'une figure, etc.). Cependant, les modèles de Recherche d'Information (RI) classiques (modèles booléen, vectoriel, probabiliste), dans leur version de base, ne prennent pas en compte cette structure.

Un état de l'art est présenté dans la section suivante. La première contribution de ce papier[1] est la proposition d'un cadre formel, présenté dans les sections 3 et 4, prenant en compte explicitement la structure du document. La seconde contribution consiste en une expérimentation du modèle, présentée dans la section 5, sur une collection d'envergure (la collection INEX[2]).

2. État de l'art

La prise en compte de la structure peut être faite soit à l'étape d'indexation soit à l'étape d'interrogation. Nous distinguons 3 types d'approches :

Modèle de requête et structure : l'intégration de la structure à l'étape de l'interrogation peut se faire en adaptant le langage SQL de façon à autoriser des requêtes portant sur la structure du document (cf. [NAV 95], XIRQ [FUH 01]). En pratique, peu d'utilisateurs sont en fait capables de formuler leurs besoins par des requêtes complexes[3] : les besoins sont en général exprimés par quelques mots-clés.

Modèle de document, structure et poids des termes : la seconde approche explorée revisite les modèles classiques en proposant un schéma de pondération de la structure [FUL 93]. Le poids alors affecté à un mot ne dépend pas seulement de sa fréquence (dans le document et la collection) mais aussi de sa position dans le document, définie par rapport aux balises de structure logique et de structure de mise en forme. Le classement final ne dépend pas uniquement de la présence d'un terme dans un document mais de la présence d'un terme étiqueté de manière appropriée.

L'intégration des balises peut être traitée de manière ad-hoc lorsque seules des balises de structure logique sont utilisées. Dans ce cas, le document peut être scindé en autant d'éléments qu'il y a de parties dans le document initial. Chaque élément est alors traité comme un document. Différentes stratégies sont alors possibles pour combiner les scores des éléments pour former le score de celui qui les contient [WIL 94].

---

1. Ce travail a été soutenu par le projet "Web Intelligence" de la région Rhône-Alpes.
2. INitiative for Evaluation of XML Retrieval. See `http://www.inex.otago.ac.nz`
3. Exemple : "Je cherche un paragraphe qui traite de course à pied, contenu dans un article qui parle du marathon de New-York et qui contient une photo d'un marathonien"

Les balises utilisées pour déterminer la structure peuvent être sélectionnées empiriquement [RAP 01]. Afin d'intégrer la structure logique dans un modèle classique, [ROB 04] duplique les éléments autant de fois que les poids le suggèrent, ce qui permet de conserver la non linéarité de la fonction de pondération `BM25`. Les poids peuvent également être appris automatiquement [BOY 96, KIM 00, TRO 05].

Dans toutes ces approches cependant, les systèmes retournent au final des documents complets. Leurs capacités à mener une RI ciblée en retrouvant des parties de documents n'ont pas été évaluées.

Modèle de document, structure et poids des chemins

Une troisième approche consiste à s'appuyer sur une représentation arborescente de la structure des documents [SCH 02, TRO 05]. Chaque élément XML, correspondant à un nœud de l'arbre, est caractérisé par un chemin allant de la racine de l'arbre jusqu'à ce nœud. La structure est prise en compte au niveau d'un mot en considérant le chemin de l'élément qui le contient. Cette approche a été utilisée en considérant un nombre limité de balises [KOT 02, WIL 94, WOL 00]. Par exemple, dans [KOT 02] les termes qui apparaissent dans le titre `journal/article/title` ont un poids plus important que ceux situés dans le résumé `journal/article/abstract`. Le poids du terme est donc une combinaison de son poids calculé classiquement et du poids associé à sa position dans le document.

Dans [TRO 05], Trotman propose de prendre en compte la structure en assignant un poids estimé par algorithme génétique à chaque nœud XML (c'est-à-dire à la position du mot dans le document). Ce poids est combiné au tf dans différents schémas de pondération. Cette technique ne permet cependant pas d'améliorer les résultats de la fonction de pondération la plus performante (`BM25`).

Dans la plupart de ces approches, très peu de balises sont prises en compte (en général moins de 5), et leur sélection nécessite souvent une intervention manuelle.

Notre approche se caractérise par :

– La prise en compte de balises de structure logique et de mise en forme, comme il en existe dans les documents XML, en levant la limitation liée au nombre de balises prises en compte comme dans [ROB 04].

– Une étape d'apprentissage automatique pour estimer le poids de chaque balise, permettant d'évaluer son impact de manière générale et non relativement aux termes qu'elle étiquette. Les poids pouvant avoir un impact négatif, cette étape peut également être considérée comme une étape de sélection de balise.

– La RI ciblée : notre modèle vise à retourner à l'utilisateur des éléments XML de la granularité la plus adaptée possible, au contraire des approches qui visent à améliorer la recherche de documents complets [TRO 05].

– L'extension de la fonction de pondération `BM25` [ROB 76] via l'intégration du poids des balises.

3. Modélisation de la structure des documents

La structure des documents est intégrée dans notre modèle à deux niveaux. Notons que les balises de structure logique contribuent à ces deux niveaux :

1) La structure logique est utilisée pour déterminer la granularité de l'indexation et donc la granularité des éléments que le système sera susceptible de renvoyer. La pertinence n'est plus estimée au niveau du document complet, mais au niveau de parties de documents, par exemple des éléments XML.

2) La structure logique et la structure de mise en forme sont intégrées au niveau de la fonction de pondération des termes, par une étape d'apprentissage au cours de laquelle un poids est associé à chaque balise. Ce poids est basé sur la probabilité que la balise mette en exergue un terme pertinent ou au contraire un terme non pertinent[4].

À l'étape d'interrogation, la probabilité pour un élément d'être pertinent est estimée en combinant les poids des termes qu'il contient avec les poids des balises qui les étiquettent.

4. Un modèle probabiliste pour la représentation de documents structurés

Soit $\mathcal{D}$ un ensemble de documents structurés. Sans perte de généralité, nous considérerons des documents XML. Chaque élément (article, section, paragraphe, etc.) sera représenté par un ensemble de termes. Dans l'exemple suivant nous disposons de trois documents D_0, D_1 et D_2 :

D_0	D_1	D_2
<article>	<article>	<article>
$t_1 t_2 t_3$	<section>	<section>
<section>	$t_2 t_4$	t_5
$t_1 t_4$	$t_2 t_5$	$t_3 t_4$
$t_2 t_5$	</section>	$t_3 t_5$
</section>	$t_2 t_1$	</section>
</article>	</article>	</article>

Le document D_2 est indexé par cinq éléments : un *article* (balise <article>), une *section* (balise <section>) et trois *paragraphes* (balise).

Nous notons :

– $E = \{e_1, ..., e_j, ...e_l\}$, l'ensemble des éléments logiques disponibles dans la collection (*article*, *section*, *p*, etc.) ;

– $T = \{t_1, ..., t_i, ...t_n\}$, un index de termes construit à partir de E ;

– $B = \{b_1, ..., b_k, ..., b_m\}$, l'ensemble des balises.

4. Ceci rejoint les principes du modèle probabiliste [ROB 76] qui, à partir d'une collection de test dans laquelle la pertinence des documents est disponible, estime la probabilité qu'un terme donné apparaisse dans un document pertinent (resp. non pertinent).

Dans la suite, la représentation d'un élément e_j est notée x_j lorsque seuls les termes sont considérés et m_j lorsque à la fois les termes et les balises sont considérés.

4.1. *Score de pertinence d'un élément XML basé sur les termes*

La pertinence d'un élément relativement à une requête Q est fonction du poids des termes qui apparaissent dans l'élément et dans la requête. On note w_{ji} le poids du terme t_i dans l'élément x_j. On définit X_j un vecteur de variables aléatoires et $x_j = (x_{j0}, ..., x_{ji}, ..., x_{jn})$ une réalisation de ce vecteur X_j, avec $x_{ji} = 1$ (resp. 0) si le terme t_i apparaît (resp. n'apparaît pas) dans l'élément e_j.

Étant données ces notations, f_{term}, la pertinence de x_j basée sur les poids des termes, est donnée par le score :

$$f_{term}(x_j) = \sum_{t_i \in T \cap Q} x_{ji} \times w_{ji} \tag{1}$$

Sous ce produit scalaire général se cachent différentes fonctions comme par exemple `ltn`, `ltc` [SAL 83] ou `BM25` [ROB 76]. Des expérimentations antérieures [GéR 08] avec `ltn` et `ltc` ayant donné des résultats médiocres relativement à ceux obtenus avec `BM25`, nous ne considérerons par la suite que `BM25` :

$$w_{ji} = \frac{tf_{ji} \times (k_1 + 1)}{k_1 \times ((1 - b) + (b * ndl)) + tf_{ji}} \times \log \frac{N - df_i + 0.5}{df_i + 0.5} \tag{2}$$

avec :

– tf_{ji} : la fréquence de t_i dans e_j ;

– N : le nombre d'éléments dans la collection ;

– df_i : le nombre d'éléments qui contiennent le terme t_i ;

– ndl : le ratio entre la taille de e_j et la taille moyenne des éléments ;

– k_1 et b : les paramètres classiques de `BM25`.

Notons que la modification des paramètres k_1 et b permet de faire de `BM25` une fonction non linéaire en la fréquence des termes (voir l'analyse approfondie de Robertson et al. [ROB 04] pour plus de détails).

4.2. *Score de pertinence d'un élément XML basé sur les balises*

De la même manière que dans la section précédente, nous définissons M_j comme un vecteur de variables aléatoires T_{ik} à valeur dans $\{0, 1\}$. Les variables aléatoires M_j et leurs réalisations m_j représentent les éléments structurés :

$M_j = (T_{10}, ..., T_{1k}, ..., T_{1m}, ..., T_{n0}, ..., T_{nk}, .., T_{nm})$

avec :
$T_{ik} = 1$ si le terme t_i apparaît dans cet élément étiqueté par b_k
$T_{ik} = 0$ si le terme t_i n'est pas étiqueté par b_k
$T_{i0} = 1$ si le terme t_i apparaît sans étiquette dans B
$T_{i0} = 0$ si terme t_i n'apparaît pas sans être étiqueté

Nous notons $m_j = (t_{10}, ..., t_{1k}, ..., t_{1m}, ..., t_{n0}, ..., t_{nk}, .., t_{nm})$ une réalisation de la variable aléatoire M_j. Dans notre exemple, nous avons b_1 = *article*, b_2 = *section*, b_3 = *p*, b_4 = *b* et T = $\{t_1,..,t_5\}$. L'élément : e_j = t_1 t_2 t_3 de D_0 peut être représenté par le vecteur :

$$m_j = \{t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{20}, t_{21}, ..., t_{53}, t_{54}\} = \{0, 1, 0, 1, 0, 0, 1, ..., 0, 0\}$$

car le terme t_1 est étiqueté par *article* ($t_{11} = 1$), et *p* ($t_{13} = 1$) mais ni par *section* ($t_{12} = 0$) ni par *b* ($t_{14} = 0$). De plus, $t_{10} = 0$ car le terme n'apparaît pas sans étiquette.

Afin d'intégrer la structure des documents, nous ne considérons pas uniquement les poids des termes w_{ji}, mais aussi le poids des balises. Nous voulons estimer la pertinence d'un élément XML e_j (représenté par le vecteur m_j). On veut donc estimer :

$P(R|m_j)$: la probabilité de trouver une information pertinente (R) étant donné l'élément m_j.
$P(NR|m_j)$: la probabilité de trouver une information non pertinente (NR) étant donné l'élément m_j.

Soit $f_1(m_j) = \frac{P(R|m_j)}{P(NR|m_j)}$ une fonction de classement. Plus grande est la valeur de $f_1(m_j)$, plus pertinent est l'élément m_j. Utilisant la formule de Bayes, nous avons :

$$f_1(m_j) = \frac{P(m_j|R) \times P(R)}{P(m_j|NR) \times P(NR)}$$

Le terme $\frac{P(R)}{P(NR)}$ étant constant au regard de la collection pour une requête, il ne modifie pas la fonction de classement. Nous pouvons donc définir la fonction f_2 (proportionnelle à f_1) : $f_2(m_j) = \frac{P(m_j|R)}{P(m_j|NR)}$.

Admettant l'hypothèse d'indépendance nous avons :

$$\begin{aligned} P(M_j = m_j|R) &= \prod_{t_{ik} \in m_j} P(T_{ik} = t_{ik}|R) \\ &= \prod_{t_{ik} \in m_j} P(T_{ik} = 1|R)^{t_{ik}} P(T_{ik} = 0|R)^{1-t_{ik}} \end{aligned} \tag{3}$$

$$P(M_j = m_j|NR) = \prod_{t_{ik} \in m_j} P(T_{ik} = 1|NR)^{t_{ik}} P(T_{ik} = 0|NR)^{1-t_{ik}} \tag{4}$$

Pour simplifier les notations, on note, pour un élément XML donné :

$p_{i0} = P(T_{i0} = 0\|R))$:	la probabilité que t_i n'apparaisse pas sans étiquette étant donné un élément pertinent ;
$p_{ik} = P(T_{ik} = 1\|R)$:	la probabilité que t_i apparaisse étiqueté par la balise k, étant donné un élément pertinent ;
$q_{i0} = P(T_{i0} = 0\|NR)$:	la probabilité que t_i n'apparaisse pas sans étiquette étant donné un élément non pertinent ;
$q_{ik} = P(T_{ik} = 1\|NR)$:	la probabilité que t_i apparaisse étiqueté par la balise k, étant donné un élément non pertinent.

Avec ces notations les équations 3 et 4 deviennent :

$$\begin{aligned} P(m_j|R) &= \prod_{t_{ik} \in m_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}, \\ P(m_j|NR) &= \prod_{t_{ik} \in m_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}. \end{aligned}$$

La fonction de classement $f_2(m_j)$ peut alors s'écrire :

$$f_2(m_j) = \frac{\prod_{t_{ik} \in m_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}}{\prod_{t_{ik} \in m_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}}$$

La fonction *log* étant monotone croissante, prendre le logarithme ne changera pas les classements. On a donc la fonction f_3 :

$$\begin{aligned} f_3(m_j) &= \log(f_2(m_j)) \\ &= \sum_{t_{ik} \in m_j} (t_{ik} \log(p_{ik}) + (1 - t_{ik}) \log(1 - p_{ik}) \\ &\quad -t_{ik} \log(q_{ik}) - (1 - t_{ik}) \log(1 - q_{ik}) \\ &= \sum_{t_{ik} \in m_j} t_{ik} \times \left(\log \left(\frac{p_{ik}}{1 - p_{ik}} \right) - \log(\frac{q_{ik}}{1 - q_{ik}}) \right) + \sum_{t_{ik} \in m_j} \log(\frac{1 - p_{ik}}{1 - q_{ik}}) \end{aligned}$$

Comme précédemment, le terme $\sum_{t_{ik} \in m_j} \log(\frac{1-p_{ik}}{1-q_{ik}})$ est constant relativement à la collection (indépendant de t_{ik}). En ne le considérant pas, on obtient :

$$f_{tag}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} t_{ik} \times \log \left(\frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})} \right) \tag{5}$$

La fonction de classement obtenue prend en compte les poids des termes (t_i) et des balises (b_k). Le poids d'un terme t_i étiqueté par la balise b_k sera noté w'_{ik} :

$$w'_{ik} = \log(\frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}) \tag{6}$$

La pertinence d'un élément XML m_j, relativement aux balises est définie par $f_{tag}(m_j)$:

$$f_{tag}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} t_{ik} \times w'_{ik} \tag{7}$$

Du point de vue pratique, nous devons estimer les probabilités p_{ik} et q_{ik}, $i \in \{1, .., n\}$, $k \in \{0, .., m\}$ pour pouvoir évaluer la pertinence des éléments. À ces fins, nous utilisons un ensemble d'apprentissage LS composé d'éléments pour lesquels la pertinence est connue. Étant donné l'ensemble R (resp. NR) qui contient les éléments pertinents (resp. non pertinents), une table de contingence peut être construite pour chaque terme t_i étiqueté par la balise b_k :

	R	NR	$LS = R \cup NR$
$t_{ik} \in m_j$	r_{ik}	$nr_{ik} = n_{ik} - r_{ik}$	n_{ik}
$t_{ik} \notin m_j$	$R - r_{ik}$	$N - n_{ik} - R + r_{ik}$	$N - n_{ik}$
Total	R	$\|NR\| = N - R$	N

– r_{ik} : le nombre de fois où le terme t_i étiqueté par b_k est pertinent dans LS ;
– $\sum_i r_{ik}$: le nombre de termes pertinents étiquetés par b_k dans LS ;
– n_{ik} : le nombre de fois où le terme t_i est étiqueté par b_k dans LS ;
– nr_{ik} : le nombre de fois où le terme t_i étiqueté par b_k est non pertinent dans LS ;
– $R = \sum_{ik} r_{ik}$: le nombre de termes pertinents dans LS ;
– $|NR| = N - R$: le nombre de termes non pertinents dans LS.

Nous pouvons maintenant estimer $\left\{ \begin{array}{lll} p_{ik} = & P(t_{ik} = 1|R) & = \frac{r_{ik}}{R} \\ q_{ik} = & P(t_{ik} = 1|NR) & = \frac{n_{ik} - r_{ik}}{N - R} \end{array} \right.$

Il vient w'_{ik} :

$$\begin{aligned} w'_{ik} &= \log \frac{\frac{r_{ik}}{R}(1 - \frac{n_{ik} - r_{ik}}{N - R})}{\frac{n_{ik} - r_{ik}}{N - R}(1 - \frac{r_{ik}}{R})} \\ &= \log \frac{r_{ik} \times (N - n_{ik} - R + r_{ik})}{(n_{ik} - r_{ik}) * (R - r_{ik})} \\ &= \log \frac{r_{ik} \times (|NR| - nr_{ik})}{nr_{ik} \times (R - r_{ik})} \end{aligned} \tag{8}$$

Cette fonction de pondération évalue la probabilité, pour une balise donnée, de distinguer les termes pertinents des termes non pertinents : elle augmente avec la capacité de la balise à distinguer un terme pertinent. Notons que l'estimation des probabilités pourrait comporter un lissage dans le cas de collection d'apprentissage de taille limitée ; cela n'a pas été utile dans le cadre de nos expérimentations.

4.3. *Estimation du poids des balises*

D'un point de vue théorique (cf. équation 8), nous pouvons estimer un poids pour chaque paire (terme, balise), c'est-à-dire la capacité pour une balise donnée de renforcer un terme donné. Ce niveau de granularité est à notre avis trop fin. En effet, on cherche à modéliser l'impact d'une balise, non pas relativement à un terme particulier, mais de manière globale. Nous pensons que la capacité d'une balise à mettre en évidence les termes pertinents (ou au contraire à diminuer leur visibilité) est une propriété intrinsèque de la balise et ne dépend donc pas des termes. L'objectif est d'évaluer si un mot apparaissant dans un titre a plus d'importance qu'un mot apparaissant dans une section, et ce indépendamment du mot en question.

Nous nous intéressons donc non plus à un poids par paire (terme-balise), mais à un poids d'une balise indépendamment des termes qu'elle étiquette. Nous obtenons finalement pour chaque balise b_k le poids $w'_k = \frac{\sum_{t_i \in T} w'_{ik}}{|T|}$.

4.4. *Score de pertinence global d'un élément XML*

À partir des poids des termes et des balises, nous devons calculer un score global des éléments. Afin de prendre en compte toutes les balises qui englobent un terme, nous proposons de combiner la moyenne des poids de ces balises avec le poids du terme lui-même.

Ainsi, notre première fonction de combinaison, f_{claw} (Combining Linearly Average tag-Weights), s'écrit comme suit :

$$f_{claw}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} w_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{9}$$

avec w_{ji} le poids du terme t_i dans le document m_j, calculé à l'aide d'une fonction de pondération classique (`ltn`, `ltc`, ou `BM25`).

Dans [GéR 08], l'intégration du poids des balises permet d'améliorer le rappel, mais de manière peu significative. Or, la fonction de pondération `BM25` est non linéaire (cf. section 4.1). En conséquence, impacter le poids d'une balise sur le poids global w_{ji} est très différent de l'impacter sur le nombre d'occurrences du terme tf_{ji}. En accord avec Robertson et *al.* [ROB 04], nous proposons une prise en compte précoce du poids des balises, en intervenant directement sur tf_{ji}. Ainsi, la non-linéarité de la fonction `BM25` est exploitée. Le poids modifié (tf_{ji} multiplié par la moyenne du poids des balises qui englobent t_i), noté ttf (Tagged Term Frequency), remplace le tf dans la fonction de pondération `ltn`, `ltc`, ou `BM25`.

$$ttf_{ji} = tf_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{10}$$

5. Expérimentations

L'objectif de nos expérimentations est de comparer notre modèle avec un système de référence basé sur un modèle de RI classique, et avec les meilleurs systèmes participant à la compétition internationale de RI XML, INEX 2008. Nous avons expérimenté ces modèles sur une tâche de RI classique où la granularité des réponses est l'article complet, ainsi que sur une tâche de RI ciblée où la granularité des réponses est l'élément XML.

5.1. *Collection INEX - Wikipédia*

Nous avons utilisé le corpus XML anglophone INEX - Wikipédia [DEN 06], développé dans le cadre d'INEX. Ce corpus est composé de 659'388 articles extraits de l'encyclopédie en ligne Wikipédia[5], et d'un ensemble de requêtes et de jugements de pertinence associés. La syntaxe Wiki originelle a été convertie en XML, en utilisant des balises représentant la structure logique des articles (article, section, paragraph, title, list, item, etc.), des balises de mise en forme (bold, emphatic, italic, small, etc.) et des balises représentant des liens (collectionlink, etc.). Les articles sont fortement structurés : il y a au total 52 millions d'éléments XML. Chaque article peut être représenté comme un arbre contenant en moyenne 79 éléments, et ayant une hauteur moyenne de 6,72. Les articles complets (contenu textuel + structure XML) représentent 4,5 Go alors que le contenu textuel seul représente 1,6 Go. L'information structurelle XML (balises + attributs) représente donc le double de l'information textuelle.

5.2. *Protocole experimental*

Dans la phase d'apprentissage, les articles, les 114 requêtes et les jugements de pertinence de la collection 2006 ont été utilisés pour estimer le poids des balises w'_k. Ensuite, l'expérimentation a consisté à traiter sur la même collection de 659'388 documents les nouvelles requêtes de l'édition 2008 d'INEX.

L'évaluation est basée sur les critères de *précision* et de *rappel*. $iP[x]$ est la précision au point de rappel x. La mesure AiP combine *rappel* et *précision* en une seule mesure en calculant la moyenne de $iP[x]$ à 101 points de rappel (x = 0,00 ; 0,01 ; 0,02 ; ... 0,99 ; 1,00). Elle fournit une évaluation du système pour chaque requête. Enfin, le calcul de la moyenne des AiP sur l'ensemble des requêtes donne la mesure globale de performance $MAiP$ (*interpolated mean average precision* [KAM 07]). Le classement principal d'INEX est basé sur $iP[0.01]$ et non $MAiP$, afin de prendre en compte l'importance de la précision aux taux de rappel faibles.

---

5. Wikipédia : http ://wikipedia.org

Étant donné que chaque expérimentation est soumise à INEX sous la forme d'une liste ordonnée d'au plus 1'500 éléments XML pour chaque requête, ces mesures favorisent, en terme de rappel, les expérimentations retournant des articles complets (et donc une plus grande quantité d'information). Pour en tenir compte, nous avons aussi calculé $R[1500]$, le taux de rappel à 1'500 documents, et $S[1500]$, la taille des 1'500 éléments retournés (en Mo).

5.3. *Sélection des balises*

Pour décomposer les articles XML en éléments à indexer, 14 balises ont été sélectionnées comme représentant la structure logique des documents XML. Il s'agit des balises : *title*, *table*, *caption*, *article*, *body*, *section*, *numberlist*, *definitionitem*, *normallist*, *th*, *td*, *tr*, *p*, *row*. En conséquence, tous les éléments retournés par notre système correspondront à l'une de ces 14 balises.

Ensuite, les 61 balises ayant un nombre d'occurrence supérieur à 300 ont été sélectionnées parmi 1'257 balises apparaissant dans les 659'388 documents (cf. table 1). Enfin, 6 balises ont été supprimées manuellement : *article*, *body* (qui contiennent la totalité d'un article), *br*, *hr*, *s* et *value* (qui sont des balises sans contenu).

Tableau 1. *Nombre d'occurrences des balises (top 20)*

Balise	#occs	Balise	#occs
collectionlink	16'645'121	normallist	1'087'545
item	5'490'943	row	954'609
unknownlink	3'847'064	outsidelink	84'1443
cell	3'814'626	languagelink	739'391
p	2'689'838	name	659'405
emph2	2'573'195	body	659'396
template	2'396'318	article	659'389
section	1'575'519	conversionwarning	659'388
title	1'558'235	br	378'990
emph3	1'484'568	td	359'908

5.4. *Pondération des balises*

Les scores des 55 balises restantes ont été calculés suivant l'équation 8. Le tableau 2 présente les balises ayant obtenu les poids les plus élevés et les poids les plus faibles. Certaines balises ayant un score élevé sont inattendues (ex. : *sub*). Malgré le score très élevé de la balise *h4*, son impact sera minime sur les estimations de pertinence des éléments XML, car elle n'apparaît que 307 fois dans la collection.

Tableau 2. *Balises ayant les poids w'_k les plus faibles et les plus forts*

Poids les plus élevés (top 6)			Poids les plus faibles (top 6)		
Balise	Poids	#occs	Balise	Poids	#occs
h4	12,32	307	emph4	0,06	940
ul	2,70	3'050	font	0,07	27'117
sub	2,38	54'922	big	0,08	3'213
indentation1	2,04	135'420	em	0,11	608
section	2,01	1'610'183	b	0,13	11'297
blockquote	1,98	4'830	tt	0,14	6'841

6. Résultats

Nous présentons maintenant les résultats obtenus par notre modèle lors de la compétition INEX 2008. Suivant la procédure d'INEX, nous avons soumis 3 expérimentations à la tâche "focused" de la piste Ad-hoc. Cette tâche impose aux systèmes de retourner à l'utilisateur une liste d'éléments XML (ou de passages de texte) non recouvrants.

Notre objectif était tout d'abord d'obtenir une expérimentation de référence performante, puis d'évaluer notre modèle en RI classique et en RI ciblée, et enfin d'analyser l'impact de la prise en compte du poids des balises dans la fonction `BM25`. La table 3 présente les 3 expérimentations. La structure n'est prise en compte ni dans *Foc-1*, où les articles complets sont retournés (granularité : articles), ni dans *Foc-2*, où ce sont les éléments qui sont renvoyés (granularité : éléments), alors que dans *Foc-3*, le poids des balises est intégré dans `BM25` dans une recherche d'information ciblée (granularité : éléments, TTF).

Tableau 3. *Expérimentations soumises à la tâche "focused"*

			Pondération	
Expérimentations	Tâche	Granularité	des termes	des balises
Foc-1	Focused	articles	BM25	-
Foc-2	Focused	éléments	BM25	-
Foc-3	Focused	éléments	BM25	TTF

6.1. *Paramétrage du système*

Les paramètres de la fonction de pondération `BM25` ont été optimisés afin d'améliorer la RI classique (granularité : articles) et la RI ciblée (granularité : élément XML). Parmi les paramètres étudiés, nous pouvons mentionner l'utilisation d'un antidictionnaire, l'optimisation des principaux paramètres de `BM25` (k_1 = 1,1 et b = 0,75), etc. Considérant les requêtes, nous avons implémenté un mode "andish" (privilégiant

les documents contenant la totalité des mots-clés de la requête) et nous avons considéré les mots-clés *or* et *and* dans les requêtes. Certains paramètres spécifiques ont aussi été optimisés pour la RI ciblée (par exemple la taille minimum des éléments retournés). Nos expérimentations sont entièrement automatiques. Seuls les mots-clés de la requête ont été utilisés (champ *title* des requêtes INEX). Nous n'avons pas utilisé les champs *description*, *narrative* ou *castitle* (partie structurée de la requête).

6.2. *Classement INEX :* $iP[0.01]$

Notre système donne des résultats très intéressants comparés aux meilleurs participants à INEX (cf. tableau 4, et critères définis en section 5.2). Nos expérimentations sont comparées sur la figure 1 à *FOERStep* (Université de Waterloo), l'équipe qui a remporté la tâche "focused". *FOERStep* donne de meilleurs résultats à des taux de rappel faible. Par contre, notre expérimentation *Foc-1* donne les meilleurs résultats à des taux de rappel supérieurs à 0,05, et également en considérant le critère $MAiP$.

Tableau 4. *Évaluation de 61 expérimentations de la tâche "focused"*

Expérimentation	$iP[0.01]$	Rang	$MAiP$	Rang	R[1500]	S[1500]
FOERStep	**0.6873**	1	0.2071	27	0.4494	78
Foc-1	0.6412	13	**0.2791**	6	**0.7897**	390
Foc-2	0.5688	37	0.1206	45	0.2775	**51**
Foc-3	**0.6640**	7	0.2342	19	0.6110	234

6.3. *Articles versus éléments*

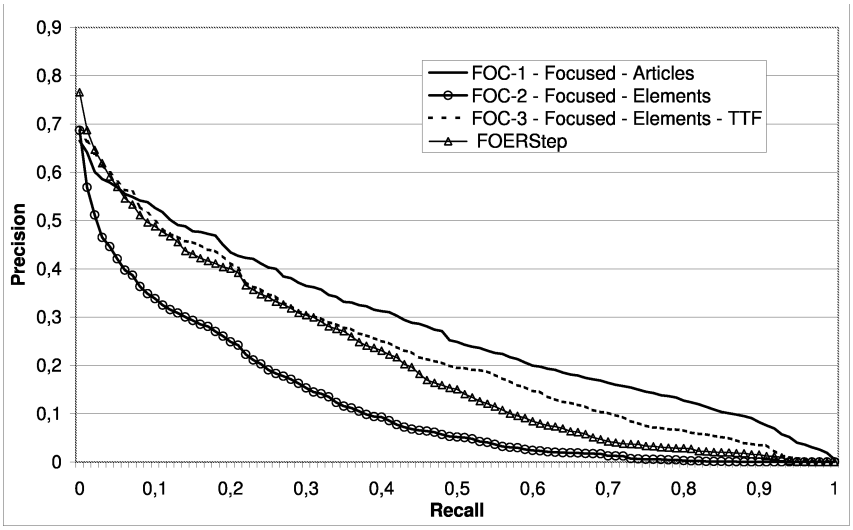
Afin de comparer la RI classique et la RI ciblée, nous avons indexé les articles complets d'une part (*Foc-1*) et les éléments XML de l'autre (*Foc-2*), et nous avons optimisé les paramètres du système dans les deux cas.

La RI ciblée (*Foc-2*), portant sur des éléments XML de taille et de granularité très variables, donne de moins bons résultats que la RI classique (*Foc-1*), bien que le paramètre `ndl` de `BM25` soit justement conçu pour prendre en compte des tailles de documents différentes, et donc des granularités de documents différentes. Les méthodes classiques de RI semblent peu adaptées à la RI ciblée lorsqu'elles sont appliquées telles quelles. D'ailleurs, 3 des 10 meilleures expérimentations sont basées sur des articles complets uniquement. La RI ciblée ne parvient donc pas encore à améliorer significativement les résultats de la RI classique.

6.4. *Impact des poids des balises sur les poids des termes*

Les paramètres utilisés pour *Foc-3* sont inchangés par rapport à ceux de *Foc-2*. La figure 1 montre que notre stratégie TTF (*Foc-3*) améliore significativement la RI

Figure 1. *Rappel / Précision de nos 3 expérimentations et des vainqueurs de la tâche "focused"*



ciblée à des taux de rappel faibles (de 0.5688 à 0.6640 selon le critère $iP[0.01]$). TTF donne également de meilleurs résultats que la RI classique (*Foc-1*). Enfin, le tableau 4 montre que *Foc-1* et *Foc-3* donnent de très bons résultats en terme de rappel : $MAiP$ de 0,2791 (resp. 0,2341) et $R[1500]$ de 0,7897 (resp. 0,6110). Le rappel à 1'500 documents décroît de 16% entre *Foc-1* et *Foc-3* alors que la taille en Mo des 1'500 documents décroît dans le même temps de 40%. Cela montre que le "tamis" de la RI ciblée élimine plus d'éléments non pertinents que d'éléments pertinents.

Ces résultats confirment aussi qu'il est important de conserver la non linéarité de la fonction de pondération `BM25`, par un impact précoce de l'information structurelle sur la fréquence des termes (stratégie TTF) plutôt que par un impact tardif de cette information directement sur les poids finaux des termes (stratégie CLAW, cf. [GéR 08]).

7. Conclusion et perspectives

Nous avons présenté dans cet article une nouvelle approche de prise en compte de la structure XML pour la RI ciblée, basée sur les principes du modèle probabiliste de RI. Nous considérons à la fois la structure logique et la structure de mise en forme. La structure logique est utilisée lors de la phase d'indexation, afin de définir les types d'éléments XML indexés (et potentiellement retournés) par le système. La structure logique et la structure de mise en page sont intégrées dans le modèle de document : lors d'une phase d'apprentissage, un poids est calculé pour chaque balise, basé sur la probabilité que la balise distingue les termes pertinents des termes non pertinents.

Lors de la phase d'interrogation, le calcul de la pertinence d'un élément XML pour une requête est une combinaison des poids des termes contenus et des poids des balises qui les étiquettent.

La contribution principale de notre modèle consiste en une modélisation de la capacité des balises à mettre en évidences les termes, suivant les principes du modèle probabiliste de RI. De cette manière, le réglage du poids des balises s'effectue de manière entièrement automatique. L'intégration tardive du poids des balises dans la fonction de pondération des termes ayant montré une amélioration peu significative des résultats [GéR 08], nous avons proposé dans cet article une intégration précoce, qui permet de conserver la non-linéarité de la fonction `BM25` et donne de bien meilleurs résultats.

Nous avons évalué notre modèle lors de la compétition internationale de RI XML, INEX 2008. Notre première expérimentation *Foc-1*, en RI classique (granularité des réponses : articles complets), se classe 13$^{\text{ème}}$ sur 61. Notre seconde expérimentation *Foc-2*, en RI ciblée (granularité des réponses : éléments XML), obtient un moins bon classement : 37$^{\text{ème}}$ sur 61. L'intégration précoce du poids des balises *Foc-3*, en RI ciblée, donne de très bons résultats en obtenant une 7$^{\text{ème}}$ place sur 61, montrant ainsi l'intérêt de la RI ciblée (*Foc-3*) comparée à la RI classique (*Foc-1*), montrant également l'intérêt de la prise en compte de l'information structurelle (*Foc-2* vs *Foc-3*) et montrant enfin de bien meilleurs résultats que l'intégration a posteriori du poids des balises [GéR 08].

Nous arrivons aux mêmes conclusions que Robertson et *al.* [ROB 04], bien que les collections utilisées soient très différentes (nombre et diversité des balises considérées) : il est intéressant de prendre en compte les balises dans la fonction de pondération `BM25`, dans la mesure où elles sont prises en compte de manière précoce. Par ailleurs, au contraire de [TRO 05], la prise en compte du poids des balises permet une amélioration significative de la fonction de pondération `BM25`.

Des perspectives s'offrent à nous à plusieurs niveaux. Tout d'abord, la stratégie TTF met en oeuvre une simple moyenne du poids des balises qui étiquettent un terme. De précédentes expérimentations ont montré que cette méthode donnait de meilleurs résultats que d'autres fonction de combinaison (multiplication des poids, prise en compte de la plus proche balise uniquement, etc.). Une analyse tant théorique qu'expérimentale est nécessaire sur ce point. La moyenne arithmétique utilisée met au même plan toutes les balises englobant un terme donné. Une pondération non uniforme des poids des balises, en fonction par exemple de la distance entre le terme et la balise, pourrait se révéler plus performante. Par ailleurs, des résultats positifs en RI ciblée ouvre des perspectives intéressantes en terme de présentation des résultats à l'utilisateur.

8. Bibliographie

[BOY 96] BOYAN J., FREITAG D., JOACHIMS T., « A Machine Learning Architecture for Optimizing Web Search Engines », *AAAI Workshop on Internet-Based Info. Systems*, 1996.

[DEN 06] DENOYER L., GALLINARI P., « The Wikipedia XML corpus », *SIGIR forum*, vol. 40, 2006, p. 64-69.

[FUH 01] FUHR N., GROSSJOHANN K., « XIRQL : A Query Language for Information Retrieval in XML Documents », *SIGIR*, 2001, p. 172-180.

[FUL 93] FULLER M., MACKIE E., SACKS-DAVIS R., WILKINSON R., « Coherent Answers for a Large Structured Document Collection », *SIGIR*, 1993, p. 204-213.

[GéR 08] GÉRY M., LARGERON C., THOLLARD F., « Integrating structure in the probabilistic model for Information Retrieval », *Web Intelligence*, 2008, p. 763-769.

[KAM 07] KAMPS J., PEHCEVSKI J., KAZAI G., LALMAS M., ROBERTSON S., « INEX 2007 Evaluation Measures », *Focused access to XML documents, INEX Workshop*, 2007.

[KIM 00] KIM Y.-H., KIM S., EOM J.-H., ZHANG B.-T., « SCAI Experiments on TREC-9 », *Text Retrieval Conference (TREC-9)*, 2000, p. 392-399.

[KOT 02] KOTSAKIS E., « Structured Information Retrieval in XML documents », *Symposium on Applied Computing*, 2002, p. 663-667.

[NAV 95] NAVARRO G., BAEZA-YATES R. A., « A Language for Queries on Structure and Contents of Textual », *SIGIR*, 1995, p. 93-101.

[RAP 01] RAPELA J., « Automatically combining ranking heuristics for HTML documents », *Workshop on Web Information and Data Management (WIDM), CIKM*, 2001, p. 61-67.

[ROB 76] ROBERTSON S., JONES K. S., « Relevance weighting of search terms », *JASIST*, vol. 27, n° 3, 1976, p. 129–146.

[ROB 04] ROBERTSON S., ZARAGOZA H., TAYLOR M., « Simple BM25 extension to multiple weighted fields », *CIKM*, New York USA, 2004, p. 42–49.

[SAL 83] SALTON G., MCGILL M., *Introduction to modern Information Retrieval*, McGraw-Hill, 1983.

[SCH 02] SCHLIEDER T., MEUSS H., « Querying and ranking XML documents », *JASIST*, vol. 53, n° 6, 2002, p. 489-503.

[TRO 05] TROTMAN A., « Choosing document structure weights », *Information Processing and Management*, vol. 41, n° 2, 2005, p. 243-264.

[WIL 94] WILKINSON R., « Effective Retrieval of Structured Documents », *SIGIR*, July 1994, p. 311–317.

[WOL 00] WOLFF J. E., FLORKE H., CREMERS A. B., « Searching and Browsing Collections of Structural Information », *Advances in Digital Libraries*, 2000, p. 141-150.

Chapitre 7

Articles Courts

GraphDuplex : visualisation simultanée de N réseaux couplés 2 par 2

Martine Hurault-Plantet* — Elie Naulleau — Bernard Jacquemin*****

** LIMSI-CNRS*
Bâtiment 508, Université Paris XI
91403 Orsay
Martine.Hurault-Plantet@limsi.fr
*** Semiosys*
7 rue des roses
85100 Les Sables d'Olonne
semiosys@semiophore.net
**** SCIMEC - CREM EA 3476*
Université de Haute Alsace, Faculté des Lettres et Sciences Humaines,
10, rue des Frères Lumière
68093 Mulhouse
Bernard.Jacquemin@uha.fr

RÉSUMÉ. *L'analyse des réseaux sociaux fait un usage intensif d'outils de visualisation et, dans le domaine de la recherche d'information, l'exploration visuelle de réseaux lexicaux est utilisée comme une aide à la désambiguïsation ou au raffinement de la requête. Ces deux types de réseaux se trouvent associés via Internet lorsqu'un contenu textuel est lié à une activité sociale (méls, blogs, travail collaboratif). Dans cet article, nous présentons un logiciel de visualisation simultanée de plusieurs réseaux, GraphDuplex, qui, combiné à des méthodes statistiques, permet par exemple d'étudier conjointement un réseau social (ou plusieurs) et son réseau lexical associé. GraphDuplex permet en particulier des requêtes dynamiques inter-réseaux, entre les nœuds ou les liens des deux réseaux.*

ABSTRACT. *While social network analysis often focuses on graph structure of social actors, an increasing number of communication networks now provide textual content within social activity (email, instant messaging, blogging, collaboration networks). We present an open source visualization software, GraphDuplex, which brings together social structure and textual content, adding a semantic dimension to social analysis. GraphDuplex eventually connects any number of social or semantic graphs together, and through dynamic queries enables user interaction and exploration across multiple graphs of different nature.*

MOTS-CLÉS : *visualisation interactive, réseau social, réseau lexical, requête dynamique.*

KEYWORDS: *interactive visualization, social network, lexical network, dynamic query.*

1. Introduction

L'exploration des propriétés d'un réseau se fait depuis longtemps à l'aide d'outils de visualisation, en supplément ou en complément des outils mathématiques classiques de la théorie des graphes. L'analyse des réseaux sociaux en particulier a suscité de nombreuses recherches sur le sujet (Freeman, 2004). Les logiciels qui en découlent proposent souvent, en plus de la visualisation proprement dite du réseau, des méthodes permettant de mieux cibler ce qui est visualisé et d'avoir du réseau des vues à la fois globales et locales. En particulier, (Brandes *et al.*, 2003) présentent un outil d'exploration d'un réseau social qui propose différents algorithmes de dessin de graphe, privilégiant la facilité de lecture, adaptés aux réseaux de petite et moyenne taille. Par ailleurs, (Perer *et al.*, 2006) ont développé un outil de visualisation interactive qui intègre un ensemble de méthodes statistiques permettant de mettre en valeur visuellement, par des couleurs ou des tailles de nœuds ou liens, des propriétés particulières du réseau. L'intérêt s'est porté aussi sur la visualisation de très grands réseaux (Batagelj *et al.*, 2003).

Moins présent sur ce thème, le domaine du traitement des données textuelles a cependant intégré depuis longtemps des outils de visualisation pour la représentation et l'analyse des réseaux lexicaux. (Véronis, 2003) s'appuie sur la construction des différentes composantes de forte densité d'un réseau lexical pour distinguer les différents usages d'un même mot, dans le but d'utiliser son environnement lexical en recherche d'information. La visualisation associée permet à l'utilisateur de naviguer dans les thèmes liés au mot sélectionné. (Tunkelang *et al.*, 2006). propose une méthode de raffinement d'une requête en recherche d'information par une navigation dans le réseau lexical des documents guidée par les termes de la requête.

Cependant, on n'a accordé jusqu'à présent que peu d'attention à la visualisation simultanée de plusieurs réseaux. Il s'agit de réseaux distincts dont les nœuds ont, en plus de la relation qui les lie entre eux à l'intérieur de chaque réseau, une autre relation qui les lie aux nœuds d'un autre réseau[1]. Les nœuds dans chaque réseau sont de même nature, en revanche ils sont en général de natures différentes d'un réseau à l'autre. Nous présentons dans ce qui suit le logiciel GraphDuplex[2] qui permet de visualiser simultanément plusieurs réseaux qu'on peut coupler deux par deux. Ce couplage permet des requêtes dynamiques. En effet, la sélection d'un nœud[3] de l'un des réseaux couplés entraînera une modification visuelle de l'autre réseau, suivant la relation qui les lie. Ce couplage est paramétrable et suivant le type des données propose différentes relations (égalité, relations d'ordre, opérateurs ensemblistes, ...). GraphDuplex visualise les réseaux dans des fenêtres séparées. Chaque fenêtre possède un tableau de bord qui permet de régler différents paramètres de

[1] Dans les graphes N-partis, on ne considère que les liens qui relient des nœuds appartenant à des ensembles différents. Ici, on considère aussi les liens entre nœuds d'un même ensemble.

[2] L'application et le code source Java sont téléchargeables sur http://www.semiophore.net avec une vidéodémonstration : http://semiosys.free.fr/video/graphduplex/ .

[3] On peut sélectionner plusieurs nœuds ou encore des liens.

visualisation. Les paramètres globaux sur le réseau sont l'algorithme de dessin du graphe[4], la possibilité de déplacer chaque nœud du graphe ou le graphe dans son ensemble, et enfin la possibilité de déplacer une loupe sur le graphe afin d'en visualiser les détails. L'utilisation de cette loupe est particulièrement intéressante lors de la visualisation de grands graphes. Les paramètres sur les nœuds (ou sur les liens) comprennent l'affichage ou non des libellés, un choix de représentation des propriétés du nœud (ou du lien) par des variations de couleur, de taille, ou encore par des secteurs distributionnels. Les ajustements des paramètres de visualisation permettent déjà de mettre en valeur certaines propriétés du réseau. Il s'y ajoute un ensemble de possibilités de filtrage interactif qui permettent de ne visualiser que des parties du réseau qui possèdent en commun une ou plusieurs propriétés données sur les liens ou sur les nœuds. Par ailleurs les deux réseaux sont couplés, c'est-à-dire qu'une action de clic sur l'un des éléments d'un réseau déclenche la mise en évidence visuelle des éléments de l'autre réseau qui lui sont liés. Par exemple, cliquer sur un nœud-individu du réseau social sélectionne visuellement le sous-réseau lexical du vocabulaire de cet individu. Les données des réseaux sont chargées dans GraphDuplex soit à partir de données sauvegardées dans GraphDuplex lors d'une précédente session, soit, initialement, à partir des données d'une base de données, à laquelle on accède par un fichier XML. Ce fichier contient les interrogations de la base de données permettant de sélectionner les données qu'on veut visualiser.

Ce logiciel a été développé dans le cadre du projet Autograph[5] sur la conception d'outils de visualisation pour la gouvernance des communautés collaboratives sur Internet, dont, en particulier, la communauté des contributeurs à Wikipédia. Les productions écrites des wikipédiens ne se limitent pas aux articles encyclopédiques, elles comprennent aussi toutes les discussions qui s'y réfèrent. D'une part cette communauté constitue un réseau social, qui se subdivise en sous-communautés suivant le type de lien social (par exemple, travail sur un même domaine, ou participation à des tâches semblables d'administration de l'encyclopédie), et on peut donc étudier ce réseau social en tant que tel. Et d'autre part cette communauté partage un lexique, utilise, ou n'utilise pas, des termes semblables, et le réseau lexical qui en découle peut également être étudié en tant que tel. Mais ces deux types de réseaux sont également liés, chaque acteur du réseau social utilisant une partie du lexique, et chaque mot du lexique étant utilisé par un ensemble d'acteurs. C'est l'ensemble de ces propriétés, thèmes et sous-thèmes des différentes communautés, qu'une interface de visualisation simultanée de plusieurs réseaux permet d'explorer. Dans cet article, nous montrons les différentes possibilités du logiciel sur l'exemple du réseau social des arbitres du Comité d'arbitrage de Wikipédia associé au réseau lexical du vocabulaire qu'ils utilisent au cours des arbitrages.

[4] Un ensemble de 10 algorithmes de dessin de graphe sont disponibles, apportés par les librairies Jung et Graphviz.

[5] http://autograph.fing.org/texts/PresentationAutograph

2. Le réseau des arbitres

2.1 L'arbitrage dans Wikipédia

Wikipédia est un projet encyclopédique libre sur Internet (Zlatic et al., 2006), couvrant tous les domaines du savoir, au sein de différentes communautés de langue gérant leur projet de manière autonome. Ce savoir doit être présenté de manière objective, suivant le principe de la *neutralité de point de vue* (Viégas et al., 2004), et l'ensemble du processus éditorial, de l'écriture des articles à l'organisation de la macrostructure, est géré collectivement. Cela a impliqué la mise en place progressive de divers instruments et procédures de régulation et de contrôle (Viégas et al., 2007). En particulier, un comité d'arbitrage a été mis en place pour régler les litiges d'édition sévères entre contributeurs.

Dans l'instance française de Wikipédia, le comité d'arbitrage est un groupe composé de sept membres de la communauté des contributeurs, élus par la communauté pour une période de six mois. Ils sont chargés de recevoir les plaintes des contributeurs en conflit ouvert (avec insultes dans les pages de discussion par exemple), lorsque toutes les possibilités de médiation sont épuisées. Les délibérations et les votes du comité d'arbitrage sont publics sur des pages de Wikipédia qui leur sont dédiées[6] et cherchent autant que possible l'unanimité, privilégiant donc le consensus comme c'est la règle dans les articles. Les sanctions votées par ce comité peuvent aller du blocage (interdiction technique et temporaire de contribuer sur un ou plusieurs articles) au bannissement définitif (interdiction de participer à tout contenu de Wikipédia).

2.2 Réseau social et réseau lexical

L'encyclopédie Wikipédia peut être librement téléchargée[7] et exploitée. Nous disposons de la sauvegarde de la base Wikipédia française réalisée le 2 avril 2006, soit plus de 600 000 pages comprenant notamment près de 370 000 pages d'articles auxquelles sont associées plus de 40 000 pages de discussion sur article.

Le corpus des arbitrages est constitué des quatre-vingts pages d'arbitrages de notre base Wikipédia, suffisamment bien formées pour que l'information qu'elles contiennent puisse être appréhendée automatiquement. Cent dix protagonistes et dix-neuf arbitres ont confronté leurs avis au cours de ces débats. Chaque page d'arbitrage doit respecter une structure donnée, qui consiste d'abord en une description du conflit (les parties concernées, la nature du conflit), ensuite les preuves et arguments des protagonistes, puis les commentaires des arbitres, et enfin le vote et la décision.

---

[6] http://fr.wikipedia.org/wiki/Wikipédia:Comité d'arbitrage/Arbitrage

[7] Un fichier de sauvegarde de l'ensemble des données textuelles sous forme de données MySQL compressées est mis à disposition sur http://download.wikipedia.org/backup-index.html et régulièrement mis à jour.

Nous avons défini le lien social entre les arbitres par leur accord ou désaccord dans les votes de décision d'arbitrage. La Figure 1 montre le réseau des arbitres visualisé par GraphDuplex. Les nœuds du réseau représentent les arbitres et les liens entre les nœuds expriment leurs accords. Nous considérons qu'il y a un accord entre deux arbitres lorsqu'ils votent tous deux de la même manière, c'est-à-dire pour, ou bien contre, une proposition d'arbitrage.

Le poids sur chaque nœud correspond au nombre de participations de l'arbitre à un vote. Sa valeur varie entre 2 et 87. Ces valeurs peuvent être visualisées par des tailles différentes de nœud ou par des teintes différentes de couleur, comme sur la Figure 1 où la couleur du nœud est plus ou moins foncée suivant que l'arbitre a participé à plus ou moins de votes. Dans chaque composante connexe, on remarque un noyau d'arbitres ayant très souvent voté (nœuds plus foncés), et ayant été lors de ces votes très souvent en accord les uns avec les autres (liens plus foncés et plus épais). La plus grande des deux composantes correspond à un ensemble d'arbitres qui ont arbitré pendant une grande partie de la période considérée (2001-2006).

Le poids sur le lien entre deux arbitres correspond à leur proportion d'accord sur l'ensemble des votes auxquels ils ont participé ensemble. Sa valeur varie entre 25% et 100%. Ces valeurs peuvent être visualisées par des tailles différentes de lien ou des variations de couleur, ou les deux comme sur la Figure 1 où la taille des liens varie et leur couleur aussi varie en nuance, du foncé au clair, suivant la plus ou moins grande proportion d'accord. L'accord est indépendant du nombre de participations aux votes ; Solensean par exemple, qui a participé à moins de votes que Traroth (noeud plus clair), est en meilleur accord que lui avec les autres arbitres (liens plus foncés).

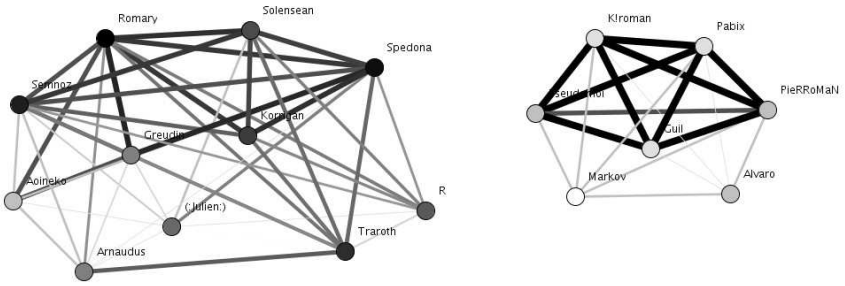


Figure 1. *Le réseau des arbitres du Comité d'arbitrage de Wikipédia entre début 2001 et avril 2006*

Le réseau lexical associé est constitué de l'ensemble des noms, adjectifs, verbes, et adverbes que les arbitres utilisent au cours de leurs débats dans les arbitrages. Dans le réseau lexical, nous n'avons conservé que les termes dont la fréquence-document dans ce corpus, c'est-à-dire le nombre d'arbitres qui utilisent ce terme, est

au moins égale à 10. Le réseau lexical résultant comporte 97 nœuds-termes[8] dont la fréquence varie entre 10 et 18. Le poids de chaque nœud est la fréquence-document du terme. Ce poids peut être visualisé par la taille et la couleur comme sur la Figure 2 où la couleur est plus ou moins foncée suivant la plus ou moins grande fréquence du nœud-terme.

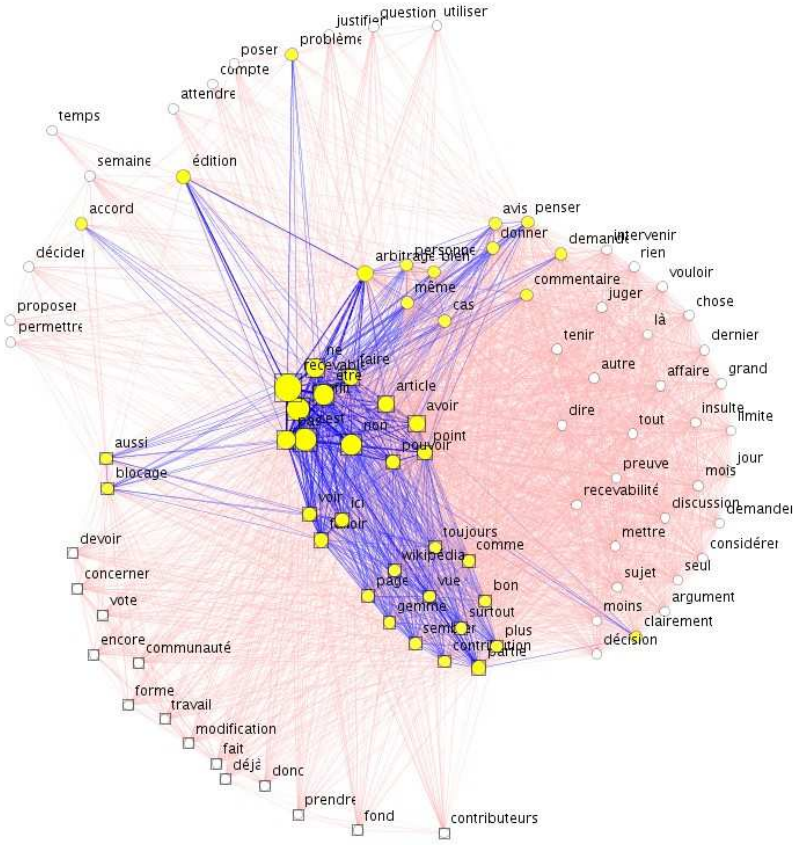


Figure 2. *Le réseau lexical du vocabulaire des arbitres dans les arbitrages Wikipédia entre début 2001 et avril 2006.*

Dans le réseau lexical, deux nœuds-termes sont reliés s'ils sont tous deux utilisés par le même arbitre. Le poids du lien est d'autant plus fort que les deux termes sont

[8] Le nombre de noeuds peut être de plusieurs milliers, et la fonction loupe permet alors de visualiser les détails du réseau.

utilisés par un plus grand nombre d'arbitres, en valeur absolue (cooccurrence) ou relativement à leur fréquence dans le corpus (mesure d'équivalence et information mutuelle). La force du lien peut être visualisée par l'épaisseur et la nuance de couleur, comme sur la Figure 2 où la couleur est plus ou moins foncée suivant le plus ou moins grand nombre de cooccurrences entre les deux termes.

3. La fenêtre de visualisation d'un réseau

3.1 *Filtrage sur les nœuds et les liens du réseau*

Le logiciel GraphDuplex permet un ensemble de filtrages interactifs sur les nœuds et sur les liens du réseau visualisé. Il est possible de filtrer les nœuds du réseau par le nom du nœud, en sélectionnant ceux qu'on veut afficher ou masquer, ou par le poids du nœud, en sélectionnant un seuil inférieur ou un seuil supérieur sur la valeur du poids, ou les deux à la fois. On peut aussi filtrer les nœuds et les liens par un autre attribut, dans notre cas les mots du lexique des arbitres.

3.1.1 *Filtrage sur le poids*

Le filtrage sur le poids des nœuds du réseau social permet de ne visualiser qu'une certaine catégorie d'arbitres, liée au nombre de participations à des arbitrages.

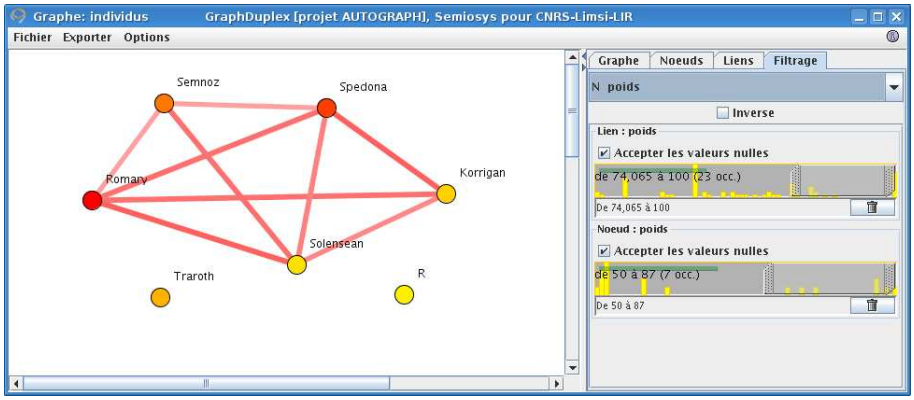


Figure 3. *Le réseau des arbitres de Wikipédia filtré par leur nombre de participations (au moins 50 votes au Comité d'arbitrage), et leur nombre d'accords (au moins 75% d'accord sur ces votes).*

Nous nous sommes intéressés plus particulièrement aux arbitres les plus actifs. Pour cela, nous filtrons le réseau sur le poids des nœuds, c'est-à-dire sur le nombre de votes à des décisions d'arbitrage auxquels les arbitres ont participé. Par ailleurs, le degré d'accord entre arbitres est encore plus visible lorsqu'on filtre les liens qui les représentent par un seuil sur leur poids, c'est-à-dire sur la proportion d'accord entre

arbitres. La Figure 3 montre le réseau obtenu en filtrant le réseau des arbitres sur une participation à au moins 50 votes, et en ne conservant que les liens d'au moins 75% d'accord. Nous voyons donc que les arbitres Traroth et R ont un accord médiocre avec les autres arbitres, alors que ces autres arbitres ont globalement un meilleur accord entre eux. L'arbitre R est d'ailleurs resté peu de temps au Comité d'arbitrage.

3.1.2 *Filtrage sur le lexique*

Le logiciel Graphduplex nous permet aussi d'étudier le vocabulaire des arbitres, et, en particulier, d'identifier quels termes les différencient. Un filtrage sur chacun des termes du vocabulaire permet de visualiser les liens entre les arbitres qui possèdent ce terme en commun dans leur vocabulaire[9]. Par exemple, nous voyons sur la Figure 4 que la sélection du mot *permettre* supprime les liens reliant l'arbitre R aux autres arbitres. Cela signifie que tous les arbitres, sauf R, utilisent le terme *permettre*. De la même manière, en sélectionnant le mot *justifier*, les liens reliant l'arbitre Solensean aux autres arbitres sont supprimés. L'arbitre Solensean est le seul à ne pas utiliser le terme *justifier*.

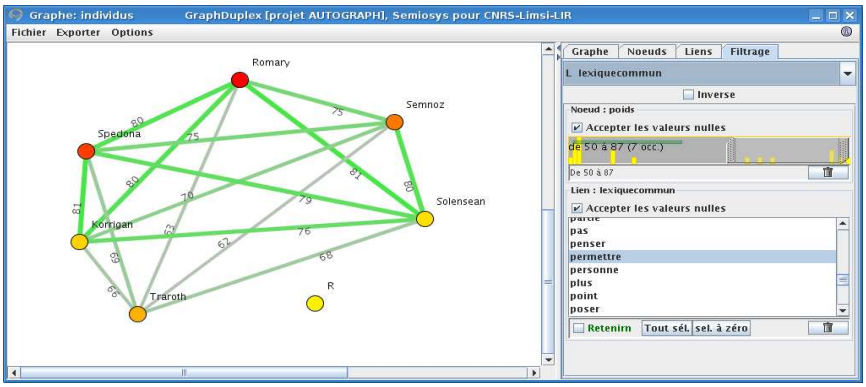


Figure 4. *Le réseau des arbitres de Wikipédia ayant participé à au moins 50 votes au Comité d'arbitrage, et ayant en commun le mot « permettre ».*

En utilisant ce filtrage sur tous les termes du lexique des arbitres, nous constatons que les arbitres qui ont participé à au moins 50 votes d'arbitrage, possèdent un très large vocabulaire en commun. Les différences que nous avons notées concernent l'arbitre R qui n'utilise pas les termes *attendre*, *permettre*, *contributeurs*, *fond*, *prendre*, et *déjà*, contrairement aux autres arbitres, et l'arbitre Solensean qui est le seul à ne pas utiliser le mot *justifier*.

[9] On peut faire le même filtrage de vocabulaire sur les nœuds-arbitres que sur les liens entre arbitres. Dans un cas on masque les arbitres qui n'utilisent pas un terme donné, dans l'autre on masque les liens entre arbitres qui n'ont pas ce terme en commun.

3.2 *Visualisation des propriétés d'un nœud*

On peut également visualiser, pour chaque arbitre, la distribution du vocabulaire qu'il utilise. La Figure 5 montre la distribution d'un même ensemble de 7 termes (sauf pour l'arbitre R qui n'utilise pas l'un des 7 mots, *attendre*) pour chaque arbitre. On constate que les arbitres R et Solensean ont un profil de lexique différent de celui des autres.

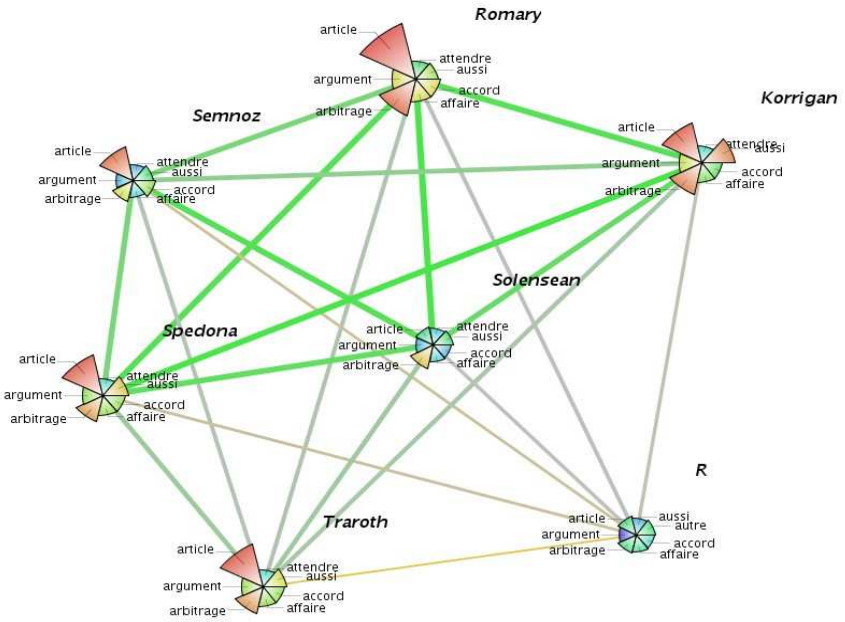


Figure 5. *Distribution des 7 mêmes termes pour chaque arbitre ayant participé à au moins 50 votes.*

4. Visualisation croisée entre deux réseaux

La sélection d'un nœud-terme du réseau lexical met visuellement en évidence dans le réseau social tous les nœuds-arbitre qui utilisent ce mot. Inversement, la sélection d'un nœud-arbitre du réseau social met en évidence dans le réseau lexical tous les nœuds-terme utilisés par cet arbitre. La sélection initiale d'un nœud dans l'un des réseaux est marquée par une couleur particulière sur le nœud, les mises en

évidence en réaction dans l'autre réseau sont marqués par un carré sur le nœud[10]. Par exemple, la Figure 6 montre les termes utilisés par l'arbitre Aoineko, sélectionné dans le réseau social.

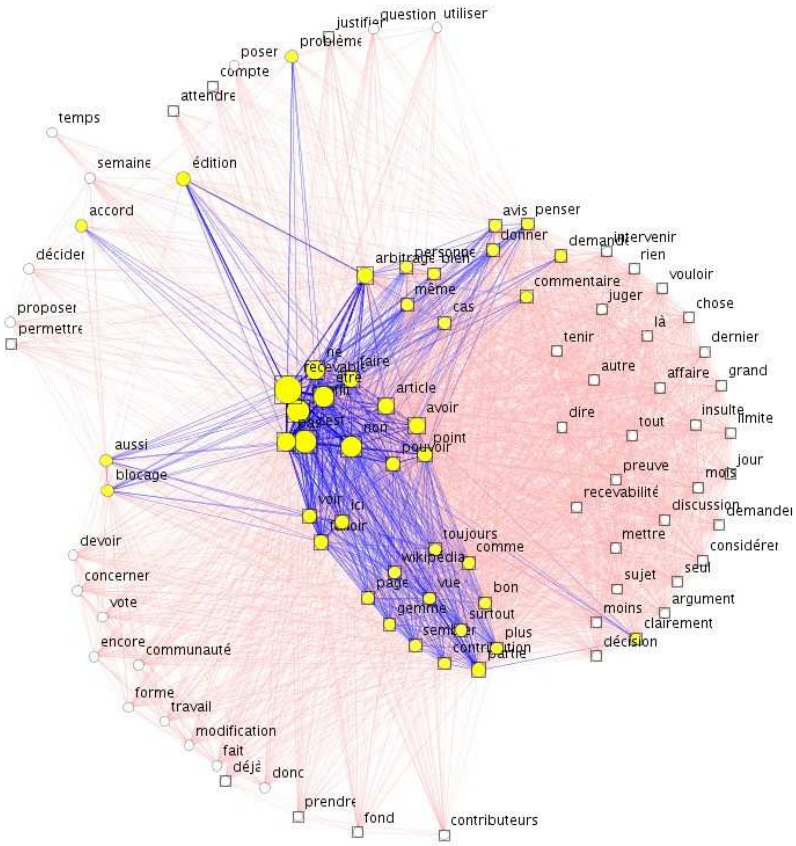


Figure 6. *Les nœuds-termes utilisés par l'arbitre Aoineko (entourés d'un carré), dans le lexique des arbitres.*

Cet arbitre utilise un large vocabulaire : on voit en effet que très peu de nœuds-termes ne sont pas entourés d'un carré. En revanche, la Figure 2, qui met en évidence les nœuds-termes utilisés par l'arbitre Greudin (entourés d'un carré), montre que

[10] Il est possible d'effectuer ses propres choix de couleurs par un paramétrage personnalisé dans l'interface.

ceux-ci sont moins nombreux. On peut ainsi comparer visuellement les tailles des vocabulaires utilisés par les différents arbitres.

5. Conclusion

Le logiciel GraphDuplex, par un ensemble de paramétrages de visualisation et de filtrage, associés à des méthodes statistiques et des méthodes de calcul sur des graphes, permet une exploration interactive de plusieurs réseaux. Ce logiciel, développé en Java, utilise deux librairies opensource Jung (O'Madadhain *et al.*, 2003) et InfoVis Toolkit (Fekete, 2004), ainsi que des composants Semiophore. Il exploite également Graphviz d'ATT[11]. La mise en correspondance des attributs typés dans une base de données et des attributs graphiques dans les graphes ont été spécifiquement développés pour GraphDuplex. Cela passe par une IHM qui donne la possibilité de régler les paramètres graphiques en fonction des attributs (champs de la base de données). La facilité à se brancher sur n'importe quelle base provient d'une couche abstraction qui décrit dans un modèle XML les liens entre les données et la morphologie des graphes (nœuds, liens) ainsi que les liens entre les différents graphes (contraintes de sélections transversales).

GraphDuplex est particulièrement intéressant si l'on veut analyser un réseau lexical lié à une activité sociale, mais il peut être utilisé à d'autres fins, comme par exemple la comparaison de deux réseaux lexicaux sur un même thème, à des temps différents. Pour montrer les possibilités du logiciel nous avons pris l'exemple du réseau social des arbitres au Comité d'arbitrage de Wikipédia, associé au réseau lexical de leurs interventions dans ce même comité. En ce qui concerne le réseau lexical, nous avons pu mettre en évidence les différences de vocabulaire entre les arbitres (Figures 4 et 5), les différences entre les distributions d'un même ensemble de mots pour différents individus du réseau social, ou bien encore les thèmes les plus fréquents. Les requêtes dynamiques inter-réseaux permettent aussi de repérer les individus du réseau social qui utilisent les termes et les thèmes mis en évidence dans le réseau lexical.

Pour compléter cette étude, un autre réseau social pourrait être ajouté, celui des contributeurs à Wikipédia qui comparaissent devant le Comité d'arbitrage. Trois réseaux liés entre eux seraient ainsi visualisés simultanément : le réseau social des arbitres, le réseau social des protagonistes des conflits, et le réseau lexical des interventions de l'ensemble des individus des deux réseaux sociaux. Des comparaisons pourraient ainsi être faites entre les vocabulaires des arbitres et des contributeurs protagonistes des conflits.

[11] http://www.graphviz.org

Remerciements

Ce travail a été réalisé dans le cadre du projet Autograph ANR-05-RNRT-03002 (S0604108 W)

6. Bibliographie

Batagelj V., Mrvar A., « Pajek – Analysis and Visualization of Large Networks », *Graph Drawing Software*, Michael Jünger and Petra Mutzel (Eds.), Springer Verlag, 2003, p. 77-103.

Brandes U., Wagner D., « visione – Analysis and Visualization of Social Networks », *Graph Drawing Software*, Michael Jünger and Petra Mutzel (Eds.), Springer Verlag, 2003, p. 321-340.

Fekete J-D., The InfoVis Toolkit. *IEEE Symposium on Information Visualization (InfoVis 04)*, Austin, TX, IEEE Press, October 2004, pages 167-174.

Freeman L. C., *The Development of Social Network Analysis: A Study in the Sociology of Science*, Vancouver, Empirical Press, 2004.

O'Madadhain J., Fisher D., White S., et Boey Y., The JUNG (Java Universal Network/Graph) Framework, UCI-ICS Tech Report 03-17, October 2003

Perer A., Shneiderman B., « Balancing Systematic and Flexible Exploration of Social Networks », *IEEE Transactions on Visualization and Computer Graphics (InfoVis 2006)*, vol. 12, n° 5, 2006, p. 693-700.

Tunkelang D., Byrd R., Cooper J., « Lexical navigation: Using incremental graph drawing for query refinement », *Graph Drawing*, Liotta G. (Eds.), Lecture Notes in Computer Science, vol. 2912, 2004, p. 316-321.

Véronis J., « Hyperlex : cartographie lexicale pour la recherche d'information», *Actes de la conférence Traitement Automatique des Langues Naturelles TALN'2003*, Batz-sur-mer, 2003, p. 265-274.

Viégas F. B., Wattenberg M., Dave K., « Studying Cooperation and Conflict between Authors with history flow Visualizations », *Actes de SIGCHI conference on Human factor in computing systems*, Vienne, Autriche, 2004, p. 575-582.

Vièǵas F. B., Wattenberg M., Kriss J., Van Ham F., « Talk Before You Type : Coordination in Wikipedia », *HICSS'07 : Actes de Hawaii International Conference on System Sciences*, 2007, p. 78

Zlatic V., Bozicevic M., Stefancic H., Domazet M., « Wikipedias : Collaborative web-based encyclopedias as complex networks », *Physical Review E*, vol. 74, n° 1, 2006, p. 6-11.

Prise en compte des liens pour améliorer la recherche d'information structurée

MATAOUI M'hamed [*,], Mohamed MEZGHICHE[**]**

** Laboratoire SI, EMP*
BP 17, Bordj el Bahri
16111, Alger, ALGERIE

*** LIFAB, Université de BOUMERDES*
35000, Boumerdes, ALGERIE
{mataoui_mhamed, mezghiche}@umbb.dz

RÉSUMÉ. *Dans cet article nous présentons deux adaptations de l'algorithme PageRank aux collections de documents XML et les résultats d'expérimentation obtenus pour la collection Wikipedia utilisée dans INEX 2007. Ces adaptations que nous appelons "DOCRANK" et "HITS_docrank" permettent un reclassement des résultats renvoyés par l'exécution de base (base run) pour en améliorer la qualité. Nos expérimentations sont effectuées sur les résultats renvoyés par les trois systèmes les mieux classés pour la tâche "Focused" d'INEX 2007. Les évaluations que nous avons menés ont montrés des améliorations de la qualité des résultats (voir très significatives pour certaines "topics", ex : 491, 521, etc.). La meilleure amélioration obtenue pour les résultats renvoyés par le système de l'université DALIAN (pour l'ensemble des 107 topics d'INEX 2007) était de l'ordre de 3.78%.*

ABSTRACT. *In this paper we present two adaptations of the PageRank algorithm to collections of XML documents and the experimental results obtained for the collection Wikipedia used in INEX 2007. Those adaptations that we call "DOCRANK and HITS_docrank" allow the re-rank of the results returned by the base run execution to improve retrieval quality. Our experiments are applied on the results returned by the best three systems ranked in the "Focused" task of INEX 2007. Evaluations have shown improvements in the quality of retrieval results (improvement of some topics is very significant, eg: 491, 521, etc.). The best improvement achieved in the results returned by the DALIAN university system (all 107 topics of INEX 2007) was about 3.78%.*

MOTS-CLÉS : *Recherche d'information structurée (RIS) , XML, liens XML, INEX.*

KEYWORDS: *Structured information retrieval (SIR), XML, XML links, INEX .*

1. Introduction

La recherche d’information sur le web diffère de la RI traditionnelle. La principale différence réside dans la structure du Web qui est basée sur les liens hypertextes qui représentent une nouvelle source d’évidence pour mesurer la pertinence des pages. Cette structure a été exploitée par plusieurs moteurs de recherche (exemple : Google).

Plusieurs algorithmes ont été proposés pour bénéficier de l'information "liens hypertextes" pour mesurer la pertinence des pages Web, les plus cités sont PageRank proposé par Sergey Brin & Lawrence Page (Brin et al., 1998) et HITS (Hyperlinked Induced Topic Selection) proposé par Kleinberg (Kleinberg, 1999).
La problématique que nous traitons dans ce papier concerne l’utilisation des liens comme source d’évidence dans le contexte de la RIS (recherche d’information structurée). La RIS (ou bien RI dans des documents XML) vise à renvoyer à l’utilisateur des réponses d’une granularité plus fine que le document entier. Cette granularité s’appelle élément XML et l’évaluation de ces éléments renvoyés se fait selon deux dimensions qui sont : la spécificité et l’exhaustivité.
Notre problématique peut être définie par ces deux questions :

- Est ce que l’exploitation des liens comme source d’évidence dans le contexte de la recherche d’information dans des corpus de documents XML, en l’occurrence la collection Wikipedia (Wikipedia, 2008), permet d’améliorer la qualité des résultats ?
- Est-ce que les algorithmes utilisés par la RI sur le Web peuvent être exploités ou bien adaptés dans le contexte de la RIS ?

Pour répondre à ces questions nous avons menés des expérimentations afin d’introduire la source d’évidence "liens XML" dans le calcul de la pertinence des éléments renvoyés.

L’article est organisé comme suit : nous commençons en section 2 par l’état de l’art des travaux relatifs à l’exploitation des liens en RIS. Dans la section 3 nous détaillons notre méthode d’utilisation des liens dans des corpus de documents XML. Les résultats d’expérimentations obtenus pour la tâche "Focused" d’INEX 2007 seront présentés dans la section 4. Enfin, nous concluons dans la section 5.

2. Travaux relatifs

Dans le contexte du World Wide Web et de la collection Wikipedia (Wikipedia, 2008), les liens sont une importante source d’évidence (Jaap *et al.*, 2008). Les deux algorithmes les plus connus qui utilisent cette source d’évidence pour améliorer la qualité des résultats renvoyés à l’utilisateur sont : PageRank (Brin *et al.*, 1998) et HITS (Kleinberg, 1999).

Peu de travaux ont été proposés pour l'exploitation des liens en recherche d'information dans des documents XML. L'un des premiers travaux est celui de Lin G. et al. (Lin *et al.*, 2003) qui proposent une méthode (appelée XRANK) permettant la prise en compte des liens XML pour le ré-ordonnancement de la liste des résultats. Dans leur méthode le score d'un élément est en fonction de trois scores relatifs aux ensembles CE, HE, CE^{-1} (CE : liens hiérarchiques entre nœuds, HE : liens Xlink entre nœuds et CE^{-1} : le même ensemble CE sauf que le sens des liens est inversé). Khairun N. F. et al. (Khairun *et al.*, 2008), Jaap K. et Marijn K. (Jaap *et al.*, 2008) utilisent les liens XML pour le "rerank" (reclassement ou ré-ordonnancement) des résultats renvoyés selon deux degrés : "local indegree" et "global indegree". Le premier représente le nombre de liens de la collection entrants à un article et le deuxième degré représente le nombre de liens entrants à un article à partir des documents renvoyés comme résultats à un topic donné. Benny K. et al. (Benny *et al.*, 2007) appliquent l'algorithme HITS sur les Top-N documents renvoyés pour filtrer les résultats renvoyés à l'utilisateur. Jovan P. et al. (Jovan *et al.*, 2008) utilisent aussi les liens dans le contexte de la tâche "entity ranking" d'INEX 2007.

Ces trois derniers travaux proposent des méthodes basées sur des adaptations de HITS au contexte de collections de documents XML. La méthode que nous proposons repose par contre sur une adaptation de l'algorithme PageRank (Brin et al., 1998).

3. Notre approche

3.1. Motivation

L'intuition qui motive nos propositions est la suivante : si un document est référencé par plusieurs documents importants de la collection alors ceci peut donner un signe sur son importance, cette importance du document aura par conséquence un impact sur les scores des éléments renvoyés par un système de recherche appartenant à ce document.

La figure suivante montre un graphe de liens entre quelques documents de la collection Wikipedia extraits comme réponses à la requête 537 ainsi que quelques liens avec des documents qui ne sont pas renvoyés comme réponses à cette requête.

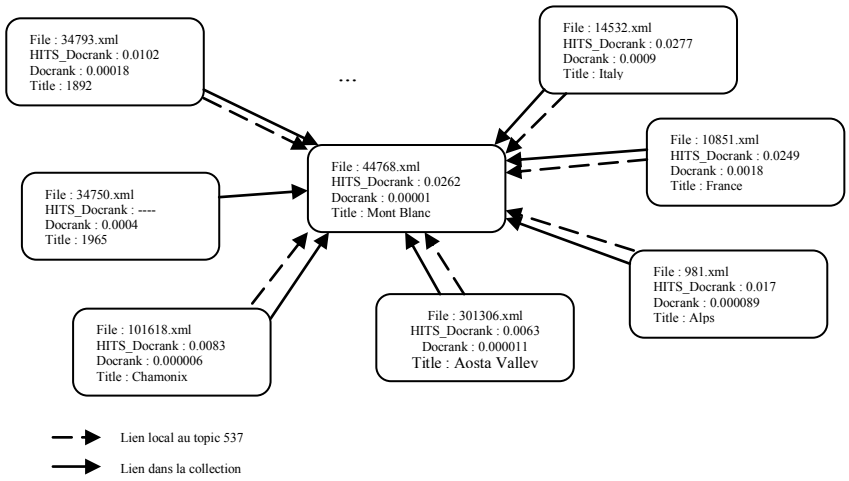


Figure 1. *Exemple d’un graphe de liens issus de la collection Wikipedia avec les valeurs DOCRANK et HITS_Docrank calculées pour le "TOPIC 537".*

Ce graphe donne une idée sur la structure des liens entre les documents de la collection Wikipedia qui sont d’une nature sémantique. Dans le topic 537 qui a comme titre "pictures of Mont blanc" nous remarquons que plusieurs documents renvoyés par le système de recherche de l’université DALIAN pointent sur l’article "44768.xml" qui a comme titre "Mont blanc" , ce qui traduit le score élevé qui lui a été affecté par l’application de notre approche. Si nous introduisons le score affecté à l’article "44768.xml" ça ne va qu’augmenter son score final, et par conséquence les scores des éléments qui lui y appartiennent, ce qui va améliorer la qualité des résultats.

3.2. Détails de l’approche :

Notre idée est d’utiliser les liens pour calculer un nouveau score pour les éléments renvoyés par un système de recherche dans des documents XML. Notre approche se base sur une adaptation du PageRank au contexte de collections de documents XML.

Le DOCRANK d’un document XML *D* dans une collection de documents XML est calculé selon la formule qui suit :

$$DOCRANK(D) = \frac{1-d}{Nbr\_docs\_coll} + d * \sum_{(i,D)\in links} DOCRANK(i) \quad [1]$$

Où *links* représente l’ensemble des paires de liens *(i,j)* internes à la collection tel que le document *i* contient un lien vers le document *j*. *Nbr_docs_coll* represente le

nombre de documents de la collection et *d* représente le facteur d'amortissement. Le calcul des DOCRANKs se fait en offline. Le HITS_docrank est calculé avec la même formule sauf qu'il est "Query-dependant" (le calcul se fait au moment de la requête) et sur un sous ensemble des résultats renvoyés (le paramètre *Nbr_docs_coll* va représenter le sous ensemble de documents utilisés pour le calcul).
Ce calcul est fait d'une manière itérative suivant le même principe de PageRank jusqu'à la convergence des valeurs DOCRANK ou HITS_docrank.

Le nouveau score d'un élément renvoyé E_i en tenant compte des liens comme source d'évidence est calculé selon les formules suivantes :

$$Nouv\_Score\_DOCRANK(E_i) = \alpha * Score\_initial(E_i) + (1-\alpha) * DOCRANK(D) \quad / E_i \in D \qquad [2]$$

$$Nouv\_Score\_HITS\_docrank(E_i) = \alpha * Score\_initial(E_i) + (1-\alpha) * HITS\_docrank(D) \quad / E_i \in D \qquad [3]$$

Où "*Nouv_Score_DOCRANK*" et "*Nouv_Score_HITS_docrank*" représentent les nouveaux scores calculés pour l'élément E_i, *Score_initial* représente le score initialement affecté par le système de recherche à l'élément E_i. Enfin, *DOCRANK(D)* et *HITS_docrank(D)* sont les scores calculés selon la formule [1] pour le document XML D auquel E_i appartient. α est un paramètre qui permet de définir le degré de contribution des différents scores dans le score final.

Toutes les valeurs des scores initiaux ainsi que les scores DOCRANKs et HITS_docrank sont normalisées avant de calculer les nouveaux scores des éléments. La normalisation des scores DOCRANKs et HITS_docrank sert à éliminer l'effet de la grande différence entre ces derniers et les scores initiaux calculés par le système de recherche.

4. Résultats d'expérimentation

4.1. Conditions Expérimentales

Configuration matérielle et logicielle :

Nous avons utilisé pour nos expérimentations un PC HP doté de 2 GO de RAM et d'un cache de 4 MO (niveau 2) et d'un disque dur de 160 GO. Le SGBD utilisé pour le stockage de l'index de la collection Wikipedia est ORACLE 11g Entreprise Edition. Pour l'indexation, nous avons utilisé le système XFIRM (Sauvagnat, 2005) développé à l'IRIT.

Collection de test :

En 2006, Denoyer et Gallinari (Denoyer *et al.*, 2006) ont créé un corpus de documents XML basé sur une partie de l’encyclopédie libre Wikipedia. L’actuel corpus XML, utilisé dans la compagne d’évaluation INEX, de Wikipedia contient plus de 650,000 documents XML en langue anglaise.

Cette collection est caractérisée par les liens qui sont d’une nature sémantique, car ils sont basés sur l’apparition des mots dans le contenu du document, c.-à-d. que si un mot représente une thématique traitée par un article de la collection il représentera automatiquement un lien vers cet article, exemple le mot "Algeria" va faire l’objet d’un lien vers l’article traitant le sujet "Algeria".

Prétraitement :

Le graphe de liens de la collection que nous avons utilisée contenait 17,039,174 liens élément-documents, sachant que les liens de Wikipedia ne pointent que sur les racines des documents et non pas sur des éléments internes aux documents. L’étape prétraitement consiste à construire le graphe "document-document" qui nous permet par la suite d’appliquer notre algorithme de calcul de DOCRANK et de HITS_docrank. Cette étape de prétraitement a résulté un graphe contenant 659,388 nœuds (qui représentent les documents de la collection Wikipedia) et 13,611,471 liens "documents-documents".

La première étape dans l’exécution de l’algorithme DOCRANK, qui est une adaptation de (Haveliwala, 1999) au contexte de la collection Wikipedia, est de monter la matrice du graphe de liens en mémoire. L’algorithme converge au bout de 76 itérations avec un seuil de convergence fixé à 1e-8. L’exécution de cette étape a duré environ 30 minutes. Pour le HITS_docrank la convergence des scores des articles se fait au bout de quelques millisecondes pour chaque topic.

Nos expérimentations ont été effectuées sur les résultats renvoyés par les trois systèmes les mieux classés pour la tâche "Focused" d’INEX 2007 (INEX, 2007), en l’occurrence DALIAN University of technology, University of WATERLOO et MAX-Planck institut fur informatik. Ces résultats concernent les topics co414 à co543 (107 topics CO (Content Only : requêtes orientées contenu) au total). La valeur prise pour le paramètre *d* est 0.85.

4.2. Résultats

Dans cette section nous présentons les résultats d’expérimentations obtenus après évaluation pour les trois systèmes en l’occurrence DALIAN University of technology, University of WATERLOO et MAX-Planck institut fur informatik par rapport à l’application du DOCRANK et HITS_docrank sur les éléments renvoyés par ces systèmes.

Comme nous l’avons déjà mentionné nos expérimentions sont effectuées sur la base des résultats renvoyés par les trois systèmes précédemment cités pour la tâche

"Focused" d'INEX 2007. Cette tâche qui s'intéresse aux éléments les plus spécifiques à la requête de l'utilisateur et qui ne soient pas imbriqués les uns dans les autres. Donc nous présentons à cet effet les résultats obtenus pour la mesure iP[0.01] (qui représente la précision interpolée au niveau de rappel 0.01) comme recommandé à INEX 2007.

DALIAN University (DUT_03_Focused)	DOCRANK	Hits_docrank Tous les documents	Hits_docrank 150 premiers documents	Hits_docrank 50 premiers documents	Hits_docrank 20 premiers documents
BaseRun	0.5271	0.5271	0.5271	0.5271	0.5271
α = 0.1	0.4533	0.3512	-	-	-
α = 0.5	0.5228	0.4183	-	-	-
α = 0.8	0.5274	0.5221	**0.5305**	0.5343	**0.5470 ***
α = 0.9	**0.5275**	**0.53**	0.5296	**0.5356**	0.5351
% d'amélioration	0.08%	0.55%	0.65%	1.61%	3.78%
* valeur *t-test* = 0.026 = 2.6%					

Tableau 1. *Valeurs iP[0.01] obtenues après application de DOCRANK et de HITS_docrank sur les résultats renvoyés par le système de l'université de DALIAN pour plusieurs variation du paramètre α*

Le tableau 1 représente les résultats obtenus après application de DOCRANK et HITS_docrank (sur plusieurs niveaux selon le nombre de documents utilisés dans le calcul) avec variation du paramètre α (voir formule [2] et [3]).
L'application du DOCRANK n'as pas prouvé une amélioration significative (0.08% dans le meilleur des cas : avec α égale à 0.9), ce qui veut dire que les liens dans le contexte global de la collection ne permettent pas d'améliorer les résultats, mais plutôt le contraire (sauf dans le cas où α est égale à 0.9). Ceci est du aux documents de la collection qui ont un score DOCRANK élevé et par conséquence des rangs élevés dans tous les topics dans lesquels ils apparaissent comme résultats. Un des exemples que nous avons rencontré durant nos expérimentations est le document "31882.xml" qui traite le sujet "United states", dans plusieurs requêtes dont le sujet n'a rien à voir avec "United states" et pour lequel des éléments appartenant au document "United states" (31882.xml) apparaissent (parce qu'ils contiennent un des mots de la requête) dans la liste des résultats et après application du DOCRANK ils auront des scores qui vont augmenter et par conséquence de meilleurs rangs, ce qui diminue la qualité des résultats dans certaines requêtes (topic). Donc, c'est ce phénomène d'infiltration des documents non pertinents qui cause la diminution de la qualité des résultats.
L'autre remarque que nous avons pu constater est que l'augmentation de la valeur du paramètre α (en d'autres termes l'impact du score DOCRANK et HITS_docrank est diminué, voir formules [1] et [2]) rend la qualité meilleure, ce qui veut dire que l'information textuelle (les scores initialement attribués aux éléments) reste importante par rapport à l'information liens XML.

Le tableau comporte aussi les résultats obtenus après application du HITS_docrank. Ces résultats sont meilleurs par rapport à ceux obtenus avec DOCRANK pour toutes les variations de α et le meilleur taux d'amélioration est celui obtenu pour α est égale à 0.8 avec les 20 premiers documents retournés pour chaque topic. Le meilleur taux d'amélioration obtenu est de 3.78%.
Ceci peut être traduit par le fait de diminution du phénomène d'infiltration des documents non pertinents que nous avons déjà cité (en d'autres termes si un document est pointé dans l'ensemble de la collection avec 1000 liens, il ne sera pointé que par 19 documents au maximum dans l'ensemble des 20 premiers documents, et ces documents sont considérés comme étant les meilleurs pour le topic en question).
Pour confirmer qu'il s'agit d'un taux significatif, nous avons calculé le *t-test* (Student Test) pour l'ensemble des 107 topics. La valeur *t-test* obtenue est égale à 0.026 (2.6%), ce qui confirme que l'amélioration est significative même si elle est relativement faible.
Pour confirmer les améliorations obtenues après application de HITS_docrank sur les résultats renvoyés par le système de l'université DALIAN, nous l'avons appliqué sur deux autres systèmes classés parmi les trois meilleurs systèmes à INEX 2007.

WATERLOO University	Hits_docrank 150 premiers documents	Hits_docrank 50 premiers documents	Hits_docrank 20 premiers documents
BaseRun	0.5108	0.5108	0.5108
α = 0.1	0.394	0.4425	0.4899
α = 0.8	0.4992	0.4948	**0.5218**
α = 0.9	0.5100	**0.5135**	0.5001
Meilleur Taux d'amélioration	-0.16%	0.53%	2.15%

Tableau 2. *Valeurs iP[0.01] obtenues après application de HITS_docrank sur les résultats renvoyés par le système de l'université de WATERLOO*

Le tableau 2 montre les valeurs de iP[0.01] obtenues après application de HITS_docrank sur les résultats du système de l'université de WATERLOO. Ces résultats confirment celles du premier tableau, et le meilleur taux d'amélioration est obtenu avec les mêmes paramètres que le premier système (c'est-à-dire α=0.8 et nombre de documents = 20 documents).

MAX-PLANCK Institut	Hits_docrank 150 premiers documents	Hits_docrank 50 premiers documents	Hits_docrank 20 premiers documents
BaseRun	0.5066	0.5066	0.5066
α = 0.1	0.3775	0.4366	0.4646
α = 0.8	0.4822	0.4792	0.4954
α = 0.9	0.5000	0.5027	**0.5072**

Meilleur Taux d'amélioration	-1.30%	-0.77%	0.12%

Tableau 3. *Valeurs iP[0.01] obtenues après application de HITS_docrank sur les résultats renvoyés par le système de l'institut MAX-PLANCK*

Le tableau 3 représente les valeurs iP[0.01] obtenues après application de HITS_docrank sur les résultats renvoyés par le système de l'institut MAX-PLANCK.

Les taux d'amélioration sont moins significatifs par rapport aux taux obtenus avec les deux systèmes précédents. Ceci est du à la stratégie de recherche adoptée par le système de l'institut MAX-PLANCK. Cette stratégie repose sur l'information "CAS-title" des topics. Ce qui élimine beaucoup de documents de la liste des top-N éléments renvoyés parce qu'ils ne respectent pas les contraintes structurelles citées dans le "CAS-title" des topics. Nous avons constaté à ce propos que dans la plupart des topics il n'existait pas de liens entre les top-N documents renvoyés ce qui justifie la qualité des résultats obtenus après application de HITS_docrank.

5. Conclusion

Cet article décrit les résultats d'expérimentation obtenue après application de deux propositions, en l'occurrence DOCRANK et HITS_docrank, qui permettent d'introduire les liens comme source d'évidence pour réordonner la liste des éléments renvoyés par un système de recherche d'information dans des documents XML, ceci dans le but d'améliorer la qualité de la recherche.

Nos expérimentations ont été effectuées sur les résultats renvoyés par trois systèmes les mieux classés dans la compagne INEX 2007 pour la tâche "Focused" sur l'ensemble des 107 topics CO.

Les résultats obtenus nous ont permis de mesurer l'impact de deux facteurs qui sont : la variation du paramètre α et le nombre de documents utilisés pour le calcul de *HITS_docrank*. Ces résultats montrent que les liens représentent une information importante qui a permis d'améliorer la qualité de la recherche.

Cependant, il serait préférable de proposer des méthodes qui permettent d'éviter le phénomène relatif aux liens issus des documents non pertinents. D'autres propositions peuvent aussi faire l'objet de traitement des liens spécifiques à XML, c'est-à-dire liens élément-élément, sachant que la collection Wikipedia actuelle d'INEX ne comporte pas encore ce type de liens.

6. Bibliographie

Benny Kimelfeld, Eitan Kovacs, Yehoshua Sagiv, Dan Yahav, *Using Language Models and the HITS Algorithm for XML Retrieval*, In INEX 2006, pp. 253–260, Heidelberg, 2007.

Brin, S., Page, L., *The anatomy of a large-scale hypertextual Web search engine*. In: Proceedings of the 7th International Conference on World Wide Web, Brisbane, Australia, pp. 107–117, 1998.

Denoyer, L., Gallinari, P., *The Wikipedia XML corpus*. SIGIR Forum 40(1), pp. 64–69, 2006.

Taher H. Haveliwala, *Efficient Computation of PageRank*, Technical Report, Stanford University, October 18, 1999.

INEX, *INitiative for the Evaluation of XML retrieval*, http://inex.is.informatik.uni-duisburg.de/2007, 2007.

Jaap Kamps and Marijn Koolen, *The Importance of Link Evidence in Wikipedia*, In : Lecture Notes in Computer Science, pp. 270-282, Heidelberg, 2008.

Jovan Pehcevski, Anne-Marie Vercoustre, and James A. Thom, *Exploiting Locality of Wikipedia Links in Entity Ranking*, In : ECIR 2008, pp. 258–269, Heidelberg, 2008.

Khairun Nisa Fachry, Jaap Kamps, Marijn Koolen, and Junte Zhang, *Using and Detecting Links in Wikipedia*, In : Focused Access to XML Documents, pp. 388-403, 2008.

Kleinberg, J.M., *Authoritative structures in a hyperlinked environment*. Journal of the ACM 46, pp. 604–632,1999.

Lin, G., Feng, S., Chavdar, B., Jayavel, S., *XRANK : Ranked Search over XML Documents*. In : SIGMOD’2003, San diego,CA, 2003.

Sauvagnat Karen, *Modèle flexible pour la Recherche d’Information dans des corpus de documents semi structurés*. Thèse de doctorat, IRIT, Université Paul Sabatier de Toulouse, 2005.

Wikipedia: *The free encyclopedia* (2008), http://en.wikipedia.org/

Structure et proximité pour la recherche documentaire

Michel Beigbeder

École Nationale Supérieure des Mines de Saint-Étienne
158, cours Fauriel
F-42023 Saint-Etienne cedex 2
mbeig@emse.fr

RÉSUMÉ. *Notre étude compare les performances d'un système de recherche d'information basé sur la proximité des occurrences des termes de la requête dans les documents avec un système classique de modèle de langue avec lissage de Dirichlet et le modèle Okapi BM25. Notre modèle basé sur la proximité calcule en chaque position du document une valeur d'autant plus grande que des occurrences de tous les termes de la requête sont proches de cette position. De plus pour le modèle à proximité nous testons dans le cas de documents structurés l'hypothèse que les termes apparaissant dans les titres doivent être considérés comme proches des positions de toute la section correspondant à ce titre.*

ABSTRACT. *Our study compares the effectiveness of an information retrieval system based on the proximity of the query term occurrences in the documents and an IRS based on a language model with Dirichlet smoothing and with the Okapi BM25 model. Our proximity based model computes at each position in the document a value much higher as some occurrences of all the query terms are close to this position. Moreover for the proximity based model we are testing the assumption that the title terms are to be considered as close to all the positions of the whole corresponding section.*

MOTS-CLÉS : *Recherche d'information, documents structurés, proximité des termes, logique floue.*

KEYWORDS: *Information retrieval, structured documents, term proximity, fuzzy logic.*

1. Introduction

La plupart des modèles de recherche d'information n'utilisent que des informations statistiques sur les documents pour leur attribuer un score de similarité avec la requête. Ainsi deux documents qui utilisent le même vocabulaire avec la même distribution du nombre des occurrences des termes ne sont pas distinguables par les fonctions d'attribution de score de ces modèles. Nous présentons ici notre modèle de correspondance qui utilise les positions des occurrences des termes. Pour une requête demandant les termes A et B, un modèle statistique calcule le même score pour deux documents qui contiennent une occurrence de chacun de ces termes quelles que soient leur position. Notre modèle calcule un score d'autant plus élevé que ces deux occurrences de ces deux termes sont proches. Il s'agit ici de l'utilisation de la première structure de tout texte qui est celle de la séquentialité des termes. Par ailleurs de nombreux documents, et en particulier tous les documents scientifiques et techniques, ont une structure logique hiérarchique composée de sections avec des titres. Ces titres décrivent d'une certaine façon le contenu de toute la section qui leur correspond. Cette dernière hypothèse rapportée dans un sytème de recherche d'information à base statistique va faire donner plus d'importance aux mots des titres, importance qui va se traduire en terme de poids. Dans notre modèle de proximité, la traduction de cette hypothèse est que les termes lorsqu'ils apparaissent dans un titre ont une influence qui se propage à toute la section qui correspond à ce titre.

Dans cet article, nous détaillons notre modèle à base de proximité pour les textes plats et son extension pour prendre en compte la propagation de l'influence des termes des titres. Nous présentons ensuite l'implémentation et la collection de test qui nous permettent de mettre en place une expérience de recherche d'information. Enfin nous présentons les résultats de performance de ce système dans les configurations plate et structurée et les comparons avec ceux obtenus par deux systèmes basés sur des statistiques, le modèle de Dirichlet et le modèle Okapi BM25.

2. Modèle de proximité

2.1. *Proximité à un terme*

Nous modélisons les textes comme des fonctions qui associent des positions — des entiers naturels — avec des termes du vocabulaire. Ce qui revient à numéroter les occurences des termes au fur et à mesure de la lecture du texte. Ce sont ces entiers qui servent à la numérotation que nous appelons dans la suite des *positions*.

Notre idée de base pour utiliser la proximité dans la recherche d'information consiste à définir une fonction définie pour chaque position d'un texte. L'idée est que la valeur de cette fonction soit d'autant plus grande que l'on est proche de tous les termes de la requête. Nous allons développer cette idée en examinant successivement le cas d'un terme puis de la combinaison des termes dans la requête.

Considérons d'abord le cas d'un terme de la requête et d'une occurrence de ce terme. La valeur à attribuer à une position doit être décroissante par rapport à la distance x à cette occurrence de ce terme, et de plus nous la gardons positive. En choisissant la fonction $x \mapsto \frac{\max(k-x,0)}{k}$ cette valeur peut être interprétée comme une proximité *floue* à l'occurrence du terme en cette position. La valeur du paramètre k définit la demie-base du triangle de cette fonction « triangulaire », il s'interprète donc comme la *portée* de l'occurence, puisque pour des positions à une distance plus grande que k la valeur de proximité floue est nulle. Dans nos expériences, nous prenons $k = 200$, ce qui correspond à une longueur moyenne de paragraphe. (Ce paragraphe comporte environ 120 mots).

Toujours pour un terme de la requête, lorsque plusieurs occurrences du terme sont considérées, la valeur de proximité doit être le maximum des valeurs de proximité à toutes les occurrences de ce terme ; avec la condition de décroissance pour la proximité par rapport à une occurrence du terme cela revient à dire que la proximité à un terme en une position est la proximité floue à la plus proche des occurrences de ce terme.

2.2. *Combinaison de proximités*

Lorsqu'une requête comporte plusieurs termes se pose la question de leur combinaison. Dans la majorité des modèles de recherche d'information cette combinaison est plutôt disjonctive. Ce qui signifie que le modèle n'impose pas que tous les termes soient présents dans les documents pour être retrouvés. Les outils de recherche sur le Web ont plutôt un comportement conjonctif, exigeant donc la présence de tous les termes ; ceci est probablement dû à l'avalanche de documents qui vérifient déjà cette contrainte. Nous avons choisi dans notre modèle de reporter ce choix au niveau de la requête en travaillant avec des requêtes booléennes. Ce choix est aussi naturel par rapport à la notion de proximité, puisqu'il y a une différence entre être proche de A et de B d'une part, et être proche de A ou de B d'autre part. La traduction de cette différence dans notre modèle est simple, pour une combinaison disjonctive, nous prenons le max des fonctions de proximité, et pour une combinaison conjonctive, nous prenons leur min ; ce qui correspond aux fonctions de combinaison des opérateurs de réunion et d'intersection des ensembles flous. Nous pouvons aussi considérer le cas des négations : la proximité à la négation d'un terme (d'une requête) se modélise comme 1 moins la proximité à ce terme (à cette requête).

2.3. *Modèle de requête*

Nous travaillons donc avec des requêtes booléennes avec les traditionnels opérateurs pour la conjonction, la disjonction et la négation. Nous pouvons aussi traiter des expressions (*phrases* en anglais). Pour celles-ci, nous considérons qu'une expression a une occurrence d'apparition à la position de son dernier terme.

Si une requête est une simple liste de termes, on peut la considérer au choix soit comme une requête purement disjonctive soit comme purement conjonctive. Puisque l'idée de notre modèle à travers les fonctions de pertinence locale que nous avons introduites est de donner des scores plus élevés aux documents lorsque les différents termes de la requête sont proches, le choix en l'absence d'opérateurs explicites doit se porter sur la conjonction.

2.4. *Score de pertinence d'un document*

La fonction que nous avons définie permet d'attribuer en chaque position du document une valeur qui représente la proximité à la totalité de la requête. Pour le cas conjonctif, cette proximité est donc grande quand une position est proche de tous les termes de la conjonction. Indirectement, on mesure donc la distance des occurrences des termes à cette position.

On peut interpréter cette valeur de proximité en une position comme une pertinence de cette position par rapport à la requête, et nous l'appelerons *pertinence locale*. En effet, pour le cas trivial d'un terme, on ne peut trouver de meilleur endroit pertinent à la requête dans un document que les endroits où se trouvent précisément ce terme. C'est d'ailleurs utilisé par de nombreux outils de recherche qui mettent en évidence les termes de la requête lors de la présentation des réponses, typiquement en surlignant en couleurs vives leurs occurrences.

Avec cette interprétation, pour donner une valeur de pertinence au document nous calculons la moyenne des pertinences locales de toutes ses positions. Cela revient à intégrer (sommer) toutes les valeurs de pertinence locale sur la longueur du document, puis à normaliser cette intégrale.

2.5. *Implémentation du calcul de la pertinence locale à un terme*

D'un point de vue pratique, une requête booléenne est représentée par un arbre, et nous avons déjà indiqué comment remonter les fonctions de pertinences locales sur les nœuds internes grâce aux opérateurs booléens des ensembles flous. La fonction de proximité locale est renseignée pour chaque feuille de l'arbre de la requête de la façon suivante. Pour une feuille de l'arbre de la requête un tableau dimensionné à la taille du document est alloué et initialisé à zéro. Pour chaque occurrence du terme (ou de l'expression) associé(e) à la feuille, la fonction triangulaire qui atteint son maximum 1 à la position de l'occurrence est combinée par la fonction max dans le tableau. Lorsque toutes les occurrences ont été prises en compte, le tableau contient la fonction de proximité floue (ou de pertinence locale) au terme.

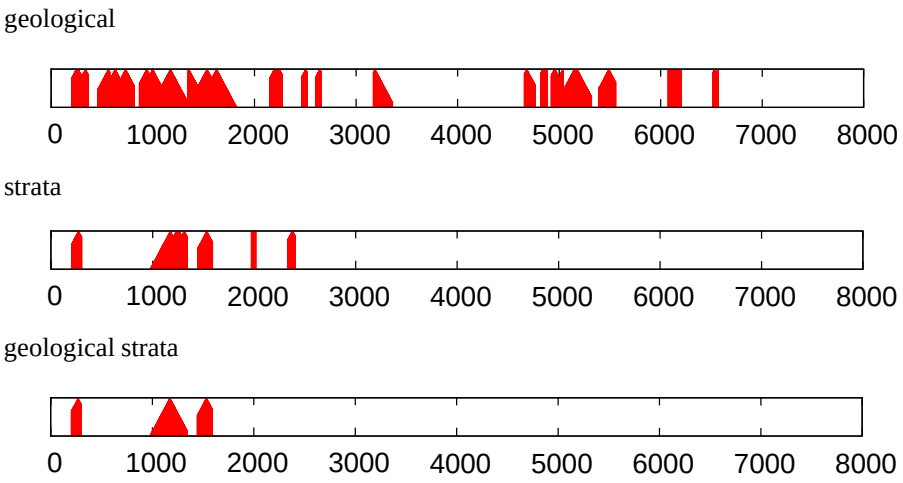


Figure 1. *Les pertinences locales au terme* `geological`*, au terme* `strata` *et à la combinaison conjonctive* `geological & strata` *dans le fichier 543667 de INEX, en prenant en compte les éléments sectionnants et propageants.*

2.6. *Propagation des termes de titre*

Nous présentons dans cette section une idée pour prendre en compte la structure des documents dans le modèle à base de proximité que nous avons présenté dans la section précédente dans le cas des textes plats. De la structure des documents nous ne considérons que ce qui concerne la hiérarchie, typiquement les chapitres, section, sous-sections, . . ., paragraphes. Nous ignorons donc d'autres aspects comme les listes, les tableaux, etc. Cette structure hiérarchique nous donne donc une arborescence liée à l'imbrication des éléments de niveaux inférieurs dans les éléments de niveaux supérieurs. Associé à cette notion de hiérarchie il y a la notion de titre, lequel décrit — certes d'une façon plus ou moins explicite — le contenu de la section qu'il intitule.

Du point de vue de la proximité, avec l'hypothèse que ces mots du titre décrivent le contenu de la section, l'idée est que les termes qui apparaissent dans un titre soient « proches » des termes qui apparaissent dans le corps de la section. Nous avons vu précédemment que la proximité était une conséquence d'une pertinence locale qui décroit avec la distance aux occurrences. Réciproquement pour être proche sur tout le corps d'une section l'idée est donc d'avoir une pertinence locale maximale — de valeur 1 — pour un terme du titre sur tout le corps de la section correspondante.

La structure effective des documents que nous manipulons contient bien d'autres types d'éléments que ceux de la hiérarchie et des titres. Nous classons donc les balises en trois catégories, celles qui définissent les éléments qui interviennent dans la hiérarchie — les *éléments sectionnants* —, celles des éléments qui contiennent un titre — les *éléments propageants* —, et enfin les autres balises.

2.7. *Implémentation du calcul de la pertinence locale à un terme*

Prendre en compte la structure du document dans les calculs des fonctions de pertinence locales n'intervient que pour les feuilles. En effet, pour tous les nœuds internes, il ne s'agit comme précédemment que de combiner les fonctions de pertinence locales des fils du nœud.

Comme pour le cas du texte plat, on considère successivement pour une feuille toutes les occurrences du terme associé. Pour une occurrence il faut trouver dans quel élément elle se trouve. Pour cela, nous utilisons une représentation de la structure où pour chaque élément sont conservées les positions des occurrences des premiers et derniers termes appartenant au contenu textuel de l'élément, y compris si ces contenus textuels sont en fait dans des éléments fils. Nous utilisons la structure de stockage de ces informations décrites dans [BEI 08].

Par un parcours récursif dans l'arbre de la structure du document, on trouve donc l'élément auquel appartient une occurrence. À partir de cet élément on remonte dans l'arbre jusqu'à trouver soit un élément sectionnant, soit un élément propageant. Si l'élément ainsi trouvé est un élément sectionnant, on introduit dans le tableau de la fonction de proximité locale le triangle en le tronquant aux limites de l'élément sectionnant. Si au contraire l'élément trouvé est un élément propageant, on continue de remonter dans l'arbre de la structure jusqu'au prochain élément sectionnant. Le tableau de la fonction de proximité locale est alors rempli avec la valeur maximale 1 sur toute la longueur de cet élément sectionnant.

La figure 1 montre trois fonctions de pertinence locale dans le document 543667 de la collection INEX (cf. infra). La première fonction montre la pertinence locale (la proximité floue) au terme `geological`. On y voit des triangles dont certains se recouvrent, donnant l'aspect en « montagne ». On y voit aussi des triangles tronqués, indiquant que la portée d'une occurrence du terme `geological` a été limitée par son appartenance à un élément sectionnant. On y voit enfin des rectangles, indiquant qu'une occurrence de `geological` est apparue dans un titre et que la portée a été étendue uniformément à tout l'élément sectionnant englobant. La deuxième fonction montre la pertinence locale (la proximité floue) au terme `strata` et la troisième est le min des deux précédentes, puisque les deux termes sont combinés conjonctivement.

3. Implémentation

3.1. *Indexation*

La base de l'implémentation est bien sûr d'indexer les documents en gardant les positions de toutes les occurrences des termes dans les documents au fur et à mesure de leur analyse. Pour celà nous utilisons le logiciel *zettair*[1] qui a cette fonctionnalité.

1. http://www.seg.rmit.edu.au/zettair/

Nous avons implanté dans ce logiciel l'analyse et le stockage au moment de l'indexation de la structure arborescente du XML.

3.2. *Langage de requête*

Notre modèle de calcul de scores est basé sur des requêtes booléennes. Nous avons donc besoin d'un langage de requête qui permet d'introduire les opérateurs de conjonction, de disjonction et de négation par rapport aux termes et aux expressions que l'on retrouvera dans les feuilles. Le caractère `'|'` sert d'opérateur de disjonction, les caractères `' '` et `'&'` servent d'opérateur de conjonction, et le caractère `'-'` pour la négation. De plus les parenthèses permettent de contrôler les priorités entre les opérateurs et les guillemets `'"'` d'introduire les expressions. Nous avons écrit un analyseur pour ce langage de requête et l'avons inséré dans *zettair*. Cet analyseur construit un arbre pour la requête qui est ensuite passé à l'implémentation de notre modèle de calcul de score.

Le langage de requête de la version initiale de *zettair* reconnaît à la base les opérateurs `'-'` pour indiquer une préférence pour l'absence du terme qui suit dans les réponses (une sorte de négation, bien que moins forte que la négation du modèle booléen) et l'opérateur `'"'` pour introduire les expressions. Par ailleurs, il ignore les autres caractères non alphanumériques. Ainsi, une requête booléenne dans le langage de requête défini ci-dessus peut être utilisée par *zettair* avec ses modèles classiques (Okapi, Dirichlet, etc.). Simplement les `'|'`, `'&'` et les parenthèses sont ignorés.

Réciproquement, notre choix d'utiliser aussi l'espace comme opérateur de conjonction nous permet d'utiliser quasiment telles quelles des requêtes usuelles pour *zettair* avec notre modèle basé sur la proximité, avec toutefois une interprétation conjonctive au lieu de l'interprétation disjonctive des modèles classiques.

3.3. *Interrogation*

Avec ces données, nous avons implanté toujours dans ce logiciel le modèle de correspondance basé sur la proximité que nous venons de décrire. Comme nous l'avons déjà mentionné, la différence entre la version qui prend en compte la structure et celle qui travaille sur le texte plat n'est que dans le remplissage des tableaux représentant les fonctions de proximité floue des feuilles de l'arbre de la requête booléenne.

De plus notre modèle lorsqu'il est utilisé en mode conjonctif est extrêment sélectif, puisque pour qu'un document ait un score non nul il faut non seulement qu'il contiennent TOUS les termes de la requête conjonctive mais en plus qu'il y ait des occurrences de ces termes proches dans les documents. Lorsqu'il est utilisé seul, les listes de réponses sont généralement très courtes. Pour les compléter jusqu'à la limite des *runs* que l'on peut soumettre dans les campagnes de recherche d'information (typiquement 1000 ou 1500 selon les campagnes) elle sont complétées par les résultats obtenus par le modèle de base de *zettair* (modèle de langue avec lissage de Dirichlet).

4. Expérience

Nous avons construit notre expérience selon la méthodologie traditionnelle en recherche d'information ad'hoc avec une collection de test. Les données de cette collection sont celles utilisées dans l'édition 2008 de la campagne INEX. Les documents sont ceux issues d'une collecte de l'encyclopédie collaborative *Wikipedia* de 2006. Cet ensemble de documents a été utilisé dans les campagnes INEX depuis 2006. Il est composé de 659 388 documents qui correspondent à autant d'entrées dans l'encyclopédie en ligne. Ces documents sont formattés en XML et utilisent 1 263 types d'éléments différents.

Parmi ces éléments nous avons sélectionné manuellement ceux qui sont propageants : `name`, `template`, `title` et `caption` ; et ceux qui sont sectionnants : `article`, `body`, `section`, `figure`, `image`, `page` et `div`.

La campagne 2008 de INEX a utilisé 285 besoins d'information dont 135 ont été développés par les participants pour cette campagne, les 150 autres sont issus du journal d'un moteur d'interrogation de la *Wikipedia*. Les besoins d'informations sont composés des trois champs : `<title>`, `<description>` et `<narrative>` ; et parfois un quatrième : `<castitle>`. Les trois premiers sont analogues à ceux utilisées dans les jeux de *topics* utilisés dans les campagnes TREC. Le quatrième est spécifique à INEX et contient une requête dans le langage NEXI qui permet d'introduire des aspects de structure des documents dans la formalisation du besoin d'information.

Pour construire les requêtes pour notre système nous nous sommes basés sur le champ `<title>` qui est une requête composée de mots-clés, avec quelques opérateurs, `'+'` qui renforce la demande de présence du terme qui le suit dans un document retrouvé, `'−'` qui recommande l'absence du terme qui le suit, et les guillemets qui permettent d'introduire des expressions dans le besoin d'information.

Avec le langage de requête que nous avons défini ci-dessus, le champ `<title>` est directement utilisable comme requête pour notre implémentation aussi bien en mode de recherche par proximité que dans les modes de recherche classiques. Dans quelques cas, nous avons enrichi ce champ pour construire les requêtes en introduisant manuellement des regroupements de mots en expressions, des variations de formes sur certains termes avec l'opérateur `'|'`, et dans de rares cas des variations liées à la synonymie. Par exemple, voici les champs `<title>` pour les *topics* numéro 553 et 564 :

```
spanish classical +guitar players
criticism limitation null hypothesis significance test
```

et les requêtes que nous avons utilisées :

```
spanish (classical | classic) guitar players
(criticism | limitation) "null hypothesis" "significance test"
```

Enfin pour établir des références pour la comparaison de notre modèle à base de proximité, nous avons aussi lancé les requêtes avec le modèle par défaut de *zettair*, le modèle de langue avec lissage de Dirichlet et avec le modèle Okapi avec les paramètres $k_1 = 1.2$, $k_3 = \infty$ et $b = 0.75$.

Les jugements de pertinence utilisé dans la campagne 2008 de INEX sont plus riches que ce qui nous est utile. En effet, pour un document et une requête les assesseurs devaient surligner dans le texte les passages pertinents et donner de plus le meilleur point d'entrée (*Best Entry Point*) pour la lecture d'un document qui contient au moins un passage pertinent. Toutes ces informations se retrouvent dans le fichier des jugements avec un format qui étend celui utilisé par `trec_eval`[2]. On y trouve successivement le numéro de *topic*, un champ ignoré, le numéro de document jugé, la somme de la longueur des passages jugés pertinents (et donc 0 si le document n'est pas pertinent), la longueur du document. Si la somme n'est pas nulle on trouve ensuite la position du *Best Entry Point* et la liste des passages avec des couples `<position>:<longueur>`. Toutes les positions et les longueurs sont exprimées en caractères. Voici deux lignes extraites de ce fichier, la première pour un document non pertinent et la deuxième pour un document contenant deux passages pertinents :

```
544 Q0 682628 0 2055
544 Q0 177316 572 4798 1915 1915:299 3711:273
```

Nous avons donc très simplement transformé ce fichier en ne gardant que les quatre premières colonnes, ce qui en fait un fichier utilisable par l'outil d'évaluation `trec_eval`.

Il faut noter que seules 70 des 285 *topics* ont été jugés.

5. Résultats

Nous avons donc quatre *runs* avec : le modèle Okapi BM 25, le modèle de langue avec lissage de Dirichlet, le modèle avec proximité sur les textes plats, le modèle avec proximité et propagation de l'influence des termes de titre. Le tableau 1 montre la sortie de l'outil `trec_eval` pour ces quatre *runs*. Les mesures calculées par cet outil sont classiques en recherche d'information et largement documentées par ailleurs. La sortie de `trec_eval` est complétée par des colonnes intitulées % qui indique le pourcentage de la différence par rapport à la méthode de Dirichlet. Pour beaucoup de ces mesures le classement du meilleur au moins bon est Dirichlet, puis proximité avec propagation, puis Okapi, puis proximité. Les performances du modèle de proximité sans propagation sont honorables et proches de celles du modèle Okapi. Les performances du modèle de proximité avec propagation sont très proches de celles du modèle de langue avec lissage de Dirichlet.

2. http://trec.nist.gov/trec_eval/

modèle :	**prox.**	%	**Okapi**	%	**prox. et struct.**	%	**Dirichlet**
Total number of documents over all queries							
Retrieved :	96023		94013		93190		94150
Relevant :	4850		4850		4850		4850
Rel_ret :	3743		3402		3785		3503
Interpolated Recall - Precision Averages :							
at 0.00	0.7433	−15.44	0.8341	−5.11	0.8786	−0.05	0.8790
at 0.10	0.5431	−14.58	0.6138	−3.46	0.6183	−2.75	0.6358
at 0.20	0.4731	−9.40	0.5152	−1.34	0.5003	−4.19	0.5222
at 0.30	0.3813	−8.71	0.3802	−8.98	0.3911	−6.37	0.4177
at 0.40	0.3058	−8.00	0.3059	−7.97	0.3033	−8.75	0.3324
at 0.50	0.2608	−5.30	0.2339	−15.07	0.2620	−4.87	0.2754
at 0.60	0.1979	−2.27	0.1723	−14.91	0.2139	5.63	0.2025
at 0.70	0.1447	12.43	0.1108	−13.91	0.1605	24.71	0.1287
at 0.80	0.0796	11.64	0.0671	−5.89	0.1057	48.25	0.0713
at 0.90	0.0442	20.77	0.0287	−21.58	0.0661	80.60	0.0366
at 1.00	0.0103	0.98	0.0030	−70.59	0.0205	100.98	0.0102
Average precision (non−interpolated) over all rel docs							
	0.2671	−7.93	0.2688	−7.34	0.2881	−0.69	0.2901
Precision :							
At 5 docs	0.5286	−11.06	0.5543	−6.73	0.5629	−5.28	0.5943
At 10 docs	0.4671	−8.16	0.4857	−4.50	0.4929	−3.09	0.5086
At 15 docs	0.4448	−3.30	0.4343	−5.59	0.4486	−2.48	0.4600
At 20 docs	0.4186	0.00	0.4050	−3.25	0.4164	−0.53	0.4186
At 30 docs	0.3781	1.67	0.3643	−2.04	0.3838	3.20	0.3719
At 100 docs	0.2284	2.61	0.2104	−5.48	0.2289	2.83	0.2226
At 200 docs	0.1501	5.85	0.1369	−3.46	0.1484	4.65	0.1418
At 500 docs	0.0843	9.91	0.0747	−2.61	0.0827	7.82	0.0767
At 1000 docs	0.0498	7.33	0.0448	−3.45	0.0493	6.25	0.0464
R−Precision (precision after R (= num_rel for a query) docs retrieved) :							
Exact :	0.3091	−6.19	0.3105	−5.77	0.3226	−2.09	0.3295

Tableau 1. *La sortie de* `trec_eval` *pour les quatre runs : proximité dans le texte plat, Okapi, proximité avec propagation des termes de titres, et modèle de langue avec lissage de Dirichlet.*

6. Travaux reliés et conclusion

La très grande majorité des travaux sur les modèles de recherche d'information qui donnent un score aux documents de façon à pouvoir les classer (*ranking*) portent sur des modèles vectoriels et probabilistes où les seules données sont des données statistiques de comptage. Ces données ignorent donc la position des occurrences des termes.

La plupart des travaux qui ont essayé de prendre en compte la proximité des termes l'ont fait en modifiant les classiques modèles vectoriels ou probabilistes. Beaucoup de

ces travaux ont ajouté des contributions relatives à la présence d'expressions dans la suite des travaux de Fagan [FAG 87]. Le travail conclusif sur cette méthode a été fait par [MIT 97] et nous reprenons ici leur conclusion : si on part d'un système de base médiocre (par exemple *lnc.ltc* en notation SMART), cette méthode améliore les résultats ; mais si on part d'un système correct (*pivoted cosine* dans leurs expériences [SIN 96]) il n'y a pas d'amélioration, voire dégradation.

D'autres auteurs [RAS 03, BUT 06] en relâchant la contrainte d'expressions exactes et en demandant à ce que les termes de l'expression apparaissent dans un fenêtre ont obtenus des améliorations par rapport à la méthode Okapi BM25. De bonnes améliorations ont été obtenues par Hearst en filtrant les documents classés par un classique modèle vectoriel et en ne gardant que ceux qui vérifient une contrainte booléenne qui doit être vérifiée par au moins un des passages du document [HEA 96]. Cette contrainte de passage est bien moins forte (de l'ordre de 100 à 300 mots) que celle sur les expressions, même avec relaxation.

À notre connaissance, seuls Clarke et al. [CLA 00] et Hawking et al. [HAW 95] ont proposé des méthodes de notation des documents vraiment nouvelles basées sur la proximité des termes de la requête. Clarke *et al.* proposent d'utiliser une algèbre qui recherche la famille des intervalles (*extents*) les plus petits qui vérifient telle ou telle containte. Ils utilisent cette algèbre pour chercher les intervalles les plus courts qui contiennent tous les termes de la requête. Ils donnent ensuite un score à chaque intervalle d'autant plus grand que l'intervalle est petit. Le score du document est la somme des scores des intervalles retrouvés. Ils montrent de bonnes performances pour des requêtes courtes. Hawking *et al.* ont aussi modélisé et implémenté pour la campagne 1995 de TREC des idées très proches de celles de Clarke *et al.* avec des intervalles et un score d'intervalle décroissant avec la longueur de l'intervalle.

Il y a plusieurs points communs entre notre méthode et celles de Clarke *et al.* et de Hawking *et al.* : une interprétation conjonctive des requêtes, une prise en compte de proximité, un score de document calculé par sommation de pertinences partielles.

Nous avons décrit un modèle de recherche d'information qui calcule le score d'un document en prenant en compte les positions des occurrences des termes de la requête. Avec une requête conjonctive, les scores calculés sont d'autant plus grand que les termes de la requête (qui doivent être tous présents dans le document) ont des occurrences proches dans le texte du document. Nous avons présenté une extension de ce modèle pour prendre en compte une structure logique hiérarchique qui propage l'influence des termes des titres. Les expériences que nous avons présentées montrent que ce modèle de calcul de score avec proximité permet d'obtenir des performances comparables à celles des meilleurs modèles de recherche d'information connus dans l'état de l'art. Il serait intéressant de comparer cette méthode sur les mêmes bases expérimentales avec les méthodes qui prennent en compte la position des occurrences des termes que nous avons citées dans l'état de l'art.

Remerciements

Ces travaux sont soutenus par le projet *Web Intelligence* du cluster *ISLE* financé par la région Rhône-Alpes.

7. Bibliographie

[BEI 08] BEIGBEDER M., « Compression de structure XML pour la recherche d'information structurée », *actes de CORIA'08, Cinquième Conférence en Recherche d'Information et Applications*, 2008.

[BUT 06] BUTTCHER S., CLARKE C. L. A., LUSHMAN B., « Term proximity scoring for ad-hoc retrieval on very large text collections », *SIGIR '06 : Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 2006, ACM, p. 621–622.

[CLA 00] CLARKE C. L. A., CORMACK G. V., TUDHOPE E. A., « Relevance ranking for one to three term queries », *Information Processing and Management*, vol. 36, 2000, p. 291-311.

[FAG 87] FAGAN J., « Experiments in automatic phrase indexing for document retrieval : A comparison of syntactic and non-syntactic methods », PhD thesis, Cornell University, 1987.

[HAW 95] HAWKING D., THISTLEWAITE P., « Proximity Operators - So Near And Yet So Far », Department of Commerce, National Institute of Standards and Technology, 1995.

[HEA 96] HEARST M. A., « Improving full-text precision on short queries using simple constraints », *Proceedings of the 5th Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*, 1996, p. 217–232.

[MIT 97] MITRA M., BUCKLEY C., SINGHAL A., CARDIE C., « An analysis of statistical and syntactic phrases », *Proceedings of RIAO-97, 5th International Conference "Recherche d'Information Assistee par Ordinateur"*, 1997, p. 200–214.

[RAS 03] RASOLOFO Y., SAVOY J., « Term Proximity Scoring for Keyword-based Retrieval Systems », *ECIR 2003 proceedings*, n° 2633 LNCS, Springer, 2003, p. 207–218.

[SIN 96] SINGHAL A., BUCKLEY C., MITRA M., « Pivoted Document Length Normalization », *SIGIR '96, Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 8 1996, p. 21-29.

REVISE, un outil d'évaluation précise des systèmes de questions-réponses

Sarra El Ayari [*], Brigitte Grau [*], Anne-Laure Ligozat [**]

LIMSI-CNRS [*]
Bât 508
F-91403 Orsay cedex (France)

Laboratoire IBISC - Université d'Evry [**]
Tour Evry 2, 523 place des terrasses de l'agora
91000 Evry

sarra.elayari, brigitte.grau@limsi.fr
aligozat@ibisc.univ-evry.fr

RÉSUMÉ. *Des campagnes d'évaluations sont organisées chaque année pour évaluer des systèmes de questions-réponses sur la validité des résultats fournis. Pour les équipes, il s'agit ensuite de réussir à mesurer la pertinence des stratégies développées ainsi que le fonctionnement des composants. À ces fins, nous décrivons un outil générique d'évaluation de type boîte transparente qui permet à un système produisant des résultats intermédiaires d'évaluer ses résultats. Nous illustrerons cette démarche en testant l'impact d'une nouvelle définition de la notion de focus.*

ABSTRACT. *Evaluation campaigns for question answering systems aim at evaluating their final results, i.e. the number of right answers. Then, in order to improve these systems, researchers try to evaluate each component, to improve them as well as to improve the global strategy. In order to help for these precise evaluations, we have conceived and developed a glass-box evaluation framework that works from the intermediary results provided by the different components. We will exemplify its capacities by showing how to measure a change in the determination of a question feature, the focus.*

MOTS-CLÉS : *Évaluation boîte transparente, évaluation boîte noire, systèmes de questions-réponses, plate-forme d'évaluation.*

KEYWORDS: *Glass-box evaluation, black-box evaluation, question answering systems, evaluation framework.*

1. Introduction

Évaluer un système modulaire est une tâche complexe qui suppose de prendre en compte à la fois les résultats finaux obtenus par le système ainsi que les résultats intermédiaires produits par chacun des modules. Les campagnes d'évaluation de systèmes de questions-réponses organisent chaque année de nouvelles tâches afin de permettre aux équipes de tester leurs systèmes sur des problèmes de plus en plus complexes : les corpus vont d'articles journalistiques bien écrits au web avec des formats et des styles très divers, et l'évaluation peut porter sur des passages ou sur des réponses de plus en plus précises, voire multiples pour des questions dont la réponse attend une liste d'éléments. L'évaluation s'effectue sur le nombre de bonnes réponses renvoyées par le système, donc sur les résultats finaux, ce que l'on appelle une évaluation boîte noire. Pour améliorer ces systèmes complexes, chacun des participants évalue en interne la pertinence de ses composants, c'est-à-dire mène une évaluation de type boîte transparente, de son système. Si cette pratique est courante, il n'y a pas d'outils génériques permettant d'évaluer ce qui se passe à l'intérieur des systèmes (Moldovan *et al.*, 2003). Le problème principal est que leur architecture dépend des stratégies utilisées : chaque système a ses stratégies propres et de ce fait sa propre architecture.

Nous proposons un outil d'évaluation de type boîte transparente qui permet à la fois d'évaluer les sorties produites par les composants, mais aussi de tester des stratégies sans toucher au système lui-même. Nous présenterons le système FRASQUES développé au LIMSI sur lequel nous travaillons (2), puis notre outil d'évaluation (3), que nous illustrerons par l'évaluation de la pertinence du terme pivot (le focus) utilisé pour extraire la réponse attendue (4).

2. Architecture d'un système de questions-réponses

Comment répondre de façon automatique à une question ? C'est le défi qu'essayent de relever les systèmes de questions-réponses. Contrairement aux moteurs de recherche, de tels systèmes permettent à un utilisateur de poser une question en langage naturel, et lui fournissent une réponse précise : *Quelle est la nationalité d'Ayrton Sena ?* attendra la réponse *espagnol*. Le système FRASQUES (Grau *et al.*, 2006) est composé de modules que sont l'analyse des questions, le moteur de recherche et l'extraction de réponses. Ces quatre modules fonctionnent de façon linéaire, les informations extraites lors de l'analyse de la question (la catégorie, le type général, les entités nommées, le focus, les termes et leurs variations sémantiques) sont ensuite utilisées par le moteur de recherche pour créer une requête qui fournira des documents contenant les mots de la question, ou bien leurs variations. Les informations extraites lors de l'analyse de la question *De quelle organisation Javier Solana était-il secrétaire général ?* seraient la **catégorie** (*quel*), le **type général** (*organisation*), le **nom propre** (*Javier Solana*), le **type d'entité attendu** (*organisation*), le **focus** (*secrétaire général*) et le **verbe principal** (*être*). Ces informations seront également nécessaires pour la sélection des phrases réponses candidates, qui se voient attribuer un poids en fonction de leur similarité avec les éléments extraits de la question. Enfin, lors de l'extraction de la

réponse précise, le focus, le verbe principal et l'entité nommée attendue jouent un rôle déterminant grâce à des patrons d'extraction de la réponse définis sur ces éléments. L'évaluation de la pertinence de chacun de ces composants n'est possible que si l'on a accès aux résultats intermédiaires qu'ils produisent, afin de juger de leur apport réel et des phénomènes qui posent problème.

3. REVISE, un outil pour visualiser et évaluer

3.1. *Etat de l'art*

Évaluer finement l'apport des composants d'un système suppose de mesurer la contribution de chacun des modules par rapport aux résultats finaux obtenus par le système. De ce point de vue, l'évaluation de type boîte transparente n'est pas en opposition avec une évaluation boîte noire, mais complémentaire pour obtenir un diagnostique complet (Sparck Jones, 2001). Ces deux méthodes dépendent avant tout de ce que l'on veut évaluer. Selon (Gillard *et al.,* 2006), l'état de l'art des évaluations réalisées sur des systèmes de questions-réponses montre le manque de lisibilité des apports des composants sur les résultats finaux et la nécessité d'une étude plus approfondie de chacun des composants.

Dans la littérature, on trouve deux courants liés à l'évaluation des différents modules d'un système. Le premier consiste à enlever un composant et à mesurer les résultats finaux obtenus. Il devient alors également possible de le remplacer par un autre, en mesurant à nouveau l'apport ou la perte obtenus (Costa *et al.,* 2006), (Tomas *et al.,* 2005). Le deuxième courant, assez novateur dans le domaine, est illustré par le système de questions-réponses JAVELIN (Nyberg *et al.,* 2003). JAVELIN est un système de questions-réponses qui intègre un module permettant de contrôler l'exécution du processus, ainsi que les informations qui sont utilisées. Il a été conçu de façon à permettre une évaluation de type boîte transparente. Il permet également de tester différentes stratégies qui peuvent ensuite être intégrées au système. Néanmoins, il n'existe aucun outil générique pour effectuer ce type d'évaluation.

Notre stratégie consiste à étudier et éventuellement à modifier les résultats intermédiaires créés par les composants et insérer ces nouveaux résultats dans le processus de traitement sans modifier le système lui-même. L'outil que nous avons conçu permet d'observer les données, de les modifier et d'évaluer l'impact des modifications sur le système.

3.2. *Description de notre interface*

REVISE est l'acronyme de *Recherche, Extraction, VISualisation et Evaluation*, termes qui résument ce que permet de faire notre outil. Recherche et extraction sont effectuées via la base de données qui contient les résultats intermédiaires produits par les composants du système. La visualisation des données se fait grâce à un export en

XML lié à des fichiers XSLT qui permettent la mise en relief de certains phénomènes. Enfin, l'évaluation est au cœur de cet outil grâce à la possibilité de modifier les résultats intermédiaires du système de questions-réponses et de les y ré-injecter. La figure 1 illustre les points d'évaluation transparente effectués sur le système FRASQUES, que nous avons décrit précédemment (Grau *et al.*, 2006). Des formulaires PHP sont éga-

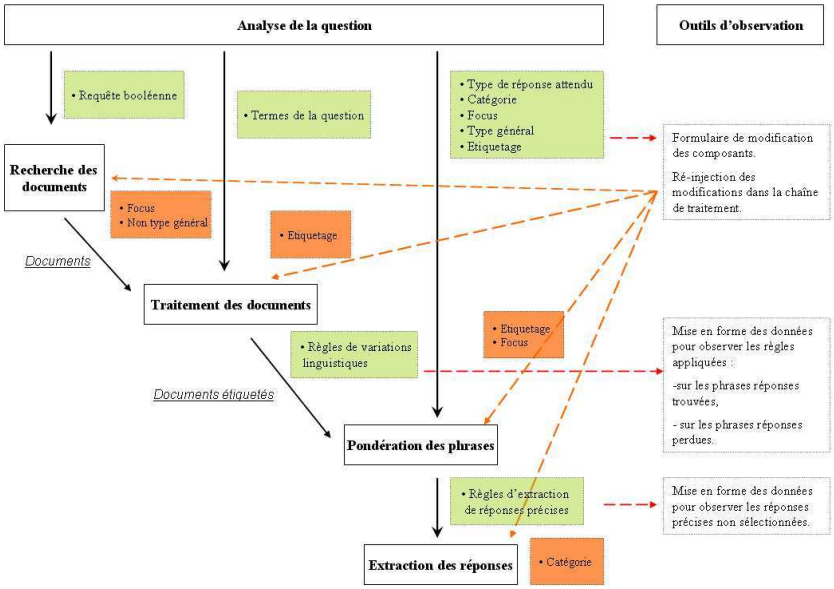


Figure 1. *REVISE, un outil d'évaluation transparente*

lement intégrés, qui permettent de modifier certains résultats dans la base de données. Les nouveaux résultats sont ensuite exportés et ré-injectés dans la chaîne de traitement avec le format adéquat.

3.2.1. *Base de données*

Les résultats intermédiaires produits par les différents composants sont décrits en XML dans un fichier contenant l'ensemble des résultats des processus effectués. La structure de ce fichier a conduit à la définition d'un schéma de base de données relationnelle. Les résultats sont stockés dans la base de données qui permet ainsi de représenter et rapprocher les différentes entités manipulées dans la chaîne, pour chaque test, c'est-à-dire l'ensemble des informations produites par une version donnée de la chaîne, pour un jeu de question et un corpus donnés : question vs passages réponses, caractéristiques de la question issues de l'analyse vs caractéristiques des phrases réponses (termes, poids, entités nommées), réponse trouvée ou non. Cette base contenant différents tests sous la forme de tables, il est possible de les comparer entre eux, de façon à prendre en compte différentes versions du système. L'intérêt de cette base de données est de donner la possibilité de faire des requêtes (prédéfinies ou libres)

sur ces résultats, et de ne sélectionner que ce qui intéresse l'utilisateur. Par exemple, on peut visualiser les informations liées aux questions de certaines catégories ou bien celles pour lesquelles on ne trouve pas la réponse, ou encore les questions pour lesquelles le focus est erroné. Le fait de pouvoir sélectionner les données en fonction de critères précis, comme ceux énoncés précédemment, permet une étude plus fine des phénomènes.

3.2.2. *Visualisation*

Un autre intérêt de notre outil est la possibilité d'interroger la base de données et de visualiser les résultats de façon lisible et organisée. En effet, grâce aux technologies XML et XSLT, les résultats sont extraits en XML et la visualisation est présentée en XHTML. En plus de l'interopérabilité de ces langages, il devient alors possible de jouer sur les traits à faire ressortir pour faire sens. Si la visualisation propre de données est importante, il devient également crucial de permettre l'émergence de formes grâce à des jeux de couleurs, pour mettre en évidence des phénomènes. Le fait de colorer le focus dans les phrases réponses candidates va permettre immédiatement de voir dans quels contextes il apparaît, de même pour les autres mots de la question. La figure 2 montre la mise en relief du focus et de la réponse attendue dans les différentes phrases réponses candidates extraites par le système FRASQUES.

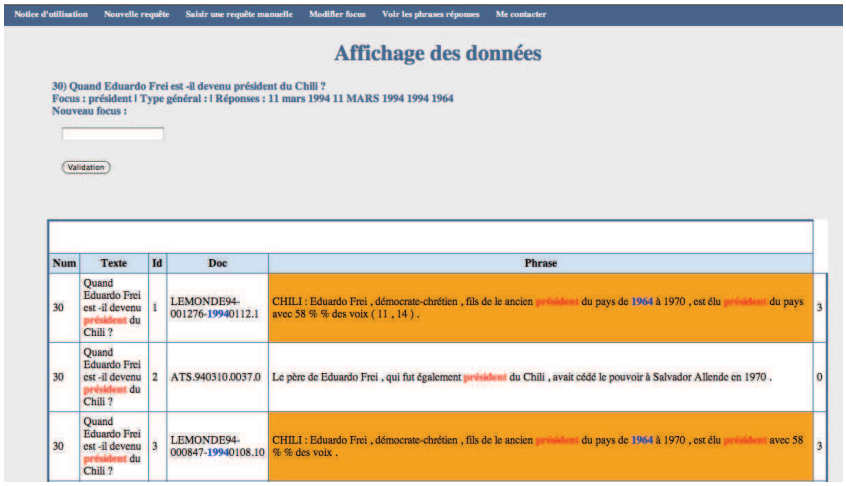


Figure 2. *Exemple de visualisation de données*

3.2.3. *Evaluation*

Enfin, en lien avec les deux points développés ci-dessus, REVISE permet d'évaluer les stratégies utilisées par un système de questions-réponses. Par exemple, il est possible de tester différentes définitions d'un critère, et de ce fait mesurer la pertinence des définitions les unes par rapport aux autres, sans modifier le système de questions-

réponses. Ayant accès aux résultats produits à différents niveaux de la chaîne de traitement, l'entrée et la sortie de chacun des modules peuvent être contrôlées, validées et le cas échéant modifiées afin de tester une hypothèse. Nous illustrerons ce point dans la suite de l'article.

3.3. *Généricité de l'outil*

REVISE est un outil qui travaille uniquement sur les résultats intermédiaires produits par un système. C'est-à-dire que les résultats obtenus par chacun des composants doivent être sauvegardés dans un même fichier. Si l'exécution de la chaîne de traitement est décrite de manière explicite, il est possible de modifier ces résultats et relancer des processus sans toucher à la chaîne de traitement. Ainsi, l'architecture de notre système de questions-réponses est décrite dans un fichier XML, qui est analysé afin de déclencher chaque processus. Ainsi, on peut aisément déterminer des points de reprises. Il devient alors possible de relancer le processus avec les résultats modifiés au point de reprise désiré. De ce fait, tout système qui produit des résultats intermédiaires peut utiliser REVISE. Afin d'illustrer ce que notre outil permet de faire, nous avons pris appui sur une évaluation de type boîte transparente effectuée sur le système de questions-réponses QRISTAL[1] dans l'article (Laurent *et al.*, 2006) : « QRISTAL est un système de questions-réponses multilingue (français, anglais, italien, portugais, polonais, tchèque) conçu pour extraire des réponses à partir de documents placés sur un disque dur, ou pour extraire des réponses à partir du web sur la base de pages ou passages retournés par des moteurs web classiques (Google, MSN, AOL, etc.) ». Cet article, écrit après participation à différentes campagnes d'évaluation (Equer[2], Clef05 et Clef06[3]) montre la nécessité pour les équipes de mesurer les performances des systèmes de façon précise, indépendamment des résultats fournis par les campagnes.

Les résultats produits par le système sont analysés, au niveau de l'analyse (syntaxique et sémantique) des questions (extraction des informations nécessaires au bon fonctionnement du système), de la sélection des blocs de réponses et de celle des phrases réponses. Il s'agit essentiellement d'observation des données, ainsi que de mesurer les taux de bonne mise en oeuvre des composants par le nombre final de bonnes réponses. Afin de tester l'impact de ces composants, chacun des modules du système a été déconnecté. Ceci a révélé l'importance de la catégorisation des questions, notamment lors de l'extraction des réponses. Les auteurs énoncent tout de même le problème de déconnection de composants tels que l'analyseur syntaxique, lequel est indispensable au bon fonctionnement de QRISTAL. REVISE permet de réaliser le même type d'évaluation de façon automatisée : les résultats produits sont stockés dans la base de données, ce qui permet d'effectuer les mêmes études :

1. QRISTAL est l'acronyme de *Questions-Réponses Intégrant un Système de Traitement Automatique des Langues*.
2. Voir http ://www.technolangue.net/article.php3 ?id_article=195.
3. Voir http ://www.clef-campaign.org/.

1) Analyse syntaxique de la question / étiquetage
Notre outil stocke les informations liées à la question dans une base de données. Une requête sur la base permet ensuite de visualiser ce qui a été extrait. Si l'extraction n'est pas satisfaisante, il est tout à fait possible de modifier les champs erronés et mesurer l'impact à différentes étapes du traitement.

2) Analyse sémantique de la question
De la même façon, les synonymes étant des résultats produits par le système, nous pouvons observer leur pertinence et leur bonne application en les visualisant dans les phrases sélectionnées correctes et incorrectes. Nous pouvons aussi visualiser les mots de la question qui ne figurent pas dans les passages réponses. Ainsi, on peut étudier à la fois la pertinence des synonymes, mais aussi le défaut de couverture des lexiques.

3) Analyse des passages extraits
L'ordonnancement des passages est également accessible, avec les scores attribués.

4) Analyse des phrases réponses
Même chose pour les phrases réponses candidates, sur lesquelles on peut également observer finement l'application ou non des patrons d'extraction de la réponse précise.

Notre outil offre, en terme de visualisation, un accès à tous les éléments jugés importants au sein du système QRISTAL, avec une normalisation de la visualisation et la possibilité de modifier ces résultats et de relancer le traitement. L'étude d'une question en particulier est tout à fait possible en indiquant son numéro ainsi que le numéro de processus qui nous intéresse.

4. Etude du focus

Dans cette partie, nous allons montrer comment REVISE peut servir à étudier un paramètre particulier d'un système, le focus. Après avoir redéfini le terme focus, nous utiliserons notre outil pour valider la nouvelle définition.

4.1. *Redéfinition du terme focus*

Le terme focus est un élément de la question, qui est le terme pivot pour extraire la réponse attendue, terme censé apparaître à proximité de la réponse (Ferret *et al.*, 2002). Il s'agit de l'élément central de la question, auquel se rattache directement la réponse. Sa définition est essentielle à l'extraction de la réponse dans notre système, car il permet d'expliciter des patrons d'extraction modélisant l'expression en langue de la relation existant entre le focus et la réponse. Le critère principal de la reconnaissance de ce focus était qu'il s'agit du sujet du verbe principal de la question.

Nous avons mené une expérience sur le focus ainsi reconnu (El Ayari, 2007), afin de tester si cette hypothèse était vérifiée par le système QALC, version anglaise du système FRASQUES. Il en ressort que des questions n'ont pas de focus, car on ne s'appuyait que sur les entités nominales, et que le choix systématique d'un groupe

nominal induit le fait que l'écriture de patrons d'extraction précis selon le type de relation qui doit être vérifié entre focus et réponse est plus difficile, car le groupe nominal choisi peut varier pour un même type de relation. Nous avons donc précisé la notion de focus et nous allons montrer comment cette nouvelle définition peut être évaluée sans avoir à modifier l'analyse des questions pour mettre en oeuvre sa reconnaissance automatique. Alors que le focus a toujours été défini comme une entité, nous proposons l'idée qu'il peut également s'agir d'un événement. Si l'on représente la question sous la forme d'un graphe, c'est le focus qui en sera le nœud central. La représentation de la phrase *Quand le pont de Normandie a-t-il été inauguré ?* serait celle de la figure 3. Les données que nous présentons sont tirées de la campagne d'évaluation CLEF07[4].

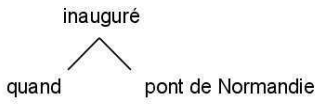


Figure 3. *Graphe de la question*

4.1.1. *Le focus est un événement*

Sa reconnaissance repose sur le sens du verbe, qui doit être assez fort pour traduire un événement. Si le verbe possède un complément, le focus sera composé.

– *Question :* Quand [débuta le procès][5] de Paul Touvier ?

– *Réponse :* Le procès de Paul Touvier s'ouvrira le **17 mars** devant la cour d'assises des Yvelines (région parisienne), a-t-on appris officiellement jeudi.

Cette question demande une précision sur l'événement *débuter le procès*. C'est cet événement qui sera le focus de la question, et qui permettra l'extraction de la réponse précise attendue par la question. La réponse est complément du verbe s'ouvrir. La notion de début est essentielle, pour sélectionner entre des phrases qui parleraient du procès, mais donneraient d'autres dates : fin, interruption, etc..

4.1.2. *Le focus est une entité*

Si le verbe n'a qu'un sens relatif, c'est alors l'entité sur laquelle la question est posée qui sera le focus. Cette entité pourra être exprimée *en intension* ou *en extension*.

– De quelle organisation Javier Solana était-il [secrétaire général] ?

– Javier Solana a officiellement été nommé mardi secrétaire général de l'**OTAN**, mettant fin à une vacance de plusieurs semaines.

Cet exemple illustre une entité exprimée en extension, c'est-à-dire nommée : *Javier Solana*, et l'autre en intension, c'est-à-dire qu'on fait référence à quelqu'un par une

4. Voir http ://www.clef-campaign.org/.
5. Le focus est indiqué entre crochets.

description : *secrétaire général*. Dans ce cas, le focus sera l'entité exprimée en intension (secrétaire général), car c'est sur la fonction que la question est posée. On voit effectivement que la réponse est liée à ce terme focus dans la phrase réponse.

– *Question :* A quel parti appartient [Thérèse Aillaud] ?

– *Réponse :* Placée dans la même situation que M. Siffre, le député et maire sortant de Tarascon, Thérèse Aillaud (**RPR**), n'enlève que le tiers des suffrages exprimés alors qu'elle avait été réélue en 1989 dès le premier tour de scrutin avec près de 63% des voix.

Ici, il n'y a ni verbe porteur d'un événement, ni entité exprimée en intension. Le focus sera *Thérèse Aillaud*, entité sur laquelle la question est posée.

4.1.3. *Quelques particularités en fonction des catégories de questions*

Dans le domaine des systèmes de questions-réponses, les questions sont souvent classées en fonction de leur pronom interrogatif et du type d'entité attendu (date, lieu, personne ou autre). Nous avons défini des catégories en fonction du type de focus exprimé et de la relation recherchée. Les principales catégories sont :

– Définition : *Qu'est-ce que l'accélération centrifuge ?*

– Combien : *Combien y a-t-il eu de mariages en Grande-Bretagne en 1993 ?*

– Quand : *Quand a eu lieu la chute du régime communiste en Afghanistan ?*

– Où : *Où se situent les îles Marquises ?*

– Quel : *Quelle était la nationalité d'Ayrton Senna ?*

– Qui : *Qui est Michael Jackson ?*

– Instance : *Citer le nom d'un corps céleste.*

Pour les questions de type Combien, nous avons deux valeurs à extraire de la question que sont l'unité et le terme focus.

– *Question :* Combien de **puits** ont dû être [fermés] suite à la rupture d'un oléoduc en Sibérie ?

– *Réponse :* La rupture d'un oléoduc dans le gisement de Samotlor, dans l'ouest de la Sibérie, a contraint les autorités à fermer **52** puits pour empêcher toute extension de la pollution, a rapporté lundi la télévision russe.

On voit que l'unité constitue un indice fort pour extraire la réponse attendue, de même que l'événement dont il est question. La réponse attendue est encadrée par ces deux éléments dans la phrase réponse, ce qui tend à justifier les deux stratégies. Dans ce cas-là, des patrons d'extraction sont constitués autour de l'unité et autour du focus.

En ce qui concerne les questions de type Instance, il s'agit de questions qui donnent la définition (ou description) de la réponse : elles donnent le type de la réponse. Il n'y a donc pas de focus, et une stratégie différente pour extraire la réponse, axée sur la vérification de ce type, est mise en place. Par exemple : *Quelle est la plus grande banque du Japon ?* ou *Que s'est-il produit en Algérie dans la nuit du 17 au 18 août*

1994 ? Nous n'indiquons donc pas de terme focus pour ces questions, sa définition ne pouvant s'appliquer ici. Nous allons illustrer le fonctionnement de REVISE sur la notion de focus qui vient d'être définie.

4.2. *Validation de l'hypothèse*

Nous rappelons que le terme focus sert essentiellement à extraire les réponses précises. La réponse devant être liée à ce terme, ces deux termes sont souvent proches dans la réponse et cela peut être mesuré par la distance en mots entre les deux entités. Pour tester la pertinence de cette nouvelle définition, nous avons comparé l'ancienne définition du focus à la nouvelle, en terme de proximité avec la réponse. Nous allons détailler les différentes étapes de notre méthodologie :

1) le choix manuel[6] du focus des questions en fonction de la définition
Pour ce faire, nous avons effectué une requête sur la base de données, en spécifiant le corpus (Clef07). Un formulaire nous a permis de modifier le focus extrait par FRASQUES.

2) la sélection des phrases réponses qui contiennent ce focus et une réponse possible
Une requête sur la table contenant les réponses a été créée, en filtrant sur la présence du terme focus et d'une réponse possible (les réponses sont également stockées dans une table). La visualisation est permise par des scripts XSLT qui permettent de choisir le format d'affichage des résultats ainsi que les codes couleur à utiliser.

3) le calcul de la distance (en mots) entre focus et réponse
Un nouveau script a été ajouté à l'interface pour calculer automatiquement ces distances. Les résultats sont stockés dans la base de données de façon à rester accessibles.

Cette méthodologie a permis de comparer les définitions, afin de mesurer laquelle est plus pertinente que l'autre pour extraire la réponse précise. Le tableau 1 présente les résultats du calcul automatique de la distance moyenne en mots entre un focus et une réponse, pour chacune des définitions. La nouvelle définition du focus apparaît

Ancien focus	Nouveau focus
8 mots	4 mots

Tableau 1. *Résultats de la distance focus/réponse*

plus pertinente que l'ancienne, avec une distance moyenne de quatre mots, soit moitié moins que l'ancienne. **345 phrases réponses** sur les données de Clef07 contiennent une réponse et le focus de la question à laquelle cette phrase répond. Nous avons ainsi classé les résultats par catégories de questions. Le tableau 2 présente le nombre de phrases réponses par distance. Nous nous sommes arrêtés à une distance maximale de

6. Cette étape est réalisée manuellement pour tester la pertinence de la nouvelle définition du focus, sans avoir à modifier l'analyse des questions.

quatre mots, une distance syntaxique supérieure rend inutilisable les patrons d'extraction. Les distances 0 et 1 comptabilisent un peu plus de la moitié des phrases réponses

Distance (mots)	Nb phrases	Catégories les plus fréquentes
0	129	Combien, Définition
1	48	Quel, Quand
2	39	Quel
3	22	Quel
4	24	Quel

Tableau 2. *Distances les plus fréquentes*

(177). Si l'on rajoute les distances 2, 3 et 4 on obtient un score de 75% des phrases réponses, ce qui est très encourageant pour l'extraction des réponses. REVISE a permis ici de vérifier notre nouvelle définition du terme focus, grâce à l'observation et la modification des résultats de l'analyse des questions, la sélection des phrases réponses extraites de la base de données qui nous intéressaient ainsi que le calcul des distances de mots entre focus et réponses. Nous avons modifié manuellement les focus extraits par le système, données préalablement insérées dans la base, à l'aide d'un formulaire PHP. Nous avons ensuite sélectionné les phrases réponses grâce à une requête pré-enregistrée, et calculé automatiquement la distance en mots entre le focus et la réponse. Notre base de données a également permis d'effectuer des calculs, comme la moyenne des distances, de façon à pouvoir faire notre évaluation. La sélection du focus, qu'il s'agisse de la première ou de la deuxième, a été faite en fonction des mots de la question uniquement. C'est-à-dire qu'aucune variation syntaxique ni sémantique n'a été prise en compte.

Étant donné que notre nouvelle définition du focus repose essentiellement sur la notion d'événement, c'est le verbe de la question qui est sélectionné. Il est fréquent que l'événement exprimé par le verbe dans la question soit sous une forme nominalisée dans la réponse. Les synonymes sont autant de termes que nous n'avons pas comptabilisés ici. La prise en compte de ces variations lors de l'extraction des réponses devrait augmenter les résultats obtenus en terme de présence du focus. La prochaine étape consistera à établir des critères de reconnaissance automatique de la nouvelle définition du focus, entre entité et événement. Nous utiliserons notre outil pour mesurer l'impact de son intégration au système au niveau de l'extraction des réponses précises.

5. Conclusion

L'intérêt d'une évaluation de type boîte transparente n'est plus à prouver. Mais il est difficile de créer une méthodologie pour évaluer la contribution de chacun des différents composants d'un système. Notre outil, REVISE, facilite l'observation des résultats produits par le système. Il permet également de réaliser des évaluations de type boîte transparente. De plus, sans modifier le système, il devient possible de tester

ses stratégies de recherche, qu'il s'agisse de l'analyse des questions ou encore de l'extraction des réponses, en ne modifiant que les résultats produits, et en relançant le système à un endroit particulier de la chaîne de traitement. De la sorte, n'importe quel système produisant des résultats intermédiaires devrait être en capacité d'utiliser REVISE.

6. Bibliographie

Costa L., Sarmento L., « Component Evaluation in a Question Answering System », *Actes de la 5e conférence Language Resources and Evaluation Conference (LREC)*, Gênes, Italie, 24-26 mai, 2006.

El Ayari S., « Evaluation transparente de systèmes de questions-réponses : application au focus », *Actes de ReciTAL*, 2007.

Ferret O., Grau B., Hurault-Plantet M., Illouz G., Monceaux L., Robba I., Vilnat A., « Recherche de la réponse fondée sur la reconnaissance du focus de la question », *Actes de Traitement automatique du langage naturel (TALN)*, 2002.

Gillard L., Bellot P., El-Beze M., « Question Answering Evaluation Survey », *Actes de la 5e conférence Language Resources and Evaluation Conference (LREC)*, Gênes, Italie, 24-26 mai, 2006.

Grau B., « Evaluation des systèmes de question-réponse », *Évaluation des systèmes de traitement de l'information*, Hermès, chapter 3, p. 77-98, 2004.

Grau B., Ligozat A.-L., Robba I., Vilnat A., Monceaux L., « FRASQUES : A Question-Answering System in the EQueR Evaluation Campaign », *Actes de la 5e conférence Language Resources and Evaluation Conference (LREC)*, Gênes, Italie, 24-26 mai, 2006.

Laurent D., Nègre S., Séguéla P., « QRISTAL, le QR à l'épreuve du public », *Actes de Traitement automatique du langage naturel (TALN)*, 2006.

Moldovan D., Paca M., Harabagiu S., Surdeanu M., « Performance issues and error analysis in an open-domain question answering system », *Actes de ACM Transactions on Informations Systems*, 2003.

Nyberg E., Mitamura T., Callan J., Carbonell J., Frederking R., Collins-Thompson K., Hiyakumoto L., Huang Y., Huttenhower C., Judy S., Ko J., Kupse A., Lita L. V., Pedro V., Svoboda D., Durme B. V., « The JAVELIN Question-Answering System at TREC 2003 : A Multi-Strategy Approach with Dynamic Planning », *Actes de Text Retrieval Conference (TREC)*, 2003.

Sparck Jones K., « Automatic language and information processing : rethinking evaluation », *Natural Language Engineering*, chapter 7, p. 1-18, 2001.

Tomas D., Vicedo J. L., Saiz M., Izquierdo R., « Building an XML framework for Question Answering », *Actes de Cross Language Evaluation Forum (CLEF)*, 2005.

Identification de Phénomènes dans l'analyse d'interactions humaines

Les traces d'interactions humaines, un nouveau domaine d'application pour la RI

Gregory Dyke[*], Michel Beigbeder[*], Kristine Lund[**], Jean-Jacques Girardot[*]

[*] *Ecole Nationale Supérieure des Mines de Saint-Etienne,*
158 Cours Fauriel, 42023 Saint-Etienne
[**] *UMR ICAR, CNRS, ENS-LSH, Université de Lyon,*
15 parvis René Descartes, F-69700 Lyon, France
{Gregory.Dyke, Michel.Beigbeder, Jean-Jacques.Girardot}@emse.fr
Kristine.Lund@univ-lyon2.fr

RÉSUMÉ. *L'étude socio-cognitive des interactions humaines médiatisées par ordinateur passe par l'analyse de corpus complexes, de plus en plus vastes, regroupant les enregistrements audio-video et les traces informatiques de l'interaction médiatisée. Dans cet article, nous présentons et modélisons l'interrogation de tels corpus au moyen de méthodes de RI. Nous montrons que, moyennant ces modèles, certaines questions de recherche pour l'analyse d'interactions peuvent se ramener à des problèmes connus de RI. Nous exposons enfin les résultats de nos premières implémentations d'algorithmes de RI pour l'interrogation de traces d'interaction.*

ABSTRACT. *The socio-cognitve study of human computer-mediated interactions can be done through the analysis of increasingly larger and complex corpora composed of audio-video recording and interaction logfiles. In this article, we present and model the querying of such corpora with IR methods. We show that these models afford the transformation of certain interaction analysis research questions into known IR problems. We describe the results of our first implementations of RI algorithms for querying interaction corpora.*

MOTS-CLÉS : *Analyse de traces, Traces d'interaction, Système dédié*

KEYWORDS: *Interactive IR and visualization, Non-probabilistic retrieval models, Domain-specific search engines, Interaction traces*

1. Introduction

Les études socio-cognitives des interactions humaines supportées par ordinateurs s'effectuent aujourd'hui le plus souvent au travers des traces de ces activités (*logs* des outils, enregistrements numériques audio ou vidéo). L'analyse de ces traces est complexe, et les données sont en trop grande quantité pour être facilement appréhendées (Dyke *et al.*, 2007), même si depuis peu quelques techniques issues de l'état de l'art informatique (p.ex. apprentissage automatique pour assister le chercheur dans le codage de ses données (Erkens *et al.*, 2008)) ont été mises en oeuvre pour faciliter la tâche d'analyse. L'utilisation de l'informatique reste cependant limitée malgré les résultats prometteurs de chercheurs utilisant cet outil pour diverses tâches (Cox, 2007).

Le temps passé par un chercheur à analyser une séquence de données est de l'ordre de 10 et 1000 fois la longueur de cette séquence [1]. Il est dès lors d'une importance capitale pour lui de pouvoir repérer rapidement dans un grand *corpus* [2] les *phénomènes*[3] qui l'intéressent. Nous nous sommes donc proposés d'adapter et d'appliquer certaines approches de la RI à l'analyse de corpus, pour permettre au chercheur de décrire, sous forme de requêtes, les aspects des phénomènes qu'il recherche, et retrouver ainsi, de manière automatique, ces phénomènes potentiels.

Dans cet article, nous passons en revue l'activité d'analyse de ces chercheurs afin de souligner les apports potentiels de la RI, et présenter les modèles de représentation de corpus et d'interrogation qui permettent de tirer parti des techniques évoquées. Nous décrivons enfin *Tatiana*, un outil destiné à aider l'analyse de traces, qui implémente certaines des fonctionnalités décrites, et montrons quelques résultats obtenus sur des corpus réels.

2. L'analyse d'interactions humaines médiatisées par ordinateur

Les raisons d'être des analyses de corpus d'interactions médiatisées par ordinateur sont nombreuses. Le but peut être la mise au point d'un outil de CSCL/CSCW (Computer Supported Cooperative Learning/Work), l'analyse de son ergonomie ou l'évaluation de ses procédures d'usage. Dans le cadre du CSCL, de très nombreux logiciels sont utilisés, des plus généraux (gestionnaires de chats, de forums) aux plus spécialisés (micro-mondes, outils de conception collaboratifs). L'évaluation d'un apprentissage effectué par des étudiants au cours d'une session s'effectue souvent au travers du calcul d'indicateurs. L'enseignant s'intéressera par exemple aux connaissances manipulées, le didacticien au transfert de ces connaissances, le cogniticien à

1. Chiffres tirés de discussions avec des chercheurs en science cognitives, en particulier lors des séminaires internes de l'UMR ICAR.
2. Nous utilisons tout au long de cet article le terme *corpus* dans l'acceptation très particulière que lui prêtent les socio-cogniticiens, pour désigner l'ensemble des documents relatifs à une observation. La section 3 en présente une description précise.
3. Au sens de la psychologie cognitive, c'est à dire un changement dans l'enchaînement habituel d'un processus social ou cognitif.

l'argumentation ou à la reformulation qui sont à l'origine de ce transfert, etc. Dans ces exemples, les captures audio et vidéo des expérimentations sont indispensables à la compréhension des interactions (Goodman *et al.,* 2006, Avouris *et al.,* 2007). Les outils de CSCW sont aussi de plus en plus utilisés dans l'industrie, entre autres comme support de réunions à distance. Les traces de ces outils constituent ainsi une mémoire de l'entreprise, dont l'importance est croissante, car de plus en plus de décisions administratives ou techniques sont prises dans de telles circonstances (Lund *et al.,* 2007).

Nous prenons essentiellement dans cet article des situations dans le domaine de l'analyse socio-cognitive des traces d'interactions, puisque la plupart des cas sur lesquels nous avons travaillé rentrent dans cette spécialité. Illustrons ce domaine au travers d'un premier exemple typique (*cas 1*) : un chercheur en cognitique s'intéresse à la manière dont la *reformulation*[4] participe au transfert des connaissances entre enseignant et étudiants dans le cadre d'un suivi de projets d'un cours d'informatique. Dans le cas étudié, trois rencontres, d'une durée d'une heure chacune, se déroulent à une semaine d'intervalle environ. Deux enseignants et dix binômes d'étudiants sont impliqués dans l'expérimentation, soit 30 heures de session et 22 participants différents. Les rencontres ont lieu en présentiel, les intervenants utilisant soit l'oral, soit un outil de communication offrant un forum, un éditeur de texte partagé, un tableau blanc, un dispositif de construction de graphes argumentatifs, etc.

A partir des données brutes (traces des outils et enregistrements audio et vidéo), le chercheur va préparer le corpus pour obtenir des *artefacts*[5] primaires présentant la granularité désirée. Ce travail implique de synchroniser les différents médias, de convertir les traces dans une représentation appropriée à l'outil d'analyse, et de regrouper ou fragmenter les événements primaires pour les amener à la granularité souhaitée. L'audio et la vidéo sont segmentés en tours de paroles, qui sont transcrits manuellement. Les événements des traces d'interactions sont regroupés, pour obtenir des interventions complètes dans le chat, des séquences signifiantes dans l'éditeur de texte, etc.

Dans l'approche traditionnelle du corpus, l'examen de celui-ci est entièrement « manuel » : le chercheur visionne de manière répétitive les séquences du corpus, pour tenter de découvrir les phénomènes qu'il recherche afin d'appuyer son analyse. Ainsi, le phénomène de « reformulation » va être caractérisé par le fait qu'un enseignant utilise oralement un terme ou un groupe de termes, que l'on retrouve peu après sous la forme d'une prise de notes, par un étudiant, dans le chat ou l'éditeur partagé. Muni de ces exemples de reformulation, le chercheur va ensuite tenter de montrer que les étudiants se sont appropriés les connaissances ainsi reformulées.

Dans une autre étude (*cas 2*), un chercheur nous avait soumis un ensemble de

4. Le terme de *reformulation* est utilisé ici non pas dans sa signification psychanalytique traditionnelle, mais pour désigner un phénomène dans lequel un locuteur B reprend, sous une forme voisine, sur le même support ou un support différent (oral-oral, oral-écrit, etc.), un concept qui vient d'être énoncé par un locuteur A.
5. Nous utilisons ici ce terme dans son acceptation anglo-saxonne, pour indiquer un objet artificiel créé par le chercheur, et destiné à l'aider dans son analyse.

requêtes, correspondant aux phénomènes qu'il recherchait dans plusieurs corpus de transcriptions orales, requêtes dont voici quelques exemples :

– chevauchement (deux locuteurs parlent simultanément) suivi de « attends » dans les PV (productions verbales) de moins de dix tokens,

– cascade de trois « attends » dans un intervalle de cinq PV maximum,

– fréquence de « oui » par sexe dans les enregistrements avec des locuteurs mixtes,

– PV avec chevauchement et « attends » précédées d'une PV avec « euh », etc.

Un autre exemple (*cas 3*) a été l'analyse d'une situation réelle du monde industriel, mettant en scène des concepteurs de métiers différents effectuant une réunion dont le but était de ratifier les décisions prises au sujet de 12 points de conception. Lors de l'analyse, les chercheurs désiraient identifier des phénomènes communs entre la façon dont est abordée chacun de ces points. A ces fins, ils ont transcrit le corpus et l'ont codé en fonction du type des énoncés (affirmations, questions, etc.) et de leur contenu (orienté-solution, orienté-technologie, etc.)

On voit à travers ces exemples que la détection des phénomènes intéressant le chercheur relève de la recherche de passages dans des documents structurés, avec utilisation d'un aspect de proximité temporelle. De manière générale, nous attendons de l'utilisation de la RI dans la détection de phénomènes qu'elle réduise le nombre de cycles nécessaires à cette détection, qu'elle en diminue le silence, qu'elle raccourcisse le temps consacré par le chercheur à cette recherche, et rende possible l'application d'une question de recherche sur des corpus de très grande taille.

3. Modèle de la représentation

Le terme de *corpus* désigne ici l'ensemble des documents que le chercheur veut étudier comme un tout. Dans l'exemple évoqué, il s'agit des documents recueillis au cours d'une expérience, qui consistent typiquement en :

– documents audio ou vidéo enregistrés pendant l'expérience, transcriptions de ces enregistrements effectuées par le chercheur (on peut considérer aussi que ces transcription constituent un artefact secondaire) ;

– traces des interactions médiatisées ;

– notes prises avant, pendant, et après l'expérience par les participants ou les observateurs ; documents distribués aux participants de l'expérience, et, de manière générale, tout autre document jugé pertinent par le chercheur.

Ces documents constituent un ensemble fini, le « corpus primaire », qui n'a pas de raison d'évoluer *a posteriori*, représentant la totalité des données primaires recueillies pendant et sur l'expérience. Cet ensemble de documents est assez disparate. Les outils de CSCL/CSCW sont nombreux, ont souvent des finalités différentes, et utilisent bien sûr des représentations différentes pour leurs traces. Certaines expérimentations peuvent en utiliser plusieurs simultanément. Nous ne pouvons envisager un outil gé-

nérique, capable d'analyser tous les formats disponibles, et ceux à venir, pas plus qu'il n'est raisonnable d'espérer proposer un jour un format « universel » pour ces données, susceptible de représenter toutes les informations disponibles dans ces différentes traces.

Nous prenons comme hypothèse que les données primaires sont immuables, mais que le chercheur va travailler sur des « représentants » de ces données primaires. L'approche choisie a été de créer un *format pivot*, pouvant représenter toutes les informations auxquelles le chercheur peut s'intéresser durant son analyse. Ce format permet au chercheur de choisir les aspects des données recueillies qu'il souhaite analyser, en laissant les autres éléments de côté. Ce choix n'est en rien irréversible : à tout moment, il est possible d'ajouter de nouveaux aspects, extraits du corpus primaire, sans perdre aucun résultat du travail effectué. Il est de même possible à deux chercheurs de travailler sur un même corpus, en s'intéressant à des données qui ne se recouvrent pas totalement, puis ultérieurement de croiser leurs analyses, pour faire apparaître de nouveaux phénomènes.

La représentation choisie pour le format pivot est une représentation XML qui utilise une structure de donnée, que nous avons baptisée *item*, et qui n'est pas sans évoquer les *frames* (Minsky, 1974). Un item est une représentation d'un objet du monde réel modélisé sous la forme d'un ensemble de facettes, chaque facette étant un couple nom-valeur décrivant un aspect de l'objet. Chaque item créé comme représentant d'un objet du corpus comporte une facette qui décrit sa provenance, facette dite *ancre*. Pour un document XML présent dans le corpus, par exemple, l'ancre est constituée de l'identification du document et du chemin d'accès à l'élément d'où l'on a extrait l'information.

Le type d'item le plus important utilisé dans l'analyse est l'*événement*. Un tel item comporte une *datation* de l'événement, sous la forme d'un couple date-durée. Les autres facettes de l'item documentent d'autres aspects de l'événement auquel s'intéresse le chercheur, tels que l'outil utilisé, l'identification du participant, la description de l'action, etc. Un message envoyé dans un chat pourra typiquement être représenté par un événement contenant des facettes "outil", "participant" et "message", en plus des facettes "ancre" et "datation". La trace primaire de l'outil peut contenir nombre d'autres informations (le numéro du message, du groupe, l'adresse IP de la machine, etc.), que le chercheur peut juger, ou non, pertinentes, et associer ou non à l'événement. La trace de l'utilisation d'un outil est ainsi traduite par une successions d'événements, qui est représentée, dans notre formalisme, par une séquence d'items. Une telle séquence chronologique d'événements associée à un outil est dite *rejouable*.

La transformation de la trace d'un outil quelconque de CSCL/CSCW en une séquence d'items s'effectue en général par un script *ad hoc* écrit en XQuery (W3C 2008) pour des traces XML, ou en Java pour les autres représentations. Cette approche permet au chercheur de ne conserver pour son analyse que les aspects des données disponibles qui l'intéressent, tout en affranchissant l'outil d'analyse de la prise en compte de formalismes différents. Pour un logiciel particulier, une dizaine de scripts de quelques lignes suffisent en général à couvrir tous les besoins du chercheur.

4. Modèle de l'analyse

Les événements recueillis au cours de l'expérimentation ont rarement la granularité adéquate à l'analyse. Le chercheur va éliminer des événements qu'il juge non significatifs, ou regrouper plusieurs événements successifs, afin d'obtenir une « action » de plus haut niveau sémantique. Ainsi, il peut choisir de supprimer une succession de déplacements d'un objet dans un tableau blanc, grouper une série d'insertion de caractères dans un éditeur partagé pour obtenir un événement qu'il qualifiera d'insertion de mot ou de phrase, etc. Il va aussi catégoriser ces événements, leur associer des commentaires, ou les lier à d'autres objets du corpus. Toutes ces opérations se traduisent par la création de nouveaux artefacts, qui concrétisent des vues spécifiques sur le corpus, ou certains aspects de la réflexion du chercheur.

Dans cette étape du travail, les affichages de données prennent toute leur importance. Le chercheur peut utiliser, en succession ou simultanément, des affichages sous forme de tables (à la manière d'Excel), de graphes, de représentations linéaires temporelles, etc. Il peut décider d'effectuer de nouvelles transformations sur ces données, leur ajouter de nouvelles facettes extraites du corpus primaire ou d'artefacts secondaires, les enrichir ou les filtrer de diverses manières. Un atout considérable pour le chercheur consiste aussi à pouvoir rejouer, avec l'outil ayant servi à produire les traces qu'il analyse, les traces elles-mêmes, et voir se dérouler sous ses yeux, en temps réel, ce que pouvaient voir sur leurs écrans les participants à l'expérience. Pour cette raison, le chercheur privilégie les outils permettant de rejouer les sessions enregistrées, tels que DREW (Corbel *et al.,* 2002), Digalo (Lotan-Kochan, 2006), CoFFEE (De Chiara *et al.,* 2007) et quelques autres.

Le travail du chercheur est donc caractérisé par une succession de cycles, au cours desquels il analyse (manuellement) ses données ou leurs représentations, crée de nouveaux objets de travail, puis en effectue à nouveau l'étude. La phase durant laquelle le chercheur examine ses rejouables pour trouver des évidences des phénomènes qu'il cherche à mettre en lumière, est, on le voit, la partie critique, en temps et en effort humain, du processus d'analyse.

5. Quels outils de la RI pour l'analyse de traces ?

L'identification des phénomènes recherchés par le chercheur, et la création interactive de nouveaux rejouables qui représentent ces phénomènes, est une des opérations les plus gourmandes en temps. C'est lors de cette phase d'examen des documents que diverses approches proposées par la RI peuvent s'avérer pertinentes.

La recherche d'information, au sens de recherche de documents, est représentée du point de vue du système comme une activité en deux phases : la phase d'indexation et la phase d'interrogation. La phase d'indexation explicite les entités qui seront manipulables au moment de l'interrogation. Dans un système classique de recherche d'information dans un corpus de documents textuels, le texte est découpé éléments (termes, mots, formes, *token* en anglais) dont on conserve les occurrences d'appari-

tion à un certain niveau de granularité (document, position dans le texte, position dans la structure du document) pour permettre un accès efficient au moment de l'interrogation.

Dans notre problématique, nous ne nous intéressons pour le moment qu'à des traces d'interactions en français ou en anglais, et pouvons donc nous appuyer sur la notion de *mot*. (Cependant, dans le cas d'un éditeur de texte partagé, un mot peut se retrouver divisé entre deux événements distincts parce que le système de centralisation met à jour ses données sur un niveau de granularité temporel qui n'est pas relié au niveau sémantique de la notion de mot ; pour que le mot soit retrouvé au moment de l'interrogation, il faut soit grouper les événements au moment de l'indexation en tenant compte des natures des médias, soit retarder ce travail jusqu'au moment de l'interrogation, ce qui est plus pertinent, dans la mesure où l'on travaille sur des artéfacts créés dynamiquement).

La deuxième notion manipulée par un système de recherche d'information est celle de *document*. C'est au moment de l'indexation que cette notion est explicitée au système. En gardant la définition de *document* comme unité retournée par le système, il faudrait donc que les documents soient les phénomènes. Le problème est que ce qui est (sera) un phénomène va parfois être découvert par le chercheur lui-même, au fur et à mesure de son interaction avec le système, et ne peut dès lors pas toujours être connu au moment de l'indexation.

Nos besoins en informations semblent plus proches de la recherche de passage, focalisée sur la recherche d'extraits de documents qui traitent du sujet évoqué par la requête. Les passages peuvent être définis de multiples façons : unités linguistiques, unités sémantiques (Hearst, 1997), unités lexicales, unités structurelles, etc. Les techniques employées consistent à définir une notion de passage, puis expliciter ceux-ci et enfin à appliquer des techniques de RI *ad hoc* où les documents sont les passages ainsi définis. Quelques travaux se sont focalisés sur définir un modèle de recherche *ad hoc* basé sur une recherche de passage (Wilkinson, 1994).

Dans notre cas, définir les passages n'est pas plus possible que de définir les documents lorsque les phénomènes ne sont connus qu'une fois le travail du chercheur terminé. Il faut donc se tourner vers une branche de la RI où ce qui est cherché est défini dans la requête elle-même. Les travaux qui se placent dans cette branche sont pour la plupart reliés à une notion de structure, en particulier les travaux autour des langages de requêtes dans les documents XML. Trois langages de requête correspondent actuellement à des standards actifs : NEXI, XPath et XQuery (W3C 2008). NEXI, initiative d'INEX (INEX, 2008), présente certaines extensions pour la recherche de texte au sein d'éléments XML, mais n'est pas assez puissant en termes de manipulation des autres types de données. XPath est d'abord un langage de désignation au sein d'un document XML, et constitue de fait un simple sous-ensemble de XQuery dans la version 1.1 de ce langage. XQuery propose la quasi-totalité des outils nécessaires, mais n'est pas complètement adapté à notre problème dans la mesure où il ne permet de retrouver que des éléments de la structure de base des documents, alors que les phénomènes ne sont pas prédéfinis et qu'il n'y a aucune raison pour qu'ils coïncident

avec un élément, d'autant que, comme nous l'avons indiqué, la structure que nous manipulons, bien qu'étant du XML, ne reflète pas la structuration sémantique de haut niveau.

Notre besoin est de pouvoir définir, dans des requêtes, des regroupements d'événements qui vérifient un certain nombre de contraintes. L'algèbre de recherche dans les textes structurés définie par (Clarke *et al.*, 1994) se rapproche de nos besoins. S'appliquant à du texte pur, cette algèbre définit la notion d'ensemble de recouvrement d'un ensemble de termes. Dans le cas de documents structurés, les balises apparaissent au même niveau que les mots, et sont traitées quasiment de la même manière, de l'indexation à l'interrogation. Leurs apparitions créent une partition exacte de l'ensemble du texte, définissent ainsi les documents. Avec des balises qui indiquent le locuteur et le texte qu'il dit (que nous appelons *PV*, *production verbale*), la combinaison texte-structure permet de repérer, par exemple, des intervalles contenant une *PV* dite par *A* et contenant *a,* suivie d'une *PV* dite par *B* et contenant *b*. Ceci nous rapproche des besoins de notre problème où le chercheur doit pouvoir poser des requêtes en désignant les intervenants et des termes utilisés dans leurs interventions. Cependant, une notion qui n'est pas prise en compte par le formalisme de l'algèbre de Clarke et al. est le parallélisme entre plusieurs documents. Dans nos corpus, le regroupement des événements et leur annotations amènent des séquences en parallèle de séquences déjà existantes (par exemple, transcription, chat, editeur de texte).

Enfin une dernière fonctionnalité dont ont besoin les chercheurs est la généralisation des requêtes. Si le formalisme précité permet d'interroger la base en instanciant toutes les données, il ne permet pas de créer des requêtes quantifiées, c'est-à-dire de poser des requêtes avec des variables : par exemple, chercher des *PV* de *X* contenant *x,* suivies d'une *PV* de *Y* (*Y* différent de *X*) et contenant *x*. Cette fois *X*, *Y*, et *x* ne sont plus des instances mais des variables et une réponse devra non seulement fournir des intervalles, mais aussi instancier ces variables. On voit donc que les besoins vont au delà de ce que traite la recherche d'information traditionnelle, même si les prémisses sont les mêmes.

6. Aspects de l'implémentation

Il n'est pas possible de détailler ici les différents aspects du modèle implémenté dans Tatiana. Un rapport de recherche, à paraître en 2009 devrait en donner les principaux aspects, ainsi qu'une sémantique opérationnelle. Le modèle proposé associe différentes propriétés aux rejouables, l'une d'entre elles étant la *traçabilité*, qui permet, à partir de n'importe quel élément d'un rejouable, de retrouver sa source dans le corpus initial. Le modèle décrit différentes transformations applicables à un rejouable, et propose en particulier trois opérations de recherche d'information au sein d'un rejouable S :

$l\ f \vdash s\ S$ (*R-1*) fournit la séquence des items de S dont les facettes, sélectionnées par s, contiennent tous les éléments de l ; le résultat est une séquence d'items I_i, ordonnés comme ceux de S, tels que $s(I_i)$ contient les éléments de l.

$l\ f \Vdash s\ S$ (*R-2*) fournit une séquence de résultats, dont chacun est une séquence R_i d'éléments I_j, tels que chaque $s(I_j)$ contienne au moins un élément de l, et que l'union des $s(I_j)$ contient tous les éléments de l.

$l\ f \Vdash\!\!\vdash s\ S$ (*R-3*) fournit une séquence de résultats, dont chacun est une séquence R_i d'éléments I_j, tels que l'union des $s(I_j)$ contient tous les éléments de l.

Informellement, l'objet l est une séquence (ordonnée) ou un sac (sans ordre) d'objets, typiquement des chaînes de caractères représentant des mots, f un prédicat de comparaison, et s une fonction de sélection d'une facette spécifique des items ; l'opération prend en charge l'itération sur cette facette des items, et recherche les objets de l, soit (*R-1*) dans un item unique (opération de recherche classique, la cible étant l'item), soit (*R-2*) dans une séquence d'items (chaque item contenant au moins l'un des objets), soit encore (*R-3*) dans un média continu obtenu par la fusion de l'ensemble des facettes. Dans tous les cas, le résultat est l'ensemble des séquences d'items contenant l'ensemble des objets recherchés.

De manière générale, le modèle proposé permet de traduire une requête complexe en un algorithme basé sur les opérations de transformations (sélections, tris, recherches, fusions, unions, etc.) appliquées à un rejouable. Cette requête peut s'effectuer de manière automatique, dispensant dans certains cas le chercheur de procéder à des regroupements manuels, et simplifiant la visualisation et la découverte des phénomènes qu'il recherche.

7. Résultats

Dans notre cas d'étude 3 (et suite à des tentatives infructueuses de fouille de données), nous avons souhaité donner le moyen aux chercheurs d'identifier des phénomènes constitués d'une séquence d'interactions, par exemple d'identifier les passages où tel locuteur répond à une question de tel autre locuteur. Cette recherche passe par une série de transformations, dont l'utilisation de la fonction *R-2* sur les facettes « locuteur » ou « texte ». Afin d'augmenter l'efficacité de ces opérations de recherche, nous effectuons une indexation partielle de séquences (indexer toutes les séquences sur un corpus n'est pas une opération pratiquable, puisqu'elle serait en $O(n^2)$; cependant, la pertinence d'une séquence étant liée à la distance entre son premier et dernier élément, il s'est avéré utile d'indexer toutes les séquences sur une fenêtre de 1 à *m* éléments, nos expérimentations nous ayant montré que, sur ce corpus particulier, une valeur de $m = 15$ donnait des résultats significatifs ; ceci revient, implicitement, à introduire une notion de proximité).

Ayant effectué cette indexation, il s'avère que la comparaison entre l'index simple des termes (implémentation de l'indexation pour *R-1*) et celle des séquences peut nous

donner des informations sur l'espérance d'occurence de ces séquences. En comparant cette espérance avec les occurences réelles, nous pouvons déterminer une *mesure de surprise* d'une séquence donnée S, que nous calculons par :

$$\frac{|E_S - O_S|}{E_S}$$

(où E_S est l'espérance d'occurences de S si les événements étaient distribués selon leur fréquence d'apparition dans l'ensemble étudié, et O_S le nombre effectif d'occurences observées de la séquence S) afin de trier les séquences identifiées par ordre de surprise. Nous avons ensuite évalué la pertinence de ces séquences « suprenantes » en les soumettant à l'approbation des chercheurs dont l'analyse de cette réunion a occupé une part importante des deux dernières années. Certaines de nos trouvailles étaient immédiatement évidentes : nous avons identifié une séquence (pour laquelle notre mesure donnait une valeur particulièrement importante), où l'un des locuteurs intervenait à plusieurs reprises, alors que ce locuteur n'avait eu que 5 tours de parole sur une réunion de 90 minutes et que ces tours avaient été presque consécutifs. D'autres de ces trouvailles étaient connues - mais seulement grâce au fait que les chercheurs avaient passé énormément de temps (entre 200 et 300 heures) à analyser ce corpus : le fait qu'un des locuteurs parlait souvent chaque fois que la discussion abordait un nouveau point - en effet il se chargeait alors de résumer les enjeux relatifs à cette décision. Enfin, nous avons identifié des séquences surprenantes que nos collègues n'avaient pas détectées, et qui leur ont apporté une vision nouvelle sur le corpus : ainsi, le fait que le locuteur *A* et le locuteur *B* n'intervenaient presque jamais après le locuteur *C* leur a permis de mettre en évidence une distinction entre deux sous-groupes de participants dans la réunion, et de quantifier cette distinction. Ce dernier résultat a été particulièrement utile pour mettre en évidence la structure interactionnelle complexe d'une réunion qui ne semblait pas présenter de structure spécifique à ce niveau. Notons encore que certaines séquences ainsi découvertes ont été effectivement reconnues comme étant des « phénomènes » par les chercheurs, même s'ils ne parviennent pas encore à en donner d'interpétation.

Dans le cas 1, évoqué dans la section 2, le chercheur souhaitait identifier la reformulation entre un dialogue oral et des notes écrites. Il s'agissait donc simplement d'utiliser la RI pour faire de la recherche de passages en utilisant les éléments rédigés comme requêtes, afin de trouver ces passages dans la transcription. L'un des avantages majeurs de l'application de la RI pour nos collègues a été la possibilité d'identifier *tous* les passages sources potentiels. En effet, au premier abord une phrase particulière dans les notes provient probablement d'un énoncé de la transcription relativement proche dans le temps. Nous avons cependant pu mettre en évidence que la formulation de la note pouvait parfois provenir de plusieurs énoncés, et que ce qui semblait être une reformulation relativement majeure d'un énoncé était en réalité la combinaison de cet énoncé avec quelque chose qui avait été dit un quart d'heure plus tôt (cf. fig. 7.1).

Chaque chercheur désirant identifier des phénomènes différents, qui ont des caractéristiques différentes pour la modélisation de la requête, nous ne sommes pas encore en mesure de présenter tous les avantages de l'application de la RI aux problèmes de

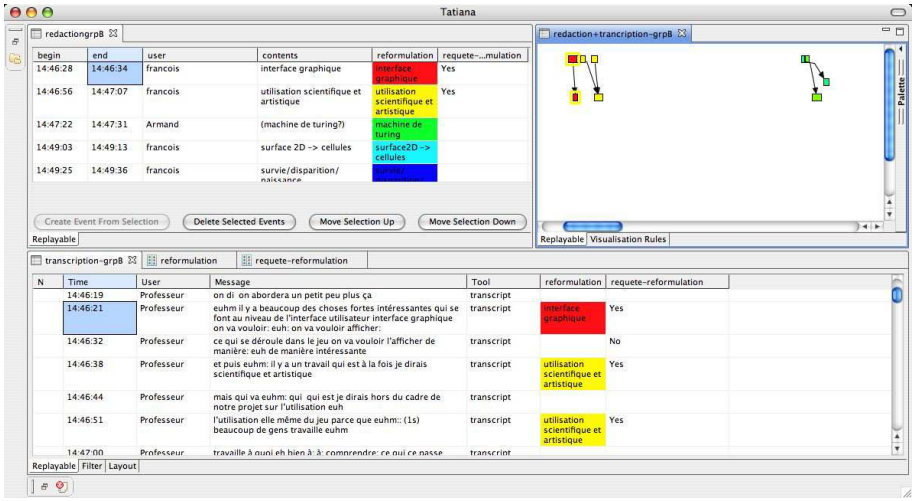


Figure 7.1. *Une capture d'écran du logiciel Tatiana. En haut, partie gauche, un affichage d'événements (après transformations) de l'éditeur de texte. En bas, la transcription après segmentation. En haut, partie droite, une visualisation temporelle linéaire des liens de reformulation trouvés par Tatiana.*

l'analyse des traces d'interaction. Nous nous sommes limités à implémenter quelques algorithmes classiques sur notre modèle de corpus, associés à un ensemble d'opérations de transformation des données, afin que ces premiers outils soient à la disposition des chercheurs souhaitant analyser leurs données. Plus d'une douzaine d'analyses sont actuellement en cours et utilisent ces outils, montrant leur pertinence pour ce domaine d'application.

8. Conclusions

Dans cet article, nous avons présenté le travail d'analyse dans une perspective socio-cognitive des interactions humaines médiatisées par des outils de CSCL et de CSCW, en montrant certaines des difficultés auxquelles sont confrontés les chercheurs désirant analyser des corpus composés de traces de telles interactions. Nous avons évoqué un modèle pour la représentation des corpus et pour leur interrogation, qui permet de rapprocher les problèmes d'analyse des chercheurs de problèmes de RI déjà connus. Le CSCL/CSCW acquiert un outil performant permettant la maitrise et l'analyse de corpus toujours plus grands, permettant aux chercheurs de travailler plus vite, et même d'identifier des phénomènes qui n'auraient pas été découverts manuellement.

9. Bibliographie

Avouris N., Fiotakis G., Margaritis M., Komis V., « Beyond logging of fingertip actions : analysis of collaborative learning using multiple sources of data. », *Journal of Interactive Learning Research*, vol. 18, n° 2, p. 231-250, 2007.

Clarke C. L., Cormack G., Burkowski F., An Algebra for Structured Text Search and A Framework for its Implementation, Technical Report n° CS-94-30, Dept. of Computer Science, Waterloo, Canada, August, 1994.

Corbel A., Girardot J.-J., Jaillon P., « DREW : "A Dialogical Reasoning Web Tool" », *ICTE2002, Intl. Conf. on ICT's in Education*, Badajos, Espagne, November, 2002.

Cox R., « Technology-enhanced research : educational ICT systems as research instruments », *Technology, Pedagogy and Education*, vol. 16, n° 3, p. 337-356, 2007.

De Chiara R., Di Matteo A., Manno I., Scarano V., « CoFFEE : Cooperative Face2Face Educational Environment », *Proceedings of the 3rd International Conference on Collaborative Computing : Networking, Applications and Worksharing*, New-York, USA, 2007.

Dyke G., Girardot J.-J., Lund K., Corbel A., « Analysing Face to Face Computer-mediated Interactions », *Developing Potentials for Learning*, Budapest, Hungary, august, 2007.

Erkens G., Janssen J., « Automatic coding of communication in collaboration protocols », *International Journal of Computer-Supported Collaborative Learning*, vol. 3, n° 4, p. ?, 2008.

Goodman B. A., Drury J., Gaimari R. D., Kurland L., Zarrella J., Applying User Models to Improve Team Decision Making, Technical Report n° 1351, Mitre, mitre.org, 2006.

Hearst M. A., « TextTiling : segmenting text into multi-paragraph subtopic passages », *Comput. Linguist.*, vol. 23, n° 1, p. 33-64, 1997.

INEX, « Initiative for the Evaluation of XML Retrieval », 2008. `http://www.inex.otago.ac.nz/`.

Lotan-Kochan E., « Analysing Graphic-Based Electronic Discussions : Evaluation of Students' Activity on Digalo », *in* , Springer (ed.), *EC-TEL 2006 : First European Conference on Technology Enhanced Learning*, Crete, Greece, p. 652-659, October, 2006.

Lund K., Prudhomme G., Cassier J.-L., « Using analysis of computer-mediated synchronous interactions to understand co-designers' activities and reasoning », *Proceedings of the International Conference On Engineering Design*, Cité des Sciences et de l'Industrie, Paris, France, august, 2007.

Minsky M., A Framework for Representing Knowledge, Technical Report n° MIT-AI Laboratory Memo 306, MIT, Boston, June, 1974.

W3C, « XQuery 1.1, W3C Working Draft 3 December 2008 », 2008. `http://www.w3.org/TR/2008/WD-xquery-11-20081203/`.

Wilkinson R., « Effective Retrieval of Structured Documents », *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, p. 311-317, July, 1994.

Proposition des cadres d'évaluation adaptés à un système de RI personnalisé

Mariam Daoud, Lynda Tamine-Lechani

Laboratoire IRIT
Université Paul Sabatier
118 Route de Narbonne
F-31062 TOULOUSE CEDEX 9
$\{daoud, lechani, bougha\}$*@irit.fr*

RÉSUMÉ. *L'évaluation d'un système de recherche d'information (RI) personnalisé consiste principalement à mesurer ses performances. Les cadres d'évaluation classiques en RI basés sur les approches orientées laboratoire méritent d'être étendues et révisées vu que le contexte de recherche de l'utilisateur n'est pas considéré dans le protocole d'évaluation et les collections de test. Nous présentons dans ce papier des cadres d'évaluation adaptés à un système de RI personnalisé. Ces cadres sont basés sur l'enrichissement des collections TREC par des contextes/profils utilisateur simulés. Plus précisément, un protocole issu de TREC adhoc consiste à construire des profils utilisateur à partir des sessions de recherche simulées par les domaines d'intérêts prédéfinis dans TREC adhoc. Le protocole issu de TREC HARD consiste à construire le profil à partir des sessions de recherche simulées par les sujets des requêtes de la collection. Les résultats obtenus confirment la stabilité de la performance de notre modèle de RI personnalisé selon les cadres proposés sur des collections de test différentes.*

ABSTRACT. *The evaluation of a personalized search systems aims at measuring its performance. Traditional laboratory-based evaluation of the information retrieval systems is challenged because of its lack of user evidence expressing his search context in the data test sets. In this paper, we present evaluation protocols of personalized search system based on enhancing TREC collections with simulated contexts/user profiles. An evaluation protocol based on TREC adhoc consists of building the user profile across simulated search sessions designed by the TREC domains. Another evaluation protocol based on TREC-HARD consists of building the user profile across simulated search defined by the topics of the collection. Experimental results show the stability performance of our personalized search system according to the proposed evaluation protocols using different test collections.*

MOTS-CLÉS : *Système RI personnalisé, cadre d'évaluation, profil utilisateur*

KEYWORDS: *Personalized IR system, evaluation protocol, user profile*

1. Introduction

L'objectif de l'évaluation d'un SRI est de mesurer ses performances vis à vis du besoin de l'utilisateur. Les protocoles d'évaluation largement adoptés en RI sont basés sur une approche de type laboratoire (*laboratory-based model*) initiée par *Cleverdon* [CLE 67] dans le cadre du projet *Cranfield project II*. Ce modèle fournit une base d'évaluation comparative de l'efficacité de différents algorithmes, des techniques et/ou des systèmes moyennant des ressources communes : des collections de test contenant des documents, des requêtes préalablement construites et des jugements de pertinence associés construits selon la technique de *pooling*, des métriques d'évaluation essentiellement basées sur le rappel-précision. L'émergence de la RI orientée utilisateur a cependant remis en cause la viabilité de ce modèle pour l'évaluation de systèmes interactifs ou de manière générale, les systèmes d'accès contextuel à l'information [ING 05]. Ces systèmes ont pour objectif de délivrer de l'information pertinente et appropriée au contexte de l'utilisateur qui a émis la requête.
Ceci a motivé les réflexions autour de méthodologies d'évaluation adaptées à une recherche d'information contextuelle. Nous proposons dans ce papier des méthodologies d'évaluation mettant en jeu des contextes utilisateurs simulés représentés par leur profil dans le but d'évaluer notre modèle d'accès personnalisé à l'information. Notre approche de base consiste à exploiter un profil utilisateur à court terme construit sur une session de recherche dans la chaine d'accès à l'information. Les cadres d'évaluation proposés sont basés sur l'enrichissement des ressources TREC [VOO 01] par des contextes utilisateurs simulés. Le profil utilisateur est intégré comme étant une composante principale de la collection de test. La notion de session est intégrée dans la stratégie d'évaluation du modèle de RI et est soit simulée par le domaine d'annotation des requêtes dans le cas du cadre d'évaluation issu de TREC *adhoc*, soit simulée par des sous-requêtes d'une même requête dans le cas du cadre d'évaluation issu de HARD TREC.
Ce papier est organisé comme suit. La section 2 présente les cadres d'évaluations proposés en RI personnalisée. Le section 4 est dédiée à la description des cadres d'évaluation *TREC adhoc* et *TREC- HARD*. La section 5 conclut et présente les perspectives de nos travaux dans le domaine.

2. Évaluation des modèles d'accès personnalisé à l'information

A ce jour, il n'existe pas un cadre standard pour l'évaluation de l'efficacité d'un modèle d'accès personnalisé à l'information. Les approches classiques d'évaluation des performances en RI sont des approches de type laboratoire. Ce type d'évaluation révèle des limitations d'évaluation des SRI avec l'émergence de la RI contextuelle.

2.1. *Limites de l'évaluation orientée laboratoire*

L'évaluation orientée laboratoire d'un SRI est principalement fondée sur l'utilisation d'une collection de test où les requêtes sont les seules ressources clés qui traduisent le besoin en information de l'utilisateur. Les principales limitations de ce cadre d'évaluation sont liées à l'inadéquation des collections de tests pour l'évaluation de la recherche d'information en contexte [TAM 09, ING 05]. D'une part, les utilisateurs directs ayant émis ces requêtes, leurs centres d'intérêt et interactions avec le SRI ne font pas partie intégrante de la collection. D'autre part, les jugements de pertinence sont thématiques et indépendants des situations et du contexte de recherche. Or, il a été bien montré dans [BOR 03, MIZ98] que la notion de pertinence est plus complexe, couvrant des niveaux divers liés à la situation de recherche en cours : pertinence cognitive, pertinence affective, pertinence situationnelle, etc.
Les premières tentatives faites dans le cadre de l'évaluation des systèmes de RI en contexte ont été proposées dans TREC à travers les tâches Interactive et HARD. Ces taches ont permis l'intégration des métadonnées concernant l'utilisateur dans le processus de RI afin d'augmenter la performance du système pour des requêtes difficiles. Les métadonnées utilisateurs concernent des critères tels que la familiarité, le genre du document, la langue, etc. Étant très spécifiques, ces tâches ne permettent pas d'évaluer un système de RI personnalisé intégrant des dimensions du contexte plus large tels qu'un profil utilisateur à centres d'intérêts multiples, un profil utilisateur mobile, etc. Ceci a conduit l'émergence des approches d'évaluations fondées sur l'utilisation des contextes de recherche simulés ou des contextes réels.

2.2. *Emergence des approches d'évaluation orientée contexte*

2.2.1. *Évaluation par simulation de contextes*

L'évaluation d'un système de RI personnalisé par simulation de contextes consiste à définir des contextes de recherche qui simulent des utilisateurs et interactions hypothétiques. Le protocole d'évaluation présenté dans [TAM 07, DAO 08] repose sur l'extension des cadres d'évaluation TREC via l'enrichissement de la collection de test par des centres d'intérêts ou profils utilisateurs simulés. Ces contextes sont créés sur la base des interactions hypothétiques fournies par les jugements de pertinence de TREC. L'approche dans [SIE 07] utilise une collection de test issue d'une ontologie Web prédéfinie et l'enrichit par des requêtes utilisateurs et des contextes de recherche simulés. Le contexte de l'utilisateur est créé comme étant son profil traduisant un centre d'intérêt et est représenté par un concept de l'ontologie. La requête est générée automatiquement par les termes représentatifs du concept. Quant aux jugements de pertinence, un document retourné par le système est considéré comme pertinent s'il est classifié dans le concept/centre d'intérêt simulé.
Les mesures d'évaluation de performance du système sont principalement basées sur les mesures classiques de rappel et précision [TAM 08, DAO 08] dans le cas d'une collection de test prédéfinie (e.g. TREC). Le référentiel d'évaluation d'une requête is-

sue d'un concept/centre d'intérêt de l'ontologie dans [SIE 07] est construit sur la base des documents Web classifiés sous ce concept. Généralement, l'évaluation par simulation de contextes présente deux avantages majeurs : (1) n'est pas couteuse en temps puisqu'elle n'implique pas des utilisateurs réels (2) permet d'effectuer une évaluation comparative.

2.2.2. *Évaluation par utilisation des contextes réels*

Ce type d'évaluation est basé sur une étude de cas de vrais utilisateurs dans des contextes de recherche réels permettant d'intégrer les interactions utilisateur-SRI.
Deux types d'évaluation par utilisation des contextes réels peuvent être adoptés ; le premier type est basée sur une approche hybride [SHE 05] qui intègre une collection de test prédéfinie (e.g. TREC) dans un protocole d'évaluation impliquant des vrais utilisateurs. Cette approche étend les collections TREC par des contextes représentés par l'historique de recherche de l'utilisateur englobant les requêtes et les documents *cliqués*. Chaque requête de la collection TREC est reformulée par les utilisateurs dans le but de définir des requêtes reliées à un même besoin en information et par la suite définir une session de recherche.
Le deuxième type d'évaluation utilise une collection Web via une interface (telle que Google API) et des vrais utilisateurs qui formulent leur requête selon un besoin spécifique. Les interactions des utilisateurs (tels que les *clics*, temps passé sur une page, etc.) seront enregistrés dans un fichier log exploité dans l'étape de l'évaluation de performance du système. L'approche dans [CHA 07] adopte ce type d'évaluation où les données de *clics* sur un document sont un indice de pertinence du document. Ces documents *cliqués* sont utilisés afin de construire le profil utilisateur en tant que hiérarchie de concepts pondérés traduisant le contexte de recherche.
Les mesures d'évaluation de performance du système peuvent être des mesures classiques de rappel et précision et leur dérivée. Ceci est faisable dans l'évaluation par utilisation des contextes réels grâce à la construction d'un référentiel de pertinence sur la base de l'ensemble des documents jugés pertinents par les différents utilisateurs pour une même requête [LIU 04]. D'autres mesures orienté rang telles que le rang moyen [CHA 07] et DCG (Discounted Cumulative Gain) [JAR 02] sont également utilisées et se basent sur la position des documents pertinents dans la liste de résultats. Dans ce cas, les utilisateurs jugent les Top N documents retournés par le système.
Les principales limitations de ce type d'évaluation sont liées au coût en temps et à la reproductibilité des résultats et par conséquent à l'inconsistance de l'évaluation comparative.

3. Proposition des cadres d'évaluation d'un système de R I personnalisé à base de sessions

Vu qu'il n'existe pas un cadre standard d'évaluation d'un système de RI personnalisé à base de sessions, nous proposons des cadres d'évaluation issus de collections TREC adaptées à un accès personnalisé à l'information par simulation de contextes.
Dans ces cadres d'évaluation, le contexte simulé représente le profil utilisateur construit

sur une session de recherche. Le cadre d'évaluation issu de TREC *adhoc* est proposé dans le but d'évaluer un modèle de RI personnalisé avec une délimitation prédéfinie des sessions de recherche. Le cadre d'évaluation issu de HARD TREC est proposé dans le but d'évaluer un modèle de RI personnalisé en l'absence des relations de corrélation entre les requêtes de la collection.

3.1. *Le cadre d'évaluation issu de TREC ad-hoc*

Ce cadre d'évaluation a été initialement défini pour l'évaluation de l'accès personnalisé guidé par le profil utilisateur, basé mots clés [TAM 08]. Nous étendons ce même cadre pour supporter un profil utilisateur basé sur un graphe de concepts issu d'une ontologie Web prédéfinie.

3.1.1. *Collection de test*

A. Requêtes

Le choix des requêtes la collection *TREC 1* est guidé par le fait qu'elles sont annotées d'un champ particulier noté " **Domain**" qui décrit un domaine d'intérêt traité par la requête. C'est à juste titre, cette métadonnée qui sera exploitée pour simuler des utilisateurs hypothétiques avec des profils issus de ces domaines.

B. Collection de documents

La collection de test de la campagne d'évaluation *TREC 1 adhoc* utilisée dans ce cadre, est celle des disques 1, 2 et 3. Les documents de cette collection sont issus de différents articles de presse tels que *Associate Press (AP), Wall street journal (WJS), Financial times*.

C. Le profil utilisateur

Le profil utilisateur est un élément intégré dans la collection de test selon un algorithme de simulation qui le génère à partir des requêtes du même domaine décrit comme suit :

1) pour chaque domaine k de la collection (noté Dom^k avec $k = (1..6)$), nous sélectionnons, parmi les n requêtes associées à ce domaine, un sous-ensemble de $n-1$ requêtes qui constitue l'ensemble des requêtes d'apprentissage,

2) à partir de cet ensemble d'apprentissage, un processus automatique se charge de construire le profil utilisateur selon la méthode propre du système de RI personnalisé.

3.1.2. *Stratégie d'évaluation*

La stratégie de validation consiste en un scénario qui se base sur la méthode de la validation croisée et ce, pour ne pas biaiser les résultats avec un seul jeu de test. Nous considérons ici que les sessions de recherche sont définies préalablement par l'ensemble de requêtes annotées des domaines de TREC. La validation croisée [MIT 97]

ou la *k-fold cross validation* est une méthode d'évaluation qui consiste à diviser la collection de requêtes de test en k sous ensembles de tailles égales (approximativement), d'utiliser $k-1$ sous ensembles pour l'apprentissage du profil utilisateur, et le k^{ime} sous ensemble pour le test. Dans notre cas, on subdivise l'ensemble des n requêtes du domaine en un sous-ensemble d'apprentissage de $n-1$ requêtes pour apprendre le profil utilisateur et en un sous-ensemble de test contenant la n^{ime} requête à tester.

3.2. *Le cadre d'évaluation issu de TREC HARD 2003*

Ce cadre est défini dans le but d'évaluer l'efficacité un modèle de RI personnalisé sur des requêtes difficiles et en l'absence d'une connaissance préalable de corrélation entre ces requêtes. Ce cadre d'évaluation consiste à définir une stratégie de test permettant d'évaluer l'efficacité du modèle à travers une séquence de sessions de recherche simulées traitant de sujets différents.

3.2.1. *Collection de test*

A. Requêtes

Le choix des requêtes de la collection *HARD* TREC 2003 a pour but d'augmenter la précision de recherche sur des requêtes difficiles. Vu qu'aucune information concernant la corrélation entre ces requêtes n'existe, nous procédons par la définition des sous-requêtes issues d'une même requête. La requête principale représente un sujet auquel les sous-requêtes générées sont rattachés définissant une session de recherche. Le processus de génération des sous-requêtes d'une même requête est détaillé comme suit :

1) Extraire le profil *pertinence* de la requête principale q en construisant l'ensemble des N vecteurs documents pertinents associés extraits du fichier de jugements de pertinence fourni par TREC, soit dp_q,

2) Subdiviser ce profil en p sous-profils, notés sp_i, $sp_i \subset dp_q$,

3) Pour chaque sous-profil *pertinence* sp_i, créer un vecteur centroïde selon la formule : $c_i(t) = \frac{1}{|sp_i|} \sum_{d \in sp_i} w_{td}$, w_{td} est le poids du terme t dans le document d calculé selon la fonction de pondération classique $tf * idf$,

4) Représenter la sous-requête par les k termes les mieux pondérés du vecteur centroïde,

5) Eliminer les documents pertinents dp_q de la requête de la collection de test.

B. Collection de documents

Le corpus HARD comprend des documents comprenant des textes issus du *NewsWire 1 999, AQUAINT corpus et U.S. government.*

C. Le profil utilisateur

Le principe de construction du profil utilisateur est analogue à celui décrit dans TREC *adhoc*. Dans ce cadre précisément, (1) la notion de domaine, clairement identifié dans

le cas de la collection *TREC ad-hoc* est remplacée par la notion de sujet de requête principal, non connu *a priori*, (2) les requêtes associées aux domaines, sont remplacées par les sous-requêtes associées à la requête principale en cours de traitement, (3) les requêtes servant à la construction du profil sont des sous-requêtes corrélées le long d'une séquence de sessions de recherche simulés.

3.2.2. *Stratégie d'évaluation*

La stratégie de validation dans ce protocole consiste à diviser l'ensemble des requêtes en un ensemble de requêtes d'apprentissage permettant de paramétrer le système (définir le seuil du mécanisme de délimitation de sessions de recherche) et un ensemble de requêtes de tests permettant d'évaluer notre modèle.

A. Phase d'apprentissage

Cette phase est une étape préliminaire dont le but est de déterminer le seuil de corrélation optimal à partir d'une séquence des sessions d'apprentissage. Cette phase consiste à calculer les valeurs de corrélation requête-profil le long d'une séquence des sessions d'apprentissage. Une séquence des sessions d'apprentissage est définie par alignements successifs des sous-requêtes des requêtes d'apprentissage. Les valeurs de corrélations sont calculées entre chaque sous-requête traitée de la séquence et le profil utilisateur crée sur l'ensemble des sous-requêtes précédentes et liés à une même requête.

Pour chaque valeur de seuil de corrélation obtenu, on calcule la précision de détection des requêtes corrélées P_{intra} et celle de délimitation de sessions de recherche P_{inter} selon les formules suivantes :

$$P_{intra}(\sigma) = \frac{|CQ|}{|TCQ|}, P_{inter}(\sigma) = \frac{|FQ|}{|TFQ|} \quad [1]$$

où $|CQ|$ est le nombre de sous-requêtes correctement classifées comme corrélées, $|TCQ|$ est le nombre total de sous-requêtes devant être identifées comme corrélées sur la séquence, $|FQ|$ est le nombre de sous-requêtes indiquant correctement des frontières de sessions de recherche et $|TFQ|$ est le nombre total de frontières de sessions de la séquence.

Le seuil de corrélation optimal $\sigma*$ est ensuite identifé pour des valeurs de précisions maximales de ($P_{intra}(\sigma)$ et $P_{inter}(\sigma)$). En effet, le seuil optimal est calculé comme suit :

$$\sigma* = argmax_{\sigma}(P_{intra}(\sigma) * P_{inter}(\sigma)) \quad [2]$$

Ce seuil de corrélation est exploité dans la phase de test dans le but de classifer des sous-requêtes dans une même session le long d'une séquence des sous-requêtes de tests.

B. Phase de test

La phase de test est basée sur l'évaluation du système de RI personnalisé le long d'une séquence de sessions issue d'un ensemble de requêtes de tests traitant de sujets différents. Cette évaluation consiste à comparer la performance de recherche classique

(requête seule) à la performance de recherche personnalisée (requête et profil associé). Toute sous-requête de la séquence traitée ayant une valeur de corrélation plus grande que le seuil optimal est considérée corrélée au profil construit dans la session en cours de traitement. Par conséquent, le profil utilisateur de la session est utilisé dans le processus de RI personnalisé de cette sous-requête.
Notons que les documents pertinents ayant servi à la création des profils utilisateurs dans cette phase ne sont pas considérés pour l'évaluation des performances associées à ces sous-requêtes. Ceci permet en effet de ne pas biaiser les résultats dans le sens des documents pertinents déjà considérés dans la création du profil.

4. Mise en œuvre et résultats

Nous avons mis en œuvre ces cadres d'évaluation dans le but de tester notre système d'accès personnalisé à l'information. Nous présentons dans la suite notre système de RI personnalisé ainsi que l'évaluation de son efficacité selon les deux cadres d'évaluation proposé.

4.1. *Conception d'un système d'accès personnalisé à l'information à base de sessions*

Notre approche en RI personnalisée porte sur la définition d'un profil utilisateur selon un graphe de concepts sémantiquement liés et issus d'une ontologie de référence, l'ODP [DAO 09]. Le profil utilisateur est construit sur une session de recherche définie comme étant une séquence de requêtes liées à un même besoin en information. Notre approche peut être décrite selon deux principaux composantes : (1) La construction du profil utilisateur sur une session de recherche, (2) la personnalisation du processus de recherche.

4.1.1. *Construction du profil utilisateur selon un graphe issu d'une ontologie*

Le processus de construction du profil utilisateur consiste principalement à combiner les profils des requêtes de la même session. En effet, la methode se résume par le suivant :

– Pour chaque requête de la session, on extrait la liste des vecteurs associés aux documents pertinents dpq de chaque requête q. Partant des vecteurs documents, un vecteur basé mots clés appelé contexte de la requête est construit puis projeté sur l'ontologie de l'ODP aboutissant à la construction du profil G_q^s de la requête q soumise à l'instant s.

– Le profil utilisateur est ainsi construit par combinaison des profils des requêtes d'une même session. Les requêtes sont groupées dans des sessions selon un mécanisme de délimitation des sessions de recherche. Celui ci est basé sur la corrélation de rangs des concepts du profil courant avec ceux de la nouvelle requête soumise. Le

profil utilisateur est alors représenté par un graphe de concepts sémantiquement liés via l'ontologie de l'ODP, noté G_u^s.

4.1.2. *La personnalisation du processus de recherche*

Le profil utilisateur G_u^s construit sur la base d'une session de recherche est exploité dans le réordonnancement des résultats de recherche d'une requête q^{s+1} de la même session. Notre fonction de réordonnancement est basée sur la combinaison des scores d'appariement original et contextuel du document :

$$S_f(d_k) = \gamma * S_i(q, d_k) + (1 - \gamma) * S_c(d_k, G_u^s) \quad [3]$$

$$0 < \gamma < 1$$

Le score contextuel du document est calculé selon une mesure de similarité entre son vecteur représentatif d_k et le vecteur contextuel représentatif du profil adéquat G_u^s.

$$S_c(d_k, G_u^s) = \frac{1}{h}. \sum_{j=1..h} score(c_j) * cos(\vec{d_k}, \vec{c_j}) \quad [4]$$

Où c_j représente un concept du profil, $score(c_j)$ est le poids de la catégorie c_j dans le vecteur contextuel et h est le nombre de concepts du profil utilisateur considérés dans le réordonnancement des résultats.

4.2. *Résultats expérimentaux*

La mise en œuvre des cadres d'évaluation proposés a pour objectif d'évaluer l'efficacité de notre modèle d'accès personnalisé à l'information intégrant le profil utilisateur dans le processus de recherche. Cette évaluation consiste à comparer les résultats obtenus par notre modèle aux résultats obtenus par la recherche classique ignorant le profil utilisateur.
Nous avons mené nos expérimentations en utilisant le moteur de recherche "*MERCURE*" et suivant la stratégie de test associée à chacun des protocoles. Le modèle de recherche classique est basé sur la fonction d'appariement BM25 donnée dans la formule suivante :

$$w_{td} = tf_d \times \frac{log(\frac{n - n_t + 0.5}{n + 0.5})}{K_1 \times ((1 - b) + b \times \frac{dl}{avgdl}) + tf} \quad [5]$$

où tf_d est la fréquence du terme t dans le document d, n est le nombre total des documents de la collection de test et n_t est le nombre de documents contenant le terme t, $K_1 = 2$ and $b = 0.75$.
Le modèle de RI personnalisé est basé sur le réordonnancement des résultats de recherche de la requête utilisant le profil avec $\gamma = 0,3$ dans l'équation (3) et $h = 3$ dans l'équation (4).
Concernant le protocole TREC adhoc, nous avons simulé six domaines présentées

dans le tableau 1 dont les requêtes sont numérotées de 51 à 100. Quant au protocole TREC HARD, nous avons utilisé 30 requêtes de HARD TREC subdivisé en 15 requêtes d'apprentissage et 15 requêtes de tests définissant respectivement la séquence des sessions d'apprentissage et celle de test.
Le profil utilisateur est construit selon un processus automatique qui se charge de récupérer tout d'abord la liste de premiers documents pertinents fournies par TREC ($\|dpq\| = 10$) pour chaque requête d'apprentissage dans le cas du protocole TREC adhoc ou la liste des documents du sous-profil pertinence ($\|sp_i\| = 10$) de chaque sous-requête dans le cas du protocole HARD TREC. Partant de cette liste de documents, le profil de la requête est construit, puis le profil utilisateur est défini par combinaison des profils des requêtes d'apprentissage dans TREC ad hoc ou des profils des sous-requêtes corrélées selon le seuil optimal dans HARD TREC.
Les résultats obtenus sont présentés en termes de précision et rappel calculées à différents points (5, 10, .., 100 premiers documents restitués). Les résultats présentés dans la figure 1 sont obtenus selon le protocole TREC adhoc et montrent un taux d'accroissement significatif de notre modèle sur l'ensemble des requêtes de tests. Plus précisément, les pourcentages d'amélioration sont de 10% et de 11.6% respectivement pour le rappel au Top-10 rappel et la précision au Top-10. La figure 2 montrent les résultats obtenus selon le protocole TREC HARD. Nous pouvons constater une amélioration significative pour notre modèle aussi bien selon la mesure du rappel que de la précision sur les n premiers documents restitués par le système. Plus précisément, les pourcentages d'amélioration sont de 23.6% et de 6% respectivement pour le rappel au Top-10 rappel et la précision au Top-10.
La différence du taux d'efficacité de la recherche personnalisé sur les deux collections TREC est du à plusieurs facteurs. Le premier concerne le degré de corrélation des requêtes d'un même domaine (TREC adhoc) ou des sous-requêtes d'une même requête (HARD TREC). Le deuxième facteur est lié à la précision du mécanisme de délimitation des sessions de recherche. En effet, ce mécanisme a un impact sur la précision de représentation du profil utilisateur qui est créé sur la base des requêtes identifiés comme corrélées selon le seuil optimal. Toutefois, les résultats montrent bien un pourcentage d'amélioration significatif sur les deux collections ce qui montre la stabilité de performance de notre modèle de RI personnalisé.

Domaines	Requêtes
Environment	59 77 78 83
Military	62 71 91 92
Law and Government	70 76 85 87
International Relations	64 67 69 79 100
US Economics	57 72 84
International Politics	61 74 80 93 99

Tableau 1. *Domaines de TREC choisies pour la simulation des profils utilisateurs*

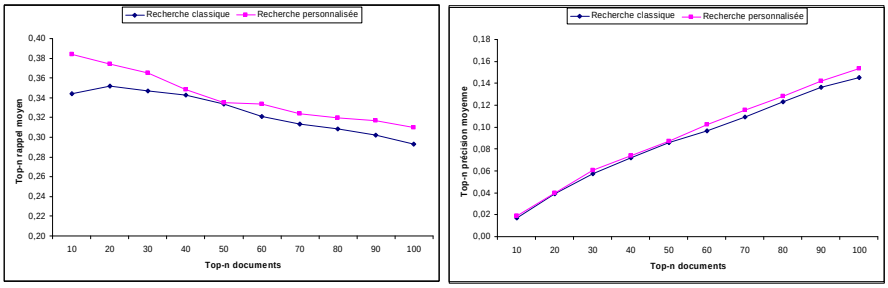


Figure 1. *Evaluation de performance du modèle de RI personnalisée en termes de Top-n précision moyenne et Top-n rappel moyen sur TREC adhoc*

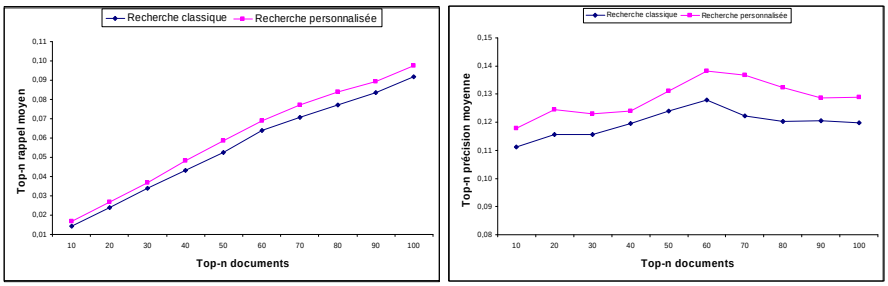


Figure 2. *Evaluation de performance du modèle de RI personnalisée en termes de Top-n précision moyenne et Top-n rappel moyen sur H A R D T R E C*

5. Bilan et perspectives

Les cadres d'évaluation que nous avons proposés dans ce papier suivent une méthodologie de simulation du profil de l'utilisateur par découpage thématique des sessions de recherche en utilisant des ressources *TREC*. Notre approche de personnalisation proposée intègre le profil utilisateur construit sur une session de recherche dans le processus de recherche d'information. L'évaluation de notre modèle selon les deux cadres montre bien la stabilité de performance des résultats sur des collections de test différentes.

Les perspectives de recherche ouvertes par ce travail portent sur l'amélioration des cadres d'évaluation proposés. Ces cadres peuvent être étendus dans le sens d'adapter une stratégie d'évaluation basée sur l'utilisation des données réelles issues du *log* des interactions utilisateurs. Plus généralement, nous envisageons de définir dans le futur un cadre d'évaluation standard des systèmes d'accès contextuel à l'information guidé par les centres d'intérêts de l'utilisateur.

6. Bibliographie

[BOR 03] BORLUND P., « The IIR evaluation model : A framework for evaluation of interactive information retrieval systems », *Journal of Information Research*, vol. 8, n° 3, 2003, page 152.

[CHA 07] CHALLAM V., GAUCH S., CHANDRAMOULI A., « Contextual Search Using Ontology-Based User Profiles », *Proceedings of RIAO 2007, Pittsburgh USA*, 2007.

[CLE 67] CLEVERDON C., « The Cranfield test on index language devices », *Aslib*, 1967, p. 173-194.

[DAO 08] DAOUD M., TAMINE L., BOUGHANEM M., « Using a concept-based user context for search personalization », *International Conference of Data Mining and Knowledge Engineering (ICDMKE), London, UK*, 2008, p. 57-64.

[DAO 09] DAOUD M., TAMINE L., BOUGHANEM M., CHEBARO B., « A Session Based Personalized Search Using An Ontological User Profile », *ACM Symposium on Applied Computing (SAC), (USA)*, ACM, march 2009, p. 1031-1035.

[ING 05] INGWERSEN P., JARVELIN K., *The turn : Integration of information seeking and retrieval in context*, Springer, 2005.

[JAR 02] JARVELIN K., KEKÄLÄINEN J., « Cumulated gain-based evaluation of IR techniques », *ACM Trans. Inf. Syst.*, vol. 20, n° 4, 2002, p. 422–446, ACM.

[LIU 04] LIU F., YU C., MENG W., « Personalized Web Search For Improving Retrieval Effectiveness », *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, n° 1, 2004, p. 28-40.

[MIT 97] MITCHELL T. M., « Machine Learning, McGraw-Hill Higher Education », , 1997.

[MIZ98] MIZZARO S., « How many relevances in information retrieval ? », *Interacting with Computers*, vol. 10, n° 3, 1998, p. 303–320.

[SHE 05] SHEN X., TAN B., ZHAI C., « Context-sensitive information retrieval using implicit feedback », *Proceedings of the 28th annual international ACM SIGIR conference*, New York, NY, USA, 2005, ACM, p. 43-50.

[SIE 07] SIEG A., MOBASHER B., BURKE R., « Web search personalization with ontological user profiles », *CIKM'07: Proceedings of the sixteenth ACM conference on information and knowledge management*, New York, NY, USA, 2007, ACM, p. 525-534.

[TAM 07] TAMINE L., BOUGHANEM M., ZEMIRLI W., « Exploiting Multi-Evidence from Multiple User's Interests to Personalizing Information Retrieval », *IEEE International Conference on Digital Information Management(ICDIM 2007)*, , 2007.

[TAM 08] TAMINE L., BOUGHANEM M., ZEMIRLI W. N., « Personalized document ranking : Exploiting evidence from multiple user interests for profiling and retrieval », *Journal of Digital Information Management*, vol. 6, n° 5, 2008, p. 354–365, Digital Information Research Foundation (DIRF).

[TAM 09] TAMINE L., BOUGHANEM M., DAOUD M., « Evaluation of contextual information retrieval : overview of issues and research », *Knowledge and Information Systems (Kais)*, , 2009, Springer.

[VOO 01] VOORHEES E. M., « Overview of TREC 2001 », *TREC*, 2001.

Recherche d'Entités Nommées dans les Journaux Radiophoniques par Contextes Hiérarchique et Syntaxique

Azeddine Zidouni*, Hervé Glotin, Mohamed Quafafou***

** Laboratoire LSIS, Univ. Aix-Marseille 2.*
{azeddine.zidouni,mohamed.quafafou}@univmed.fr

*** Laboratoire LSIS, Univ. Sud Toulon Var.*
glotin@univ-tln.fr

RÉSUMÉ. *Ce papier présente une approche pour la recherche d'entités nommées dans des transcriptions radiophoniques. Nous allons utiliser les structures des entités nommées afin d'améliorer le taux de leur reconnaissance. En effet, l'espace des entités peut être représenté par une structure hiérarchique (arbre). Ainsi, un concept peut être vu comme un noeud dans l'arbre, et une entité comme un parcours dans la structure de l'espace. Nous allons montrer l'apport de cette représentation en utilisant le modèle des Champs Aléatoires Conditionnels (CAC). La comparaison de notre approche avec la méthode des Modèles de Markov Cachés (MMC) montre une amélioration de la reconnaissance en utilisant les CAC Combinés. Nous montrons également l'impact de l'utilisation des informations a priori dans le processus en incluant les informations syntaxiques des transcriptions comme nouveau contexte.*

ABSTRACT. *This paper focuses on the role of structures in the Named Entity retrieval inside audio transcription. We consider the transcription documents strucure that guides the parsing process, and from which we deduce an optimal hierarchical structure of the space of concepts. Therefore, a concept is represented by a node or any sub-path in this hierarchy. We show the interest of such structure in the recognition of the Named-Entities using the Conditional Random Fields (CRF). The comparison of our approach to the Hiden Markov Model (HMM) method shows an important improvement of recognition using Combining CRFs. We also show the impact of the Part-of-Speech tagging (POS tagging) in the prediction quality.*

MOTS-CLÉS : *Reconnaissance d'entités nommées, Champs conditionnels aléatoires, Structures hierarchiques, Recherche d'information.*

KEYWORDS: *Named Entity Research, CRFs, Hierarchical Structure, Information Retrieval.*

1. Introduction

Les langues naturelles sont vivantes, elles traduisent une certaine évolution des pensées. Cette liaison entre la pensée et la langue rend le Traitement Automatique de la Langue Naturelle (TALN) très complexe. La langue offre de nombreuses façons d'exprimer une idée et une multitude de formes pour la représenter, plusieurs sens peuvent être associés à un mot selon son contexte d'utilisation. Les modèles de recherche d'information basés sur la sémantique tentent de représenter le fond de l'information de façon indépendante de la forme (ex. la syntaxe dans le texte). Cette tâche est d'autant plus complexe qu'il n'existe pas de bijection entre les mots et les sens associés. Les *Entités Nommées* (EN) sont des entités du monde réel, dont la forme linguistique est une représentation directe qui varie selon son contexte. L'extraction des EN dans un contexte journalistique représente une tâche importante dans la chaîne de l'analyse sémantique. Une EN représente une description conceptuelle qui fait référence à un objet[1] dont la représentation linguistique est unique. Dans le domaine de l'indexation sémantique, la représentation conceptuelle d'un objet prime sur sa représentation linguistique. En effet, avoir la description d'un objet dans une phrase nous permettra d'identifier plus facilement les objets porteurs de sens dans son contexte.

Dans les traitements textuels, on distingue deux types d'EN : les entités singulières, et les entités composées. Les entités singulières représentent généralement des noms propres, la présence de majuscules est un bon indicateur pour ce type. Les entités composées : le séquencement d'un ensemble de mots, insignifiants à l'origine, peut faire référence à un concept particulier (organismes, dates, etc.). Cette dernière rend la tâche encore plus difficile, car en plus du problème de la reconnaissance d'EN, s'ajoute le problème de la segmentation. En effet, on cherche à attribuer à chaque mot une étiquette signifiant l'existence d'une EN. Or, une étiquette peut s'étendre sur plusieurs mots. Plusieurs approches sont proposées afin de palier cette contrainte (modèles markoviens). La particularité du traitement des EN dans le cadre radiophonique réside dans la spontaniété de la parole, les entités différant d'une source à une autre. En effet, chaque source (radio) implique différents intervenants avec des origines différentes (des vocabulaires différents). Ces différences rendent la tâche de création de modèles génériques de prédiction complexe. En outre, les EN possèdent un cycle de vie (elles ont une forte apparition pendant une durée puis elles disparaissent). D'où la nécessité de construire des modèles qui se basent sur le contexte d'apparition des EN.

L'article est structuré comme suit. La section 2 décrit le problème de la *Recherche d'Entités Nommées* (REN) en utilisant les modèles graphiques. Dans la section 3, nous allons définir la structure d'arbre des EN extraite à partir du corpus de données, et expliquer le principe de la méthode utilisée pour la REN. La section 4 présente les résultats expérimentaux obtenus sur le corpus d'ESTER (Sylvain Galliano *et al.,* n.d.) en considérant les précisions des prédictions ainsi que le temps pris pour la construction des modèles. Nous allons conclure en section 6.

1. Une entité du monde réel qui est représenté par un ou plusieurs mots qui portent un sens particulier.

2. Etiquetage de données séquentielles

Un modèle séquentiel est un modèle graphique dans lequel les liaisons entre certaines variables sont spécifiques. Les modèles graphiques nous permettent d'attribuer des classes prédéfinies aux variables (problème de classification), mais le vrai point fort de ces approches est leur capacité de modéliser plusieurs variables interdépendantes (Sutton *et al.,* 2006). Les modèles de séquences supposent avoir les variables d'entrées X sous une forme séquentielle. Le problème d'étiquetage de séquences peut être formalisé de la manière suivante : Etant donnée $X = \langle x_1, x_2, \ldots x_n \rangle$ une séquence de données d'observation (données d'entrée), trouver la séquence d'états $Y = \langle y_1, y_2, \ldots y_n \rangle$ (séquence d'étiquettes associées aux données d'entrée) qui maximise la probabilité conditionnelle $P(Y|X)$:

$$Y = argmax P(Y|X). \quad [1]$$

Dans ce qui suit, nous donnons les différents modèles markoviens et les fondements théoriques des *Champs Aléatoires Conditionnels* (CAC).

2.1. *Les Modèles de Markov Cachés*

Dans un problème d'étiquetage de données séquentielles, comme la reconnaissance des EN, une première approche consiste à attribuer à chaque mot de la séquence, indépendamment des autres, une étiquette de l'ensemble Y. Ce type de traitement suppose que toutes les variables de sortie sont indépendantes (étiquettes). Ainsi, les EN de deux mots voisins sont indépendantes. Cela peut engendrer quelques lacunes comme par exemple : si on attribue au mot $Paris$ l'étiquette $location$, dans le cas où il est suivi du mot $match$ (paris match), l'étiquette sera $organisation$. Pour pallier à cet inconvénient, les *Modèles de Markov Cachés* (MMC) considèrent les variables de sortie comme une chaîne linéaire. En effet, un MMC modélise une séquence d'observations $X = \{x_t\}_{t=1}^{T}$ en supposant qu'il existe une séquence d'états $Y = \{y_t\}_{t=1}^{T}$ formée à partir des états finis des variables de sortie. Pour calculer la probabilité jointe $P(x, y)$, un MMC considère deux indépendances : (a) Chaque état y_t dépend uniquement de son état prédécesseur y_{t-1}, il est indépendant de tous les états $y_1, y_2 \ldots y_{t-2}$ (certaines approches considèrent un ensemble fini de prédécesseurs), (b) chaque observation x_t dépend uniquement de son état courant y_t. De cette manière, la probabilité jointe de la séquence d'états Y et de la séquence d'observation X est factorisée comme suit :

$$P(x, y) = p(y_1)p(x_1|y_1) \prod_{t=2}^{T} p(y_t \mid y_{t-1})p(x_t \mid y_t), \quad [2]$$

avec comme distribution de transition d'états $p(y_t \mid y_{t-1})$, $p(x_t \mid y_t)$ est la distribution d'observation, $p(y_1)$ est la distribution de l'état initial.

Les MMC sont des modèles génératifs, qui assignent une probabilité jointe à la paire de séquences (observation, variable de sortie). Pour définir la probabilité jointe, les modèles génératifs énumèrent toutes les séquences d'observation possibles. Pour cela, elles doivent construire des modèles qui contiennent beaucoup de dépendances caractéristiques[2] entre les variables d'observation. Or, le problème d'inférence dans ce type de modèles difficilement traitable (vu le nombre de dépendances dans les problèmes de TALN, les approches existantes appliquent des approximations). D'où l'apparition des modèles discriminatifs.

2.2. *Les Champs Markoviens Aléatoires*

Dans un Champ Markovien Aléatoire (CMA), le système de voisinage est déterminé uniquement par un ensemble d'arêtes dans le graphe.

Définition 2.2.1 *Soit $G = (V, E)$ un graphe, avec V l'ensemble des sommets, et E l'ensemble des arêtes. Le système de voisinage dans G est déterminé uniquement par un ensemble d'arêtes de E ; le voisinage $N(v)$ du noeud $v \in V$ est définit par $N(v) = \{u \in V | < v, u > \in E\}$. Soit $X_V = \langle X_v | v \in V \rangle$ un vecteur aléatoire avec X_v est une variable aléatoire associée au sommet v. La distribution de probabilité strictement positive $P(x_v) = P(X_v = x_v)$ est dite champ aléatoire.*

Une distribution de probabilités P dans un graphe G est dite Champ Markovien Aléatoire si pour toute configuration x_v et pour tout sommet $v \in V$, on a :

$$P(x_v | x_{V-v}) = P(x_v | x_{N(v)}). \qquad [3]$$

La probabilité conditionnelle de x_v connaissant toutes les autres probabilités des sommets du graphe (x_{V-v}) n'est rien d'autre que la probabilité du même sommet connaissant les probabilités de ses voisins $x_{N(v)}$. Donc la probabilité d'un état v ne dépend que des probabilité de son voisinage $N(v)$. Cette méthode de calcul qui ne se base que sur le voisinage peut s'avérer intéressante pour beaucoup de problèmes. Cependant, elle présente quelques lacunes dans les tâches séquentielles. L'exemple de la figure 1, présenté dans (McCallum *et al.,* 2003b), illuste un modèle à états finis conçu pour différencier les deux séquences rib et rob. En supposant que la séquence d'observation est rib, la première observation r coïncide avec deux états (1 et 4). Ainsi les probabilités sont identiques pour les deux transitions ($P = 0.50$). Etant donné que les transitions sont conditionnées par les observations, les états n'ayant qu'un seul état suivant ignorent l'observation (les états 1 et 4). Ainsi les deux séquences ri et ro seront équivalentes, indépendament de la séquence observée. De même pour rib et rob. D'autant si le mot rob est plus fréquent, il sera favorisé par rapport à rib, et donc ce dernier ne sera jamais reconnu.

2. Par exemple, la dépendance entre la capitalisation des mots et leurs suffixes

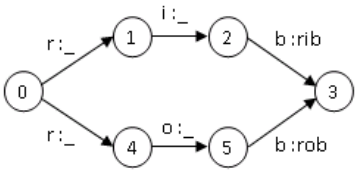


Figure 1. *Graphe de séquences illustrant l'effet de biais : les séquences RIB et ROB sont équivalentes car elles sont conditionnées seulement par les observations.*

Nous allons exposer dans la section suivante l'approche des CAC qui prend en compte cet effet de biais.

2.3. *Les Champs Aléatoires Conditionnels Séquentiels*

La puissance des modèles graphiques réside dans leur capacité à modéliser plusieurs variables indépendantes. L'étiquetage textuel est un cas particulier des modèles graphiques où les variables d'entrées sont présentées sous une forme séquentielle (séquence de mots). Pour cette raison, nous allons étudier un cas particulier des CAC qui est le CAC-Séquentiel (*Linear-Chain CRFs*). Les Champs Aléatoires Conditionnels Séquentiels (*Linear-Chain Conditional Random Fields*) (Lafferty *et al.,* 2001) est un cadre probabiliste discriminant utilisé pour la segmentation et l'étiquetage des données séquentielles (une segmentation peut être vue comme un étiquetage). L'avantage que présente CAC par rapport aux modèles markoviens classiques est la prise en compte du problème du biais des étiquettes (l'exemple illustré dans la figure 1). En effet, les transitions des états ne dépendent pas seulement que des états mis en cause dans la transition (les états voisins), mais aussi des états du modèle global(McCallum *et al.,* 2003a). D'autre part, CAC peut prendre plusieurs paramètres en entrée (autres caractéristiques des éléments d'entrée, comme la position syntaxique des mots dans un cadre textuel), ce qui permet l'utilisation de plusieurs niveaux hiérarchiques d'étiquetage. Cette propriété nous permet d'utiliser des information annexes afin de mieux décrire les données d'entrée. La probabilité conditionnelle $P(Y|X)$ est exprimée sous une forme exponentielle en CAC, comme le montre la formule suivante :

$$P(Y|X) = \frac{1}{Z_X} exp(\sum_{i=1}^{n} \sum_{k=1}^{K} \lambda_k f_k(y_i, x_i, v(x_i), X)), \qquad [4]$$

où $\{f_k(y_i, x_i, v(x_i), X)\}_{k=1}^{K}$ est un ensemble de fonctions aléatoires, appelées *fonctions caractéristiques*. Elles sont généralement des fonctions binaires de la présence des caractéristiques en fonction de l'étiquette candidate y_i, du mot (x_i), de son voisinage $v(x_i)$, et de sa séquence (X). Dans ce cas, le voisinage $v(x_i)$ ne se limite pas seulement aux mots voisins mais il peut inclure toutes les caractéristiques annexes

des mots. λ_k est un paramètre de pondération associé à chaque fonction caractéristique f_k. Z_X est un facteur de normalisation, c'est la somme de toutes les séquences candidates :

$$Z_X = \sum_{l \in L(X)} exp(\sum_{i=1}^{l} \sum_{k=1}^{K} \lambda_k f_k(y_i, x_i, v(x_i), X)), \quad [5]$$

avec $L(X)$ l'ensemble de toutes les sollutions possibles de X. Le problème d'estimation de paramètres (phase d'apprentissage) est de déterminer le vecteur $\theta = \langle \lambda_1, \lambda_2 \ldots \lambda_s \rangle$ à partir des données d'apprentissage $D = \{(x^i, y^i)\}_{i=1}^{N}$ avec une distribution empirique[3]. Dans (Lafferty *et al.,* 2001), l'auteur présente un algorithme qui maximise la fonction objective de log-vraisemblance $\ell(\theta)$. Le but étant de trouver le meilleur vecteur ℓ qui caractérise le mieux les données d'apprentissage :

$$\ell(\theta) = \sum_{i=1}^{N} log P_\theta(y^i | x^i). \quad [6]$$

Dans le cas où la probabilité conditionnelle porte sur une séquence d'éléments (comme c'est le cas dans les problèmes du TALN), l'algorithme de $Viterbi$ peut être appliqué pour maximiser cette fonction, comme c'est le cas dans l'implémentation que nous utilisons.

3. Etiquetage par le contexte

3.1. *Le corpus ESTER*

Nous avons utilisé dans nos expérimentations le corpus de données annoté de la compagne ESTER (Sylvain Galliano *et al.,* n.d.). La campagne ESTER[4] a pour but de mesurer les performances des systèmes de transcription, recherche d'information, et de compréhension d'émissions radiophoniques francophones. Les transcriptions du corpus ESTER sont enrichies par un ensemble d'informations annexes, comme le découpage automatique en tours de paroles, le marquage des EN. L'évaluation de la qualité de ces informations annexes est une tâche importante qui va permettre la mesure des performances d'un système d'indexation complet. Les données, composées de journaux et d'émissions radiophoniques, sont segmentées en sections, chaque section

3. La densité empirique d'une variable à valeurs discrètes est simplement constituée de la proportion des observations prenant chaque valeur.

4. campagne d'Évaluation des Systèmes de Transcription Enrichie d'Émissions Radiophoniques

est dédiée à une thématique définie, qui implique des intervenants et des invités. Le corpus fourni se compose de 90 heures de radio en français, transcrites et annotées manuellement. Ce corpus est divisé en trois parties : la première, sert à l'apprentissage des modèles (*appr.*) ; la seconde, ensemble de développement (*dev.*), sert à l'ajustement des paramètres des modèles ; la troisième est dédiée aux tests et à l'évaluation des performances (*test*). La table 1 illustre la répartition des sources de données de la compagne ESTER.

Source	appr.	dev.	test
France Inter	32h	2h	1h
RFI-FM	21h	1h	2h
France Info	8h	1h	1h
RTM	21h	1h	1h
Total	82h	5h	5h

date	appr.	dev.	test
1998	18h	1h	1h
1999	4h	1h	-
2000	13h	1h	1h
2003	47h	2h	3h
Total	82h	5h	5h

Tableau 1. *distribution des données par sources et par dates.*

Nous Nous somme intéressé à la tâche d'extraction de l'information dans la compagne d'ESTER (Sylvain Galliano *et al.,* n.d.). Cette tâche consiste en l'annotation des EN à paritr des transcriptions manuelles des émissions radiophoniques, une tâche indispensable pour l'analyse sémantique des données.

3.2. *Le contexte hierarchique des entités*

L'étiquetage utilisé pour l'identification des EN est une description à plusieurs niveaux. En effet, chaque entité y est représentée par un concept y_1 ou plusieurs concepts $y_1, y_2, \ldots y_k$. Ainsi, nous avons $y = y_1.y_2 \ldots y_k$ où chaque concept y_i est subsumé par le concept y_{i-1} pour $i \in \{2, 3, \ldots k\}$ et le concept y_1 est subsumé par le concept le plus général dans notre représentation $Entity$. En conséquence, chaque concept est un noeud dans la hièrarchie des concepts, et chaque EN est représentée par un chemin dans la structure (Figure 2). Dans les annotations d'ESTER, le nombre maximal de niveaux est de 3. De ce fait, une étiquette est de la forme $y = y_1.y_2.y_3$ avec $y_1 \in Niveau(N_1)$, $y_2 \in Niveau(N_2)$, et $y_3 \in Niveau(N_3)$. Par exemple pour *Michael*, on associe l'étiquette *pers.hum*, où *pers* correspond au concept le plus général qui est personne ($pers \in Niveau(N_1)$), *hum* est le concept le plus spécifique qui est humain ($hum \in Niveau(N_2)$). Cette forme d'identification, détaillée, rend la tâche de reconnaissance plus difficile et complexe. Un système de reconnaissance de qualité acceptable peut reconnaître les concepts généraux, mais il y a moins de chances qu'il reconnaisse toute la chaîne. La simple application des CAC est de considérer chaque étiquette y comme une chaîne de concaténation des trois niveaux. L'inconvénient avec cette approche est qu'elle considère toutes les étiquettes indépendantes. En conséquence, le nombre d'étiquettes est plus important (elle contient toutes les combinaisons possibles des trois niveaux) et nécessite un grand nombre de données d'apprentissage pour construire un modèle qui caractérise le mieux toutes les

étiquettes. Notre approche consiste à construire un modèle de prédiction pour chaque niveau de conceptualisation. Chaque modèle M_j avec $j \in \{N_1, N_2, N_3\}$ est identifié dans un domaine non ambigu $D_j = \left\{\left(x^i, y_j^i\right)\right\}_{i=1}^{N}$ où N est l'ensemble des mots d'apprentissage. Chaque modèle M_k va nous fournir la prédiction y_j du mot x pour le niveau de conceptualisation j. La prédiction finale du mot x est alors la concaténation des trois prédictions : $y = y_1.y_2.y_3$. En utilisant la structure hiérarchique des concepts définie précédemment, nous pouvons vérifier la validité de la prédiction y. En effet, si y ne représente pas un chemin valide[5] dans la structure, des approximations peuvent êtres effectuées afin de valider cette prédiction. Pour illustrer ce cas de figure, donnons l'exemple suivant.

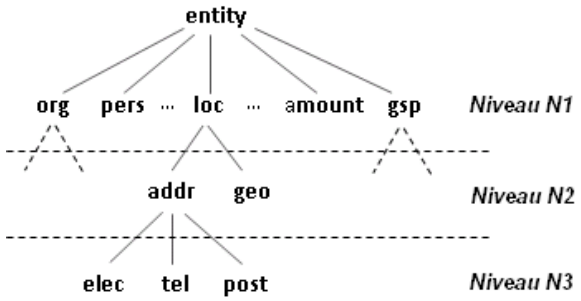


Figure 2. *Les niveaux hiérarchiques des EN*

3.2.1. *Exemple : approximation hiérarchique*

La figure 3 illustre l'application d'un étiquetage par niveaux pour une phrase qui se compose de 8 mots $\langle m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\rangle$. L'étiquetage en niveau N_1 nous donne 3 labels N_1^1, N_1^2 et N_1^3 (figure 3.1). La figure 3.2 montre l'étiquetage en niveau N_2, et la figure 3.3 l'étiquetage en N_3. L'étiquetage final (complet) est la concaténation des étiquettes des 3 niveaux (figure 3.4). Dans ce dernier, on remarque que le mot m_4 est étiqueté en N_2 et N_3 mais non étiqueté en niveau N_1. Dans ce cas une amélioration peut être apportée par approximation de concepts en attribuant au mot m_4 l'étiquette N_1^X associée à la branche $N_1^X.N_2^3.N_3^2$ dans l'arbre des concepts. Notre approche consiste alors à construire un modèle pour chaque niveau indépendamment des autres. Cela implique une diminution considérable de la complexité ainsi qu'une augmentation des fréquences d'apparition des étiquettes.

3.3. *Les modèles enrichies*

La phase d'apprentissage des CAC permet l'utilisation de plusieurs informations pour caractériser le voisinage du mot ($v(i)$). En utilisant cette propriété, on peut ap-

5. Un chemin valide dans l'arbre est un chemin qui a comme début la racine de l'arbre, et une feuille comme fin.

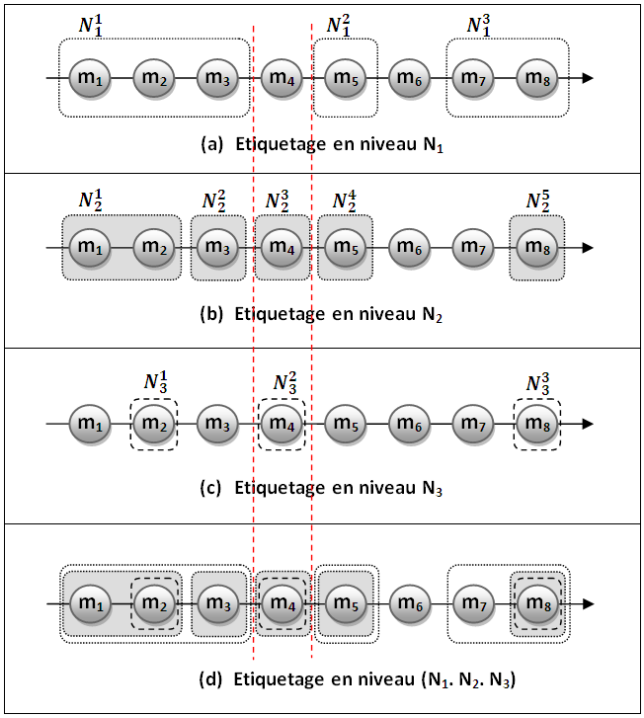


Figure 3. *Exemple d'utilisation de la hiérarchie pour l'amélioration de la prédiction*

pliquer un apprentissage par combinaison de niveaux. Ainsi, nous allons construire un modèle pour chaque niveau sachant les prédictions des autres niveaux. Le principe consiste à construire le modèle simple M_j puis les modèles combinés M_j^{comb} pour chaque niveau (étape 2 dans la figure 4). Le modèle combiné d'un niveau j utilise les prédictions fournies par les modèles simples $s/s \neq j$ comme données d'entrée. Dans la phase de test, on utilise les modèles simples pour génèrer les prédictions associées à chaque niveau (étape 3 dans la figure 4). Ces prédictions seront ainsi utilisées comme des données d'entrées pour les modèles combinés (étape 4 dans la figure 4). Par exemple, pour avoir la prédiction du 3ème niveau, on utilise comme connaissance a priori les prédictions des niveaux 1 et 2 générés par les modèles M_1 et M_2. Cette démarche nous permet de raffiner les prédictions pour chaque niveau, car on dispose de plus d'information a priori.

3.4. *Le contexte syntaxique*

Les CAC construisent des modèles qui caractérisent les données fournies en entrée. Chaque élément de ces données peut être définie par une ou plusieurs caractéristiques

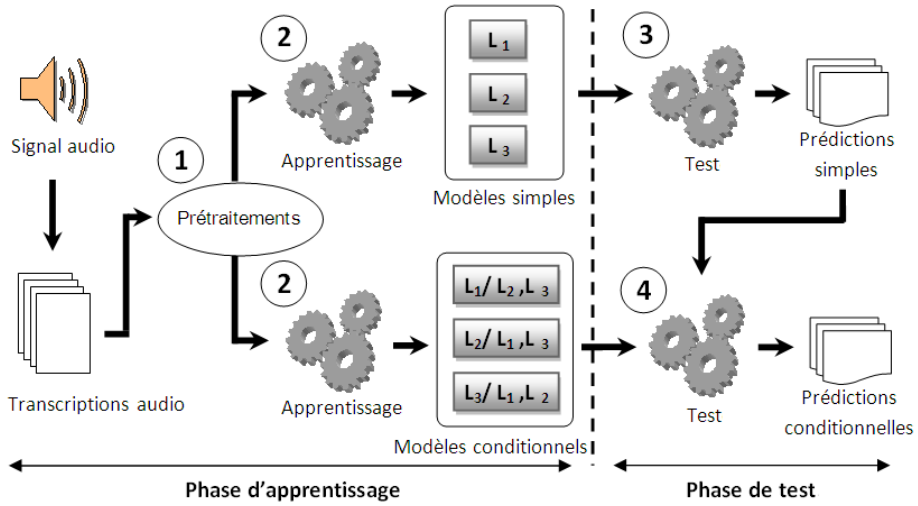


Figure 4. *Processus d'apprentissage conditionnel*

(appelées aussi attributs). Ces attributs peuvent ainsi être utilisés pour améliorer la qualité de prédiction des modèles(Sutton *et al.,* 2005). Dans le processus de REN, on peut identifier un mot simplement par sa représentation textuelle. Mais on peut aussi inclure d'autres informations annexes. Dans notre méthode nous avons inclus les étiquettes syntaxiques des mots[6]. En effet, on construit des modèles qui prennent en compte le contexte syntaxique dans la prédiction des mots. Pour effectuer les annotations syntaxiques des mots, nous avons utilisé le logiciel TreeTagger (Schmid, 1994). Ses performances atteignent une précision de 96% sur le corpus de test proposé par ses concepteurs.

4. Expérimentations et résultats

Pour tester les performances de notre approche, nous avons utilisé le corpus annoté d'ESTER. Etant dans un cadre de parole spontanée, la notion de segmentation en phrase est inexistante. En conséquence, nous avons utilisé une segmentation en tours de parole entre les intervenants. Nous avons utilisé le logiciel CRF++[7] avec une fenêtre de voisinage de 10 mots. Nous avons utilisé comme mesure de performance le *rappel*, la *précision*, et la *F(1)-mesure*. Nous avons fait en premier lieu une comparaison entre les modèles simples M_j et les modèles combinés M_j^{comb} pour chaque niveau $j \in \{1, 2, 3\}$ (figure 5). On remarque une amélioration de la F(1)-mesure de

6. En anglais : Part Of Speech tagging (POS tagging)
7. http ://crfpp.sourceforge.net/

1 à 3% dans les modèles combinés. A cet effet, nous avons considéré les prédictions combinées dans le résultat final.

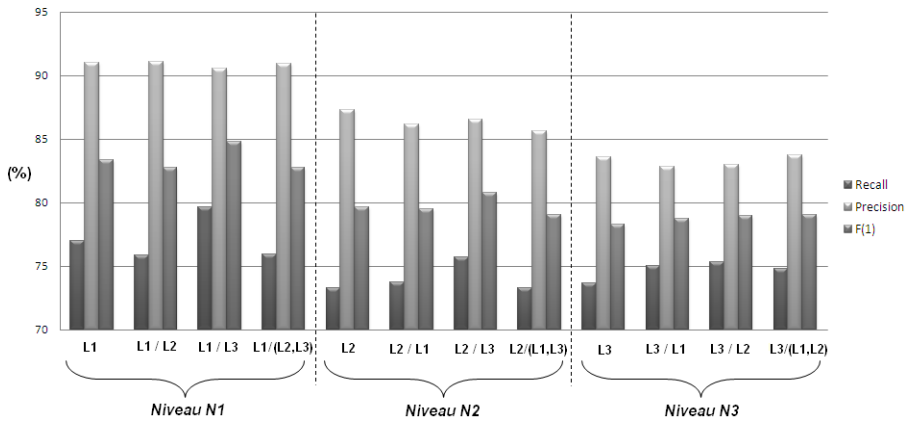


Figure 5. *Les résultats des CAC simples et combinés pour chaque niveau de conceptualisation*

Nous avons comparé les prédictions des différentes méthodes graphiques (MMC, CAC Séquentiels) avec les prédictions de notre approche (CAC Combinés). Les résultats illustratifs sont présentés dans le tableau 2. On remarque que les CAC nous donnent une meilleure précision par rapport au modèle markovien. Les CAC combinés apportent une fine amélioration la F(1)-mesure de 1% par rapport aux CAC séquentiels, dû aux approximations de concepts. L'intégration du contexte syntaxique améliore significativement le rappel (il passe de 61% à 74%). Ces résultats montrent que l'utilisation du contexte dans le processus de REN donne une meilleure qualité et un meilleur taux de prédiction. Par ailleurs, l'utilisation de l'apprentissage par niveaux devise par 5 le temps nécessaire pour la construction des modèles de prédiction (phase d'apprentissage). Ceci est dû principalement à la diminution de la complexité du corpus : la complexité passe de $O\left((N_1)^2 \times (N_2)^2 \times (N_3)^2\right)$ à $O\,(N_1)^2 + O\,(N_2)^2 + O\,(N_1)^2$ avec (N_j) la taille du domaine de sortie du niveau j.

Mesures	Rappel	Précision	F(1)-Mesure	Temps CPU
MMC	$57,9$	$69,8$	**63,0**	239h
CAC Classiques	$61,0$	$85,8$	**71,3**	274h
CAC Combinés	$61,4$	$86,5$	**72,2**	58h
CAC Classiques + Syntaxe	$74,0$	$83,8$	**78,5**	290h
CAC Combinés + Syntaxe	$75,9$	$84,2$	**79,9**	59h

Tableau 2. *Résultats des prédictions, les temps sont donnés pour l'apprentissage des 82h de temps de parole.*

5. Conclusion

La REN est une tâche centrale dans la chaîne des traitements sémantiques des transcriptions radiophoniques. Nous avons exposé les performances des méthodes graphiques dans la résolution du problème de classification, et plus particulièrement le problème d'étiquetage des données séquentielles. L'utilisation des structures des étiquettes dans le processus d'apprentissage apporte un gain dans la qualité d'étiquetage, car elle apporte une connaissance a priori des données. Nous avons montré que dans le domaine de l'indexation sémantique, la représentation conceptuelle surpasse la représentation linguistique des entités. En effet, l'utilisation de la hiérarchie des concepts améliore la classification et permet l'utilisation d'heuristiques d'approximation de concepts. Cette approche diminue considérablement le temps d'apprentissage. L'utilisation du contexte syntaxique des transcriptions apporte une connaissance supplémentaire sur données et améliore nettement le rappel. Nous allons, dans nos travaux futurs, combiner les connaissances extraites de plusieurs corpus et faire les correspondances entre les hiérarchies de concepts.

6. Bibliographie

Lafferty J. D., McCallum A., Pereira F. C. N., « Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data », *ICML '01 : Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 282-289, 2001.

McCallum A., Freitag D., Pereira F. C. N., « Maximum Entropy Markov Models for Information Extraction and Segmentation », *ICML '00 : Proceedings of the Seventeenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 591-598, 2000.

McCallum A., Li W., « Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons », *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, Association for Computational Linguistics, Morristown, NJ, USA, p. 188-191, 2003a.

McCallum A., Rohanimanesh K., Sutton C., « Dynamic Conditional Random Fields for Jointly Labeling Multiple Sequences », *Workshop on Syntax, Semantics, Statistics*, 2003b.

Schmid H., « Part-of-speech tagging using decision trees », *Proceedings of International Conference on New Methods in Language Processing*, 1994.

Sutton C., McCallum A., « Joint Parsing and Semantic Role Labeling », *In Conference on Natural Language Learning (CoNLL)*, 2005.

Sutton C., Mccallum A., « An Introduction to Conditional Random Fields for Relational Learning », *in* , L. Getoor, , B. Taskar (eds), *Introduction to Statistical Relational Learning*, MIT Press, 2006.

Sylvain Galliano Edouard Geoffrois J.-F. B. G. G. D. M., Choukri K., « Corpus description of the ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News », *Language Resources and Evaluation Conference (LREC 06)*, n.d.

Introduction de la sémantique d'un document sous le modèle de langage

Arezki Hammache* — Mohand Boughanem — Rachid Ahmed-Ouamer***

* *Laboratoire LARI, Université Mouloud Mammeri*
15000 Tizi-Ouzou, Algérie
{arezki20002002, ahm_r}@yahoo.fr

** *Laboratoire IRIT, Université Paul Sabatier*
118route de Narbonne 31062 Toulouse Cedex 09, France
bougha@irit.fr

RÉSUMÉ. *La plupart des systèmes de recherche d'information classiques se basent sur une indexation par termes simples. Cependant, ces derniers délivrent beaucoup de résultats en réponse aux requêtes des utilisateurs. Ceci est du en partie au fait que le contenu sémantique d'un document (ou d'une requête) ne peut pas être capturé précisément par un simple ensemble de mots clés indépendants. Deux directions sont explorées pour incorporer la sémantique dans les modèles de langage. La première se base sur l'exploitation des liens entre termes tout en utilisant une même unité d'indexation. La seconde se base sur l'utilisation d'unités d'indexation plus complexes en plus de l'utilisation de termes simples. Dans ce papier est détaillée l'approche que nous proposons pour incorporer la dimension sémantique de document, et qui rentre dans le cadre de la seconde direction.*

ABSTRACT. *Most traditional information retrieval systems are based on simple terms indexing. However, they deliver massive results in response to users queries. This is partly due to the fact that semantic content of a document (or a request) can not be accurately captured by a simple set of independent keywords. Two directions are investigated to incorporate semantics in the language models. The first is based on the exploitation of terms dependency while using the same indexing unit. The second is based on the use of more complex indexing units. In this paper we detail our approach to incorporate the semantic dimension of document.*

MOTS-CLÉS : *Recherche d'information, modèles de langage, indexation sémantique.*

KEYWORDS: *Information retrieval, language model, semantic indexing.*

1. Introduction

La plupart des systèmes de recherche d'information (SRI) actuels privilégient la minimisation de temps de réponses par rapport à la qualité des documents retournés à l'utilisateur. En effet, ces derniers délivrent de grandes quantités de documents en réponse aux requêtes des utilisateurs, ce qui génère ainsi, une surcharge informationnelle dans laquelle il est difficile de distinguer l'information pertinente de l'information secondaire ou même du bruit. L'une des raisons qui a engendré ceci est la non prise en compte de toutes les caractéristiques d'un document dans le processus d'indexation et de recherche. En effet, les SRI implémentent les techniques traditionnelles de la RI, qui considèrent un document comme un ensemble de termes (sac de mots). Cependant, l'une des critiques formulées à l'utilisation des termes simples comme unité d'indexation est que le contenu d'un document ne peut pas être capturé précisément par un simple ensemble de mots clés indépendants.

Le modèle de langage offre un cadre probabilistique pour la description du processus de la RI. Parmi les propriétés qu'offre ce modèle est la combinaison des différentes représentations d'un document comme l'intégration des informations sémantiques d'un document. Deux alternatives ont été explorées pour intégrer le contenu sémantique d'un document dans ces modèles. La première se base sur l'exploitation des liens entre termes tout en utilisant une même unité d'indexation. La seconde se base sur le développement de modèles pour une représentation plus détaillée du contenu des documents et des requêtes, et cela par l'utilisation d'unités d'indexation plus complexes en plus de l'utilisation des termes simples. Notre approche s'inscrit dans cette deuxième orientation.

L'utilisation d'unités d'indexation complexes en RI nécessite la prise en compte de plusieurs paramètres à savoir : la technique d'extraction des termes composés utilisée, la prise en compte de l'adjacence et de la directionnalité des termes composants, la pondération des termes composés et l'intégration des termes composés dans le modèle de Ranking. L'apport de notre approche concerne les deux derniers paramètres.

Nous organisons ce papier comme suit : Dans la section 2 sont abordées la modélisation de langue en recherche d'information et l'intégration de la composante sémantique dans ces modèles. La section 3 est consacrée à la présentation de l'approche que nous proposons pour intégrer les informations sémantiques dans les modèles de langage. Un exemple d'illustration de l'approche proposée est donné en section 4. La dernière section fait la synthèse de cette étude.

2. Modélisation de langue et incorporation de la sémantique

2.1. *La modélisation de langue en recherche d'information*

L'approche de modélisation de langue part d'un principe différent de ceux des approches traditionnelles ; on ne tente pas de modéliser directement la notion de pertinence (à l'exception de (Lavrenko *et al.,* 2001)) ; mais on considère que la pertinence d'un document face à une requête est en rapport avec la probabilité que la requête puisse être générée par un modèle de langue d'un document. Ainsi un modèle de langue (Md) est construit pour chaque document, et le score d'un document est déterminé par la probabilité de génération de la requête sachant le modèle de ce document, P (Q | Md). Cette approche (ML) permet de combiner les deux composantes (indexation et Ranking) dans un seul modèle unifié.

La plupart des modèles de langue développés pour la RI utilisent le principe de génération de la requête par un modèle de document. Les approches de modélisation de langage pour la RI peuvent être classées en trois catégories :

– Génération de la requête par le modèle de document (Ponte *et al.,* 1998) (Hiemstra, 1998).

– Génération de document à partir du modèle de la requête : cette approche procède dans le sens inverse que la première. Ainsi, un modèle de langage de la requête est construit, ensuite les documents sont classés selon leurs probabilités que leur contenu soit généré par le modèle de la requête. Le travail de Lavrenko et Croft (Lavrenko *et al.,* 2001) s'inscrit dans cette catégorie d'approche .

– Similarité entre modèle de document et modèle de la requête : Dans cette approche, un modèle de langage est construit pour chaque document et un autre pour la requête. Un score de similarité est calculé entre ces deux modèles (Lafferty *et al.,* 2001).

Deux problèmes fondamentaux liés à l'utilisation des modèles de langage en RI sont la source de plusieurs études. Le premier problème concerne la clairsemence de données (Data Sparseness) : qui consiste à attribuer la probabilité zéro à tout document ne contenant pas tous les termes, même s'il est pertinent. Pour y remédier la technique de lissage est utilisée, et permet d'attribuer une probabilité non nulle pour les termes non observés dans le document. Cette technique (Smoothing) peut jouer divers rôles, par exemple la combinaison de multiples sources d'information sur un document (informations sémantiques). Le second problème est lié à l'utilisation de modèle de langage uni-gramme qui ne prend pas en compte la dépendance entre les termes (document, requête).

2.2. *Incorporation de la sémantique dans les modèles de langage*

Deux directions ont été explorées pour incorporer la sémantique dans les modèles de langage : la première se base sur l'exploitation des liens entre termes

(requête, document) tout en utilisant une même unité d'indexation (termes simples). La seconde se base sur le développement de modèles pour une représentation plus détaillée du contenu des documents et des requêtes, et cela par l'utilisation d'unités d'indexation plus complexes en plus de l'utilisation des termes simples.

2.2.1. *Utilisation des termes simples comme unité d'indexation*

Plusieurs travaux ont été réalisés pour incorporer les relations existantes entre unités d'indexation, selon deux approches. La première consiste à incorporer les relations entre les termes des documents ou de la requête (Gao *et al.,* 2004), les relations entre termes expriment les dépendances. Dans la seconde approche l'incorporation des relations entre les termes de la requête et les termes des documents est réalisée soit : en modifiant la représentation (modèle) de la requête en ajoutant les termes liés, cette opération est nommée expansion du modèle de la requête (Bai *et al.,* 2005) (Lafferty *et al.,* 2001) (Wei *et al.,* 2007) ; soit en modifiant la représentation de document, en donnant une probabilité plus grande aux termes liés aux termes de document que pour les autres termes. Cette opération est nommée expansion de modèle de document (Berger *et al.,* 1999) (Cao *et al.,* 2005) (Cao *et al.,* 2007) (Liu *et al.,* 2004) (Tao Tao *et al.,* 2006) (Wei *et al.,* 2006).

2.2.2. *Utilisation de termes composés et de termes simples comme unité d'indexation*

Pour incorporer la sémantique dans les modèles de langage, une autre piste peu explorée consiste à utiliser des unités d'indexation plus complexes en plus de termes simples ; ces unités d'indexation peuvent être vues comme des concepts, plusieurs vocables sont utilisés (phrase, N-gramme, collocation, termes composés) ; ce qui améliore la représentation des documents et des requêtes. L'intérêt d'utiliser des termes composés comme unité d'indexation est que les termes composés sont moins ambigus et plus précis que les termes simples. Par exemple le terme « Java » est ambigu, par contre les termes composés « Ile de java » et « Langage java » sont non ambigus. Et le terme « Voiture Electrique » est plus spécifique que les termes « Voiture » et « Electrique » pris isolement.

Les termes composés permettent de construire des unités d'indexation non ambiguës et plus précises et peuvent par conséquent améliorer la précision de la RI. Miller et al (Miller *et al.,* 1999) ont proposé d'intégrer les bi-grammes dans leur modèle initial. Song et Croft (Song *et al.,* 1999) ont proposé un modèle de langage qui combine le modèle bi-gramme et le modèle uni-gramme en utilisant l'interpolation linéaire. Srikanth et Srihari (Srikanth *et al.,* 2002) ont développé un modèle dit bi-terme (modèle bi-gramme dans l'ordre est ignoré). Jiang et al (Jiang *et al.,* 2004) ont proposé un modèle de langage pour incorporer des phrases (deux termes) en utilisant la méthode de lissage Backoff. Alvarez et al (Alvarez *et al.,* 2004) ont proposé l'incorporation des termes composés sans contraintes d'adjacence ou d'ordre.

Les techniques qui permettent l'identification des termes composées sont scindées en trois catégories : linguistiques, statistiques et mixtes. Les techniques statistiques auxquelles nous nous intéressons sont basées sur des informations tirées de corpus d'où leur flexibilité et leur portabilité (ie : elles ne dépendent ni de la langue du corpus ni du domaine traité par le corpus). Dans les techniques statistiques les termes composés sont extraits en se basant soit sur leurs fréquences observées dans le corpus soit en utilisant des mesures d'association qui déterminent le degré d'association entre les termes composants. Les mesures d'association permettent de calculer « un score d'association » pour chaque paire de termes candidat dans le corpus ; ce score indique le potentiel de ce candidat d'être reconnu comme un terme composé. Plusieurs mesures d'association ont été proposés dans la littérature telles que : l'Information Mutuelle, le coefficient de Dice, X2 score, etc. (Liu *et al.*, 2004).

3. Approche proposée

Notre objectif est de représenter au mieux le contenu sémantique des documents. Nous réalisons cela sous le cadre des modèles de langage. Comme nous l'avons vu auparavant, deux alternatives ont été explorées pour intégrer les informations sémantiques dans les modèles de langage.

La première consiste à exploiter les liens entre les termes des documents, requêtes (requête → document). Dans le cas de l'expansion de la requête l'exploitation des liens entre termes est réalisée au moment de la recherche, ce qui affecte négativement les temps de réponses, de plus l'efficacité d'un système de recherche d'information dépend fortement du nombre de termes de la requête. Dans le cas de l'expansion du modèle du document des inconvénients surgissent selon l'approche adoptée.

L'autre alternative qui consiste à intégrer la sémantique dans les modèles de langage est l'utilisation d'unités d'indexation plus complexes à côté d'unités d'indexation simples. Notre approche s'inscrit dans cette optique. Six paramètres sont généralement à considérer dans l'utilisation des termes composés comme unité d'indexation à coté de l'utilisation des termes simples à savoir : la technique d'extraction des termes composés utilisée, l'adjacence et la directionnalité des termes composants, la taille des termes composés, la pondération des termes composés et la manière d'intégrer ces termes dans le modèle de Ranking.

Nous présentons ci-dessous notre approche pour incorporer les informations sémantiques dans le modèle de langage en définissant les caractéristiques mises en jeu dans l'indexation par des termes composés. Plus particulièrement on décrit en détail la formule de pondération des termes composés proposée et l'intégration des termes composés dans le modèle de langage. Pour les autres caractéristiques : (identification des termes composés, directionnalité, taille des termes composés et distance entre termes composants) nous adoptons des solutions existantes.

– *Identification des termes composés.* Nous optons pour l'approche statistique pour l'extraction des termes composés du fait que nous voulons élaborer une approche plus générale qui ne dépend pas de la langue ou du domaine du corpus. Pour l'identification des termes composés nous utilisons la mesure d'Information Mutuelle (PMI) basée sur l'étude menée par Petrovic et al (Sasa *et al.*, 2006) qui ont montré que cette mesure donne de meilleurs résultats que ceux obtenus par les autres mesures telles que le coefficient de Dice ou la mesure Chi-square.

– *Directionnalité.* Srikanth et Srihari (Srikanth *et al.*, 2002) ont montré que la prise en compte de la directionnalité (cas de deux termes) est plus précise pour la RI. En se basant sur ce constat nous adoptons la directionnalité des termes composés dans notre approche.

– *Taille des termes composés.* En principe les termes composés peuvent être de n'importe qu'elle longueur (supérieure ou égale à 2). Dans notre cas nous limitons la taille des termes composés à deux qui est une pratique commune, et scalable pour de grandes collections hétérogènes.

– *Distance entre termes composants.* La cooccurrence des termes est une source d'information importante et efficace pour la désambiguïsation des termes (Agirre *et al.*, 2001). Nous traitons cette relation de cooccurrence en terme d'extraction des termes composés dont les termes composants sont adjacents, exemple : « Génie Logiciel », « Microsoft Word » seront utilisées comme unités d'indexation car les termes composés sont moins ambiguës et plus précis que les termes qui les composent, de ce fait le contenu sémantique des documents et des requêtes est plus précis. Un deuxième type de relation de cooccurrence consiste en l'extraction des termes de voisinage d'un terme donné. Exemple : docteur (hôpital, infirmière, aide soignant) peut être utilisé dans une étape ultérieure (expansion de la requête).

– *Pondération des termes composés.* La pondération des termes composés est un problème non résolu en RI. En effet il n'existe pas de schéma bien accepté pour la pondération des termes composés. Des alternatives ont été proposées ; parmi elles l'adaptation de schéma bien connu de pondération de termes simples TF-IDF. Cependant, les schémas de pondération proposés dans (Baziz *et al.*, 2005) et (Liu *et al.*, 2004) ne tiennent pas compte d'un facteur important qui est l'importance des termes composants dans le terme composé ; dans les schémas précédents cette importance est considérée identique. Or, dans la réalité un des termes composants peut être plus important que les autres termes. Exemple : le terme « ordinateur » est plus important que le terme « personnel » dans le terme composé « ordinateur personnel ». Cette dominance de terme est déterminée par la spécificité du terme, cette dominance est généralement supposée qu'elle est en corrélation avec l'IDF du terme. Ainsi, nous proposons d'exprimer l'importance d'un terme composant « t » dans un terme composé « tc » de la manière suivante :

$$imp(t / tc) = \frac{idf(t)}{\sum_{t_i \in t_c} idf(t_i)} \quad [1]$$

En supposant que l'auteur d'un document utilise les termes composants isolément pour exprimer le terme composé comme abréviation après un nombre d'occurrences de terme composé. Par exemple un document contenant le terme composé « énergie électrique », l'auteur utilise le terme « énergie » simplement pour désigner le terme composé « énergie électrique ». Néanmoins, un problème surgi lorsqu'un terme composant est partagé par deux voire plusieurs termes composés, dans ce cas il faut trouver le terme composé auquel renvoie le terme simple. Par exemple : si on a le terme « énergie fossile » dans le document précédent. Alors il faut choisir à quel terme composé le terme « énergie » renvoie. Nous proposons d'utiliser un facteur qui combine l'importance du terme composant dans les termes composés et la fréquence des termes composés dans le document pour désigner le terme composé adéquat. Le terme composé qui maximise ce facteur est choisi.

Termes / Facteurs	Energie électrique	Energie fossile
Importance (énergie)	0.6	0.1
Nombre d'occurrences	20	12
Produit	12	1.2

Table 1. *Exemple*

Dans cet exemple on voit bien que le terme composé « énergie électrique » maximise le produit des deux facteurs : importance du terme « énergie » dans le terme composé et la fréquence du terme composé dans le document, par conséquent le terme « énergie » utilisé isolement dans le document renvoie au terme composé « énergie électrique ». La fréquence d'un terme composé « tc » dans un document dépend du nombre d'occurrences de ce terme dans le document et du nombre d'occurrences des termes composants pour lesquels le terme composé maximise le facteur discuté auparavant. Formellement elle est exprimée ainsi :

$$F(tc) = nbr(tc) + \sum_{i=1}^{2} imp(t_i / t_c) \times nbr(t_i) \quad \text{si}$$

$$imp(t_i / t_c) \times nbr(tc) = \max_{t_i \in t_{c'}} ((imp(t_i / t_{c'}) \times nbr(tc')) \qquad [2]$$

Tel que « tc' » est l'ensemble des termes composés qui contient le terme « ti », *F(tc)* représente la fréquence du terme composé « tc », *nbr(tc)* est le nombre d'occurrences du terme composé « tc », $imp(t_i/t_c)$ est l'importance du terme « ti » dans le terme composé « tc » et *nbr(ti)* est le nombre d'occurrence du terme « ti ». Exemple : supposant que le nombre d'occurrences de terme « énergie » dans l'exemple précédent est 10. Alors la fréquence du terme composé « énergie électrique » est : F ('énergie électrique') = nbr('énergie électrique') + imp('énergie'/ 'énergie électrique') * nbr('énergie') = 20 + 0.6*10 = 26.

– *Intégration des termes composés dans le modèle de langage.* Dans notre approche nous considérons qu'un document « D » de la collection « C » est représenté de deux façons différentes : la première est la représentation par des termes simples notée « Dt », la seconde est la représentation par des termes composés notée « Dtc » ; ici la notion d'un terme composé se réfère à tous les termes de longueur « un » ou « deux », les termes de longueur « un » sont les termes qui ne participent pas a la composition des termes de taille « deux ». Le même raisonnement est appliqué pour la collection, ainsi la collection est représentée par deux représentations : une représentation avec des termes simples « Ct » qui est obtenue par la concaténation des représentations par des termes simples des documents formant la collection, et une représentation avec des termes composés « Ctc » qui est obtenue par la concaténation des représentations par des termes composés des documents formant la collection. A partir de ces deux représentations de document « Dt » et « Dtc » nous proposons deux modèles de document exprimés ainsi :

– Pour la représentation avec des termes simples le modèle obtenu est le modèle uni-gramme, en interpolant le modèle de document avec celui de la collection, exprimé ainsi :

$$P(t / M_{Dt}) = \lambda P_{ML}(t / M_{Dt}) + (1-\lambda) P_{ML}(t / C_t) \qquad [3]$$

Tel que : $P_{ML}(t / M_{Dt}) = \dfrac{tf(t,D)}{|D|}$ et $P_{ML}(t / C_t) = \dfrac{df(t)}{\sum_{t_i \in C_t} df(t_i)}$

Où ? est un facteur d'interpolation compris entre 0 et 1, *tf(t, D)* est la fréquence de terme « t » dans le document D, *df(t)* est le nombre de documents contenant le terme « t » et Ct est le vocabulaire d'indexes (termes simples).
Les documents de la collection vis-à-vis d'une requête « Q » sont alors ordonnés en utilisant la formule suivante :

$$P(Q / M_{Dt}) = \prod_{t_i \in Q} P(t_i / M_{Dt})$$

– Pour la représentation avec des termes composés nous exprimons le modèle ainsi :

$$P(t / M_{Dtc}) = \sum_{t_c} P(t / t_c) \times P_{ML}(t_c / M_{Dtc}) \qquad [4]$$

Comme nous l'avons noté auparavant un terme composant peut être partagé par deux voire plusieurs termes composés, dans ce cas il faut trouver le terme composé auquel renvoie le terme simple. Nous avons proposé un facteur qui permet de déterminer le terme composé concerné. Notons « t_{cmax} » ce terme composé. Ainsi la probabilité suivante devient alors :

$$\sum_{t_c} P\left(t / t_c\right) = imp(t / t_{c\max}) \quad \text{calculée avec la formule [1]}$$

En remplaçant cette probabilité dans la formule [4] celle-ci devient :

$$P(t / M_{Dtc}) = imp(t / t_{c\max}) \times P(t_{c\max} / M_{Dtc}) \quad [5]$$

Et $$P(t_c / M_{Dtc}) = \lambda P_{ML}(t_c / M_{Dtc}) + (1-\lambda) P_{ML}(t_c / C_{tc}) \quad [6]$$

Est obtenue en interpolant le modèle de document avec celui de la collection, *PML (tc/MDc)* est estimée ainsi :

$$PML\left(tc / MDc\right) = \frac{\mathrm{tf}(\mathrm{t_c})}{\sum_{\mathrm{t_{ci}}} \mathrm{tf}(\mathrm{t_{ci}})}$$

Tel que $tf(t_c)$ est la fréquence du terme composé « tc » dans le document D, calculée selon la formule [2]. Et *PML (tc/Ctc)* est calculée ainsi :

$$PML\left(tc / Ctc\right) = \frac{df(t_c)}{\sum_{t_{ci} \in C_{tc}} df(t_{ci})}$$

Où $df(t_c)$ est le nombre de documents contenant le terme composé « tc » et Ctc est le vocabulaire d'indexes (termes composés). Les documents de la collection vis-à-vis d'une requête « Q » sont alors classés en utilisant la formule suivante :

$$P(Q / M_{Dtc}) = \prod_{t_i \in Q} \left(\sum_{t_i \in t_{c\max}; t_{c\max} \in D} imp(t_i / t_{c\max}) \times P_{ML}(t_{c\max} / M_{Dtc}) \right)$$

Les documents sont représentés par deux représentations différentes : avec des termes simples et avec des termes composés. Le modèle de Ranking doit combiner ces deux représentations. Le modèle obtenu est donné ainsi :

$$P(t / M_D) = \alpha P(t / M_{Dt}) + (1-\alpha) P(t / M_{Dtc})$$

$$P(t / M_D) = \alpha [\lambda P_{ML}(t / M_{Dt}) + (1-\lambda) P_{ML}(t / C_t)] + (1-\alpha)$$

$$[\lambda P_{ML}(t / M_{Dtc}) + (1-\lambda) P_{ML}(t / C_{tc})]$$

$$P(t / M_D) = \lambda [\alpha P_{ML}(t / M_{Dt}) + (1-\alpha) P_{ML}(t / M_{Dtc})] +$$

$$(1-\lambda)[\alpha P_{ML}(t / C_t) + (1-\alpha) P_{ML}(t / C_{tc})]$$

4. Exemple d'illustration

Dans l'exemple suivant « Ct » est le vocabulaire d'indexes (termes simples), « Ctc » est le vocabulaire d'indexes (termes composés), « C » est l'ensemble de documents de la collection et Q1, Q2, Q3, Q4 des requêtes.

Ct = {m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14, m15}

Ctc = {m1, m2, m3, m6, m8, m9, m11, m13, m14, m15, m1m2, m1m10, m5m6, m12m4, m7m8, m12m2}
C = {d1, d2, d3, d4}; Q1= m1m2 ; Q2= m5m6 ; Q3= m1 ; Q4= m12m4 ;
Dt1={m1(5), m2(2), m5(6), m6(3), m8(2), m10(1), m15(1)};
Dtc1={m1m2(3.58), m1m10(1), m5m6(4.83) , m8(2), m15(1)}
Dt2={m2(1), m3(4), m4(2), m6(3), m8(1), m12(4)};
Dtc2={m2(1), m3(4), m6(3), m8(1), m12m4 (2.77)};
Dt3={m1(2), m4(3), m9(3), m12(5)}; Dtc3={m1(5), m2(3), m9(3), m12m4 (2.77)}
Dt4={m2(3), m5(6), m6(5), m7(6), m8(2)};
Dtc4={m2(3), m5(6), m6(5), m7m8(5.43)}
Dt5={m1(5), m11(8), m14(3), m15(2)}; Dtc5={m1(5), m11(8), m14(3), m15(2)}
Dt6={m2(3), m8(2), m9(1), m11(7) m12(2), m13(2)};
Dtc6={m8(2), m9(1), m11(7), m13(2) , m2m12(2.208)}

	d1			d2		
	scts	sctc	c(scts,sctc)	scts	sctc	c(scts,sctc)
Q1=m1m2	**0,022299**	**0,21309**	0,11801	0,00240	0,00206	0,00223
Q2=m5m6	**0,029928**	**0,28388**	0,15686	0,00293	0,00222	0,00258
Q3=m1	0,20147	0,16867	0,18506	0,02647	0,02222	0,02434
Q4=m12m4	0,00046	0	0,00023	0,02365	0,18713	0,10539
	d3			d4		
	scts	sctc	c(scts,sctc)	scts	sctc	c(scts,sctc)
Q1=m1m2	**0,038955**	**0,051334**	0,04514	0,00369	0,00314	0,00341
Q2=m5m6	0,00047	0,00025	0,00035	**0,038669**	**0,045998**	0,04234
Q3=m1	0,23235	0,27633	0,25434	0,02647	0,02222	0,02434
Q4=m12m4	0,01912	0,16319	0,09115	0,00046	0	0,00023
	d5			d6		
	scts	sctc	c(scts,sctc)	scts	sctc	c(scts,sctc)
Q1=m1m2	0,00974	0,00722	0,00848	0,00358	0,00074	0,00216
Q2=m5m6	0,00046	0	0,00023	0,00031	0	0,00015
Q3=m1	0,22091	0,21667	0,21879	0,02647	0,02222	0,02434
Q4=m12m4	0,00046	0	0,00023	0,00154	0	0,00077

Table 2. *Représentation des scores des documents.*

On peut noter les remarques suivantes à partir de la table 2 dans laquelle scts, sctc et c(scts,sctc) désignent respectivement le score d'un document avec des termes simples, le score d'un document avec des termes composés et la combinaison des deux scores précédents :

– Avec la requête Q1= « m1m2 » on note que pour le document « d1 » qui contient le terme composé « m1m2 » son score passe de **0,022299** avec des termes simples à **0,21309** avec des termes composés. Par contre pour le document « d3 » qui contient les termes m1 et m2 séparément son score passe de **0,038955** avec des

termes simples à **0,051334** avec des termes composés. Cela montre que le document « d1 » est plus approprié pour la requête Q1= « m1m2 » que le document « d3 » car le premier contient le terme composé, par contre le second contient les termes composants séparément.

– Avec la requête Q2= « m5m6 », on remarque que pour le document « d1 » qui contient le terme composé « m5m6 » son score passe de **0,029922** avec des termes simples à **0,28338** avec des termes composés. Par contre pour le document « d4 » qui contient les termes m5 et m6 séparément son score passe de **0,038669** avec des termes simples à **0,045598** avec des termes composés. Cela montre que le document « d1 » est plus approprié pour la requête Q2= « m5m6 » que le document « d4 » car le premier contient le terme composé, par contre le second contient les termes composants séparément. Cela montre que la représentation avec des termes composés influe considérablement sur les scores des documents en réponse aux requêtes contenant des termes composés.

– Avec la requête Q3= « m1 », on note que le changement des scores obtenus par des termes simples par rapport aux scores obtenus par des termes composés est presque identique pour tous les documents. Cela montre que la représentation avec des termes composés n'influe pas sur les scores des documents en réponse aux requêtes contenant uniquement des termes simples.

5. Conclusion

Nous avons exposé dans cet article une approche qui permet de représenter au mieux le contenu sémantique d'un document. Et cela par l'utilisation des termes composés comme unité d'indexation à coté des termes simples. Pour cela nous avons défini les six caractéristiques mises en jeu dans l'indexation par des termes composés, dans le cadre de modèle de langage. L'exemple d'illustration que nous avons présenté indique que l'approche répond bien à l'objectif fixé . La prochaine étape consiste à mettre en œuvre cette approche, la tester et la comparer aux autres approches (ex : bi-gramme).

6. Bibliographie

Agirre E. et Martinez D. *Knowledge sources for word sense disambiguation* 2001.

Alvarez C., Langlais P. et Nie J-Y. « Word pairs in language modeling for information retrieval », *Proc. of the conference on computer assisted information retrieval*, 2004

Bai, J. et al. « Query expansion using term relationships in language models for information retrieval », *CIKM*, 2005, p. 688-695.

Baziz M. et al. « Semantic cores for representing documents », *20th ACM symposium on applied computing, SAC'2005*, Santa Fe, New Mexico, USA, 13 - 17 mars 2005. ACM-SAC: Press, New York, NY, USA, 2005, p. 1011 - 1017.

Berger A. et Lafferty J. « Information retrieval as statistical translation », *Proc. of 1999 ACM SIGIR conference on research and development in IR*, 1999, p. 222-229.

Cao G., Gao J. F. et Nie J. Y. *Extending query translation to cross-language query expansion with Markov chain.* 2007.

Cao, G., Nie, J.Y. et Bai, J. « Integrating word relationships into language models », *Proc. of 17th ACM SIGIR conference*, 2005, p. 298–305.

Gao J. F.et al. « Dependence language model for information retrieval », *Proc. of 27th ACM SIGIR conference on research and development in IR*, 2004.

Hiemstra D. « A linguistically motivated probabilistic model of information retrieval », *Second european conference, ECDL'98*, Nicolaou C. et Stephanides C. (Eds.), Research and advanced technology for digital libraries, Springer Verlag, 1998.

Jiang M., Jensen E. et Beitzel S. « Effective use of phrases in language modeling to improve information retrieval », *Symposium on AI & math*, Special session on intelligent text processing, Florida, January 2004.

Lafferty J. et Zhai, C. « Document language models, query models, and risk minimization for information retrieval », *Proc. of 24th annual international ACM-SIGIR conference on research and development in information retrieval* , 2001, p.111-119.

Lavrenko V. et Croft W. B. « Relevance-based language models », *Proc. of 24th annual international ACM-SIGIR conference on research and development in IR*, Croft W.B. et al. (Eds.), New Orleans, Louisiana, 2001, p.120-127.

Liu, X. et Croft, W. B. « Cluster-based retrieval using language models », *Proc. of 27th ACM SIGIR*, 2004, p. 186-193.

Miller D. R. H., Leek T. et Schwartz R. M. « A hidden Markov model information retrieval system », Hearst *et al*. (Eds.), 1999, p. 214–221.

Ponte J.M. et Croft W. B. « A language modeling approach to information retrieval », Croft *et al*. (Eds.), 1998, p. 275–281.

Sasa P. *et al.* « Comparison of collocation extraction measures for document indexing », *Journal of Computing and Information Technology*, vol. 14, n°4, 2006, p. 321–327.

Song F. et Croft W. B. « A general language model for information retrieval » *Proc. of SIGIR'99*, 1999.

Srikanth M.et Srihari R. « Biterm language models for document retrieval », *Proc. of 25th annual international ACM SIGIR*, Finland, 2002, p. 425–426.

Tao Tao, *et al.* « Language model information retrieval with document expansion », *Proc. of the human language technology conference of the north American chapter of the ACL*, New York, 2006, p. 407–414.

Wei, X. et Croft W. B. « LDA-based document models for ad-hoc retrieval », *Proc. of 29th annual international ACM SIGIR conference on research and development on IR*, 2006.

Wei, X. et Croft, W. B. « Investigating retrieval performance with manually-built topic models » Proc. *of RIAO 2007 - 8th conference large scale semantic access to content (text, image, video and sound)*, paper number 12, 2007.

Survey of the Adequate Descriptor for Content-Based Image Retrieval on the Web: Global versus Local Features

Hichem Bannour* — **Lobna Hlaoua**** — **Bechir Ayeb*****

Départements des sciences d'informatiques,
** Institut Supérieur des Sciences Appliquées et de Technologies de Sousse (ISSATS),*
*** Ecole Supérieure des Sciences et de Technologie de Hammam Sousse (ESSTHS),*
**** Faculté des Sciences de Monastir (FSM), Tunisia.*

*Hichem.Bannour@issatso.rnu.tn**
*Bannour.Hichem@yahoo.com –lobna1511@yahoo.fr ** –Ayeb_b@yahoo.com ****

ABSTRACT. *The need for effcient content-based image retrieval has increased hugely. Two methods are recognized for describing the content of images: using global features and using local features. In this paper, we propose two methods for image retrieving based on visual similarity. The first one characterizes images by global features, when the second is based on local features. In the global descriptor attributes are computed on the whole image, whereas in the local descriptor attributes are computed on regions of the image. The aim of this paper is to compare global features versus local features for Web images retrieval.*

RÉSUMÉ. *On reconnait actuellement, dans les systèmes de recherche d'image par contenu, deux méthodes pour la description du contenu des images : à travers des attributs locaux ou à travers des attributs globaux. Dans ce papier, nous proposons deux méthodes pour la recherche d'image qui sont basées sur la similitude visuelle. La première caractérise les images par des attributs globaux, alors que la seconde est basée sur les attributs locaux. Concernant le descripteur global, les attributs sont calculés sur l'ensemble de l'image, alors que pour le descripteur local, les attributs sont définis sur les régions de l'image. L'objectif de ce papier est d'évaluer les performances des attributs locaux contre les attributs globaux pour la recherche des images Web par contenu.*

KEYWORDS: *Content-based image retrieval, image segmentation, image features, local descriptor, global descriptor.*

MOTS-CLÉS : *Recherche d'image par contenu, segmentation d'image, attributs d'image, descripteur local, descripteur global.*

1. Introduction

The digit contents are being generated with an exponential speed. As the amount of collections of digital images increases, the problem finding a desired image in the web becomes a hard task. Therefore, an efficient method to retrieve digital images on the Web is required.

There are two approaches to image retrieval: Text-Based approach and Content-Based approach.

– The former solution is a more traditional approach, which indexes images by using keywords. The keyword indexing of digital images is useful, but it requires a considerable level of effort and is often limited to describe image content.

– The alternative approach, the content-based image retrieval, also called CBIR, indexes images by using the low-level features of the digital images, and the searching task depends on features being automatically extracted from the image.

Most current CBIR systems retrieve images from a collection on the basis of the low-level features of images, such as color, texture, and shape. Almost all these systems are founded on the premise that images can be characterized by global descriptors to retrieve purposes in a database (Flickher *et al.,* 1995, Wu *et al.,* 2004, Quack *et al.,* 2004, Pi *et al.,* 2005). The global descriptor consists of features computed on the whole image. For example, in (Rubner *et al.,* 1997) authors proposed a Histogram search algorithms to characterize an image by its color distribution or histogram, they proposed the earth mover's distance (EMD) using linear programming for matching histograms.

However, in most cases the images represent a scene consisting of different objects (or regions), and therefore, a description of these regions should allow a better representation of the image content. The solution consists in separating the different regions of the image using a segmentation algorithm, then to use the appropriate features calculated on each region of the image to describe (Liu *et al.,* 2000, Jing *et al.,* 2004, Chen *et al.,* 2002). These features constitute the local descriptor. For example, The Stanford SIMPLIcity system (Wang *et al.,* 2001) uses statistical classification methods to group images into rough semantic classes, such as textured-non textured, graph-photograph.

In this paper we propose two methods for content based image retrieval. Our methods describe a given image on the basis of color and textures features, and are based on statistical moments for color characterization and the Tamura features (Tamura *et al.,* 1978) for texture description. Our methods, namely GDIR and LDIR, use respectively global features and local features for image description. Compared to other works, GDIR and LDIR proved to achieve higher accuracy.

Another issue in this work is to evaluate the accuracy of global descriptors versus local descriptors for image characterization and retrieval in the Web domain. In (Shyu *et al.,* 1998), authors compared local and global descriptor for medical image retrieval. They concluded that the empirical evaluation of their current implementa-

tion illustrates that local features significantly improve retrieval performance in the domain of HRCT of the lung. But, their experiments still miss details to compare efficiently the two descriptors.

Motivated by the above considerations and the lack of an accurate comparison in the literature between the two descriptors, we propose in this work to evaluate the accuracy of local versus global descriptor for web image retrieval.

The rest of the paper is structured as follows. In Section 2, we present the features used for image description. In Section 3, we introduce the global image description method. Section 4 illustrates the local image description method. Simulation and retrieval results will be reported in Section 5. The paper is concluded in Section 6.

2. Image Features

In this section, we introduce the image features used by our two methods for images description.

2.1. *Color Features*

The statistical moments is considered to be invariant to image shift, rotation and scale. Actually, moments also represent fundamental geometric properties of a distribution of random variables. In this proposal we used the statistical moments for color description. The used color descriptor is composed by the following attributes:

– Colors expectancy:

$$E_i = \frac{1}{N}\sum_{j=1}^{N} P_{ij} \qquad [1]$$

– Colors variance:

$$\delta_i = \left(\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - E_i)^2\right)^{\frac{1}{2}} \qquad [2]$$

– Skewness:

$$\sigma_i = \left(\frac{1}{N}\sum_{j=1}^{N}(P_{ij} - E_i)^3\right)^{\frac{1}{3}} \qquad [3]$$

Where P_{ij} is the $(i,j)\ pixelcolor$, N is the total number of pixels in the image.

These values allow to estimate the average color, the dispersion of color values from the average and the symmetry of their distribution in a region of the image (or respectively on the whole image).

2.2. *Textural Features*

In this work, we used Tamura texture features as texture descriptor of an image in the database. Tamura et al. (Tamura *et al.*, 1978) took a different approach based on psychological studies on human visual perception. They developed computational approximations for six different visually meaningful texture properties, namely, *coarseness*, *contrast*, *directionality*, *line-likeness*, *regularity*, *and roughness*. However, only three of the six proposed features correspond strongly to human perception and are widely used. These features are *coarseness*, *contrast* and *directionality* which describe respectively the "coarse vs. fine", "high vs. low" and "directional vs. non-directional" of a textured regions. In this proposal, we use these three described features in both descriptors.

3. Global Image Descriptor

The global image descriptor is composed by color and texture features being computed on the entire image.

The texture features are not always an accurate description of the image because they are computed on the whole image. Therefore, in the retrieval process we provide two alternatives to user, the first one is based on color features, the second is based on combined features (color and texture). When the retrieval based on color is fruitless, the user can use the other alternative. By integrating these two options, retrieval accuracy may be improved significantly.

4. Local Image Descriptor

The local image description is founded on the premise that images can be characterized by attributes computed on regions of the image. To separate the different regions of a given image we used an image segmentation method. So to compute our local image descriptor, we use the SOM algorithm to separate the homogeneous regions, than for each region we calculate the color and texture features described in section 2 - An example of local color descriptor is shown in Figure 3.

4.1. *Image Segmentation by Color Clustering*

Different types of neural networks have been proposed for the segmentation of color image (Dong *et al.*, 2005) (Wang *et al.*, 2003) (Ong *et al.*, 2002). However, SOM has the advantages of nonlinear projection, topology preserving, prominent visualization and rapid convergence, which makes it particularly useful for the color clustering (Kohonen, 1995).

For the broad domain images, such as the images in World Wide Web or in images library, precise object segmentation is nearly as difficult as image understanding. However, semantically precise segmentation is not needed to our system because our approach is insensitive to segmentation. In this work we chose to use the Self Organizing Map "SOM" for image segmentation.
The SOM is structured as a two-layer neural network with a rectangular topology as shown in Figure 1. Three inputs (R,G and B) are fully connected to the neurons on a 2-D plane. Each neuron is a cell containing a template against which inputs are matched. The template is the weight values to the neuron i, which is represented by $w_i = [w_{i1}, w_{i2}, w_{i3}]^T$. The SOM training has the following procedure:

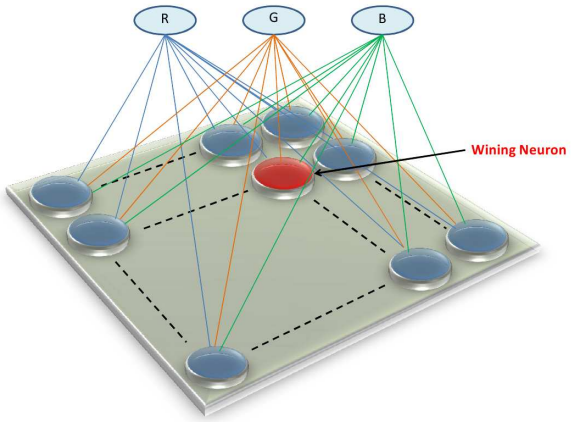


Figure 1. *SOM Topology.*

1) Initialization: Define the SOM map, and set the size of SOM to $n * n$. Set the neighborhood radius to n and the learning rate to 1. Randomly initialize the weight vector. The neighborhood type is Gaussian. The SOM training is successively performed by two phases. The weight vectors of the map are ordered in the first phase, and fine-tuned in the second phase.

2) Input: The input colors are randomly initialized. The image colors are reiteratively used to train the network for few times. During the training, each color point is cyclically chosen from the data set, and presented to all neurons on the map simultaneously.

3) Competitive Process: At time t, color point x(t)=[r(t),g(t),b(t)], T is presented to the network. The winning neuron c is computed with the shortest distance between the color point and weight vectors by the formula:

$$\|x(t) - w_c(t)\| = \min_i \|x(t) - w_i(t)\| \qquad [4]$$

$$where: \quad c = \min_i \|x(t) - w_i(t)\| \qquad [5]$$

4) Cooperative Process: The topological neighbors of winning neuron c are determined by the Gaussian function centered at neuron with the effective scope of $\Re_c(t) = [c_{k-1}, c_k, c_{k+1}]$.

5) Adaptive Process: The weights of winning neuron c and its neighbor neurons are updated within the neighborhood using formula 6 when $k \in \Re_c(t)$

$$w_i(t+1) = w_i(t+1) + \alpha h_{c_i}(t)[x(t) - w_i(t)] \qquad [6]$$

where, $\alpha(t)$ is the learning factor, and $h_{ci}(t)$ is the neighborhood function centered around the winning neuron .

6) Iteration: The next color point is presented to the network at time $t+1$. the learning rate α is decreased to $\alpha(t+1) = \alpha(0)(1 - t/T)$. The neighborhood radius is decreased to $\Re_c(t+1) = \Re_c(t)(2 - t/T)$. The new winning neuron is chosen by repeating the procedure from step 2 until all iterations have been made $t = T$.
T is the number of color points for training.

5. Experiments

Our methods has been implemented with a general-purpose image database including about 100 000 pictures, which are stored in JPEG format with size 384*256 or 256*384. To perform our proposal results, we evaluate retrieved images on the basis of local descriptor and global descriptor. The remaining experimental results are evaluated in terms of precision and recall. We used also the accuracy measurement to compare our results, which is the mean of recall and precision. The assessments are giving according to 10 classes, each containing 100 pictures, defined in the COREL database (COR 1999).

5.1. *Simulation*

Figure 2 demonstrates the results of image segmentation. Figure 2(a) represents the input image. 2(b) shows the obtained image after a segmentation by SOM with a map sized to 16*16 and 2(c) illustrates the obtained color classes by the same network. In 2(d) image segmentation by a SOM map of to 2*2 and in 2(e) the obtained color classes by the same network.

For the rest of our experiments, we used the following parameters for SOM:

– SOM size is 2*2.

– The neighborhood radius $\Re_c(t)$ is 1.

– the learning rate α is 0.8 and decreasing with time following this formula: $\alpha(t+1) = \alpha(0)(1 - t/T)$.

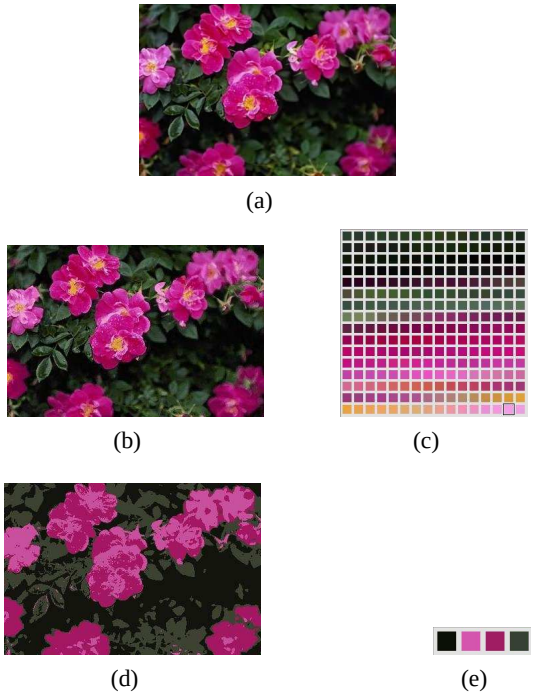


Figure 2. *Image segmentation using SOM*

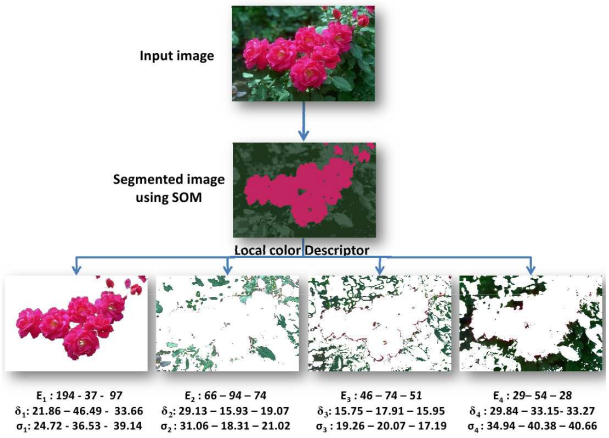


Figure 3. *An example of local color descriptor.*

In figure 3 we illustrate the process to obtain the local color descriptor. After image segmentation, we compute for each region the colors expectancy E_i, the colors variance δ_i and the Skewness σ_i, and put these values in the image vector descriptor.

5.2. *Similarity Measures*

In this section, we describe the similarity measures that we used for image retrieval. Each image in the database is represented by a vector descriptor containing both color and texture features which have been described above. In the retrieval process, for a given query we evaluate the relevance of each image according to a distance measurement defined as follows:

$$d = \sqrt{\sum_{X,X' \in F} \sum_{i=1}^{n} \sum_{j=1}^{3} (\min_{k=1}^{n}(X_{ij} - X'_{kj}))^2} \quad [7]$$

where :

– F is the set of image features $F = \{E_i, \delta_i, \sigma_i, C_i, Co_i, D_i\}$
with E_i is the Expectancy (Equation.1), δ_i is the Variance (Equation.2), σ_i is the Skewness (Equation.3), C_i is the Contrast, Co_i is the Coarseness and D_i is the Directionality.

– X and $X^{'}$ are features of respectively the query image and the target image.

– n is the number of regions in the image.

– 3 is the size of color components (R,G,B).

This equation corresponds to the Euclidean distance, which allows to measure the similarity of two images according to the used features F, on each color components (R,G,B). For both images, we compute the distance between features computed on a region of the query image, and the most close region on the target image. This distance is applicable to the local descriptor and the global descriptor. In global descriptor the number of regions in the image is 1, while in local descriptor the number of regions is n.
The retrieval result is a set of images ranked according to the scores given by the above equation.

5.3. *Connection to other works*

Because we have access to the SIMPLIcity system (Wang *et al.*, 2001), we compare the accuracy of our methods to it using the same COREL database. SIMPLIcity had been compared with the original IBM QBIC system and found to perform better. Also, we compare our methods to the EMD-based color histogram system (Rubner

et al., 1997) to prove that statistical moments work faster and give better results than histograms. To qualitatively evaluate the accuracy of our methods over the image database, we randomly pick 10 query images, namely, Africa people, beach, buildings, buses, dinosaurs, elephants,flowers, horses, mountains and food.

To perform a fair comparison to the other works, we used the same experimental protocol than the one of SIMPLIcity. Precision within the first 100 retrieved images was computed for our methods, SIMPLIcity and EMD-based color histogram. Recall was not used in the SIMPLIcity experiments, so in this experiment too.

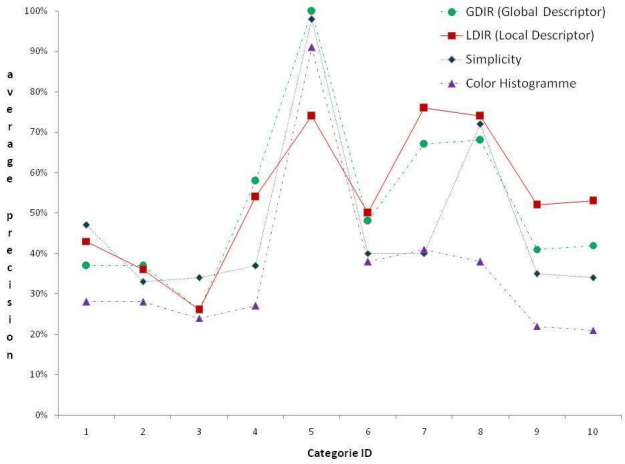


Figure 4. *Comparing global and local descriptor with SIMPLIcity and color histogram methods on average precision.*

Figure 4 shows the performance of our methods when compared to the SIMPLIcity and the EMD-based color histogram systems. Clearly, the color histogram-based matching systems perform much worse than the GDIR and LDIR systems in almost all image categories. To compute the feature vectors over 100 000 color images of size 384*256 requires approximately 120 minutes for the GDIR, and 652 minutes for LDIR, making a computation time per image of 0.072 sec for GDIR and 0.391 for LDIR. So, it is clear that our methods based on statistical moments work faster and give better results than the histogram based method.

Except for the Africa people, buildings and dinosaurs category, our methods has achieved better results than SIMPLIcity. For the other categories the difference between our methods and the other systems is quite significant. On average, the precision of GDIR and LDIR are higher than those of SIMPLIcity and EMD-based color histogram, and respectively equal to 52%, 54%, 47%, and 36%.

5.4. *Local descriptor vs. global descriptor*

Figure 5 shows that the accuracy of the local descriptor follows a linear curve, while the global descriptor curve varies according to the sought image. Thus, the accuracy of the global descriptor depends on images in the database and the query used for retrieving propose, while the local descriptor is more robust to these criteria. Also, the plot shows that the local descriptor accuracy is higher than the global one, except to images representing a bus, this is because of the different backgrounds of this category of images.

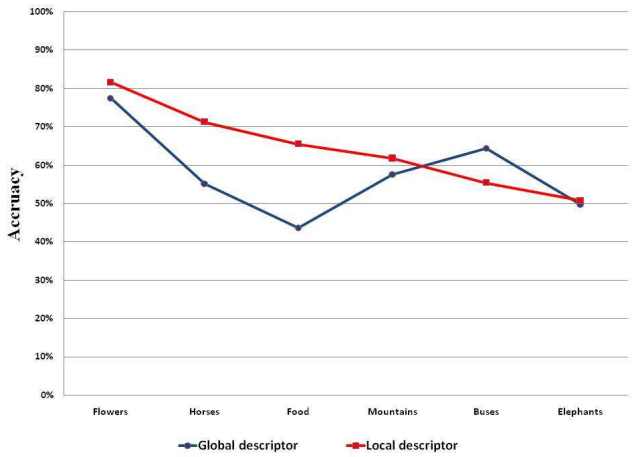


Figure 5. *Evaluation of descriptors accruacy on the COREL image base.*

Table 1 shows the results obtained for each of the seven category we defined into the COREL Database. The "size" row is the number of correct images in the database, the "Res" rows are the number of images returned by our methods when performing a query, the "Pre" rows are the precision, the "Rec" rows are the recall and the "Acc" rows represent the accuracy. Precision rate allows to estimate the relevant images ratio and the recall rate allows to estimate the ratio of relevant images omission.

A comparison between the global descriptor results and those of the local descriptor shows that, in the most of cases, the local descriptors can improve significantly the precision of the retrieval result. However, the recall is almost the same for both descriptors. Note that the accuracy of the local descriptor is also better than the global one. The average values confirm these findings more clearly.

However, the experiment with synthetic images (buses) shows that the global descriptor allows a better retrieval result. The system achieves an accuracy of 99% with the global descriptor, when it achieves an accuracy of 76.37% with the local descriptor.

From these results, we can see that the local descriptor achieves a higher accuracy when the desired image possesses several meaningful regions. However, when the

Table 1. *Obtained results on a subset of the COREL database using global descriptor and local descriptor.*

Category	size	Global Descriptor results				Local Descriptor results			
		Res	Pre (%)	Rec (%)	Acc (%)	Res	Pre (%)	Rec (%)	Acc (%)
Flowers	100	125	68,80	86,00	77,40	119	88,24	75,00	81,62
Horses	100	149	44,30	66,00	55,15	108	68,52	74,00	71,26
Food	100	86	37,21	50,00	43,60	122	59,02	72,00	65,51
Mountains	100	180	41,11	74,00	57,56	135	52,59	71,00	61,80
Buses	100	99	64,65	64,00	64,32	106	53,77	57,00	55,39
Elephants	100	86	53,49	46,00	49,74	97	51,55	50,00	50,77
Average values			**51,59**	**64,33**	**57,96**		**62,28**	**66,50**	**64,39**
Dinosaurs	100	98	100,0	98,00	99,00	119	69,75	83,00	76,37

image possess insignificant backgrounds, like in synthetic images where backgrounds do not represent any relevant information, the global descriptor is more useful.

Finally, we notice an important property during our experiments, is that the global descriptor allows a better Recall for the first 20 retrieved images; however the local descriptor allows a better recall on the total retrieved image.

6. Conclusion

In this paper, we proposed two methods of content based image retrieval according to visual similarity. The first method consists in indexing the images automatically through global features calculated on the whole image, while the second consists in indexing the image using features calculated on the regions of the image. An empirical assessment of the two methods shows that the local descriptor significantly improves the performance of research in the Web domain as it can retrieve more relevant images.

However, these methods are still limited to visual similarity retrieving and the used descriptors are often describing a statistical relationship on images features. This implies that searching task is semantically very poor and usually presents a very low individual meaning. So it is clear that there are a large semantic gap between the extracted features and the semantic level of the users' expectation expressed throw their queries. Hence, we plan to perform our image retrieval system using the semantic features.

7. References

Chen Y., Wang J. Z., « A Region-Based Fuzzy Feature Matching Approach to Content Based Image Retrieval », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n° 9, p. 1252-1267, September, 2002.

COR, « Corel image database », 1999.

Dong G., Xie M., « Color clustering and learning for image segmentation based on neural networks », *IEEE Transactions on Neural Networks*, vol. 16, n° 4, p. 925-936, July, 2005.

Flickher M., Sawhney H., Niblack W., Ashley J., Huang Q., Dom B., Gorkani M., Hafner J., Lee D., Petkovic D., D.Steele, Yanker P., « Query by Image and Video Content: The QBIC System », *IEEE Computer*, vol. 28, n° 9, p. 23-32, September, 1995.

Jing F., Li M., Zhang H.-J., Zhang B., « An efficient and effective region-based image retrieval framework », *Image Processing, IEEE Transactions on*, vol. 13, n° 5, p. 699-709, May, 2004.

Kohonen T., « Self-Organizing Maps », Springer-Verlag, Berlin, Germany, 1995.

Liu F., Xiong X., Chan K. L., « Natural Image Retrieval based on Features of Homogeneous Color Regions », *SSIAI '00: Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, IEEE Computer Society, Washington, DC, USA, p. 73, 2000.

Ong S. H., Yeo N. C., Lee K. H., Venkatesh Y. V., Cao D. M., « Segmentation of color images using a two-stage self-organizing network », *Image and Vision Computing*, vol. 20, n° 4, p. 261-271, April 1, 2002.

Pi M., Mandal M., Basu A., « Image retrieval based on histogram of fractal parameters », *Multimedia, IEEE Transactions on*, vol. 7, n° 4, p. 597-605, Aug., 2005.

Quack T., Mönich U., Thiele L., Manjunath B. S., « Cortina: a system for large-scale, content-based web image retrieval », *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, ACM Press, New York, NY, USA, p. 508-511, 2004.

Rubner Y., Guibas L. J., Tomasi C., « The earth movers distance, multi-dimensional scaling, and color-based image retrieval. », *APRA Image Understanding Workshop*, p. 661-668, May, 1997.

Shyu C. R., Brodley C. E., Kak A. C., Kosaka A., Aisen A., Broderick L., « Local versus Global Features for Content-Based Image Retrieval », *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries*, IEEE Computer Society, Washington, DC, USA, p. 30, 1998.

Tamura H., Mori S., Yamawaki T., « Texture Features Corresponding to Visual Perception », , vol. 8, n° 6, p. 460-473, 1978.

Wang J. H., Rau J., Liu W. J., « Two-stage clustering via neural networks », *IEEE Transactions on Neural Networks*, vol. 14, n° 3, p. 606-315, May, 2003.

Wang J. Z., Li J., Wiederhold G., « SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, n° 9, p. 947-963, 2001.

Wu H., Lu H., Ma S., « WillHunter: interactive image retrieval with multilevel relevance », vol. 2, p. 1009-1012 Vol.2, Aug., 2004.

Aide à l'interprétation de documents juridiques

Une approche centrée utilisateur

Youssouf SAIDALI* —Julien LECANU* —Eric TRUPIN* Jacques LABICHE*

** Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes, Université de Rouen,*
Avenue de L'Université, 76800 Saint Etienne du Rouvray
Prenon.Nom@univ-rouen.fr

RÉSUMÉ. *Nous présentons un projet de recherche en cours visant à améliorer les interactions d'utilisateurs de différentes catégories professionnelles avec un système d'information dédié au droit du transport et de la logistique. L'objectif vise à concevoir et à mettre au point un environnement numérique de travail (ENT) destiné à un public professionnel (entreprises de la filière logistique, juristes, risk managers, assureurs, avocats, .). et non professionnel (usagers ou salariés des transports). Après avoir posé la question de l'appropriation des contenus dans le cadre des documents numériques, nous décrirons les spécificités de notre corpus de travail. Nous placerons alors notre projet dans un cadre théorique actuellement novateur au sein des sciences cognitives, celui de l'énaction. Ceci nous amènera à proposer une approche résolument centrée utilisateur dans la conception de l'ENT. Nous terminerons par une description des spécifications du futur ENT, qui privilégie une démarche interprétative dans la formulation/reformulation de requêtes, ainsi que la représentation graphique des données.*

ABSTRACT. *In this paper, we present a research project which aims to improve human interactions of various professional categories with an information system dedicated to transport law and logistics. The goal is to conceive and develop a digital environment of work for professionals (companies of the logistic die, lawyers, risk managers, insurers, lawyers,) and non-professionals (users). After having discussed about contents appropriation in digital documents, we will describe specificities of our corpus. Then we will place our project within an innovative theoretical framework in cognitive sciences, that of "enaction". This will lead us to propose user centered approach in the design of the digital environment. We will end with the specifications description of the digital environment, which privileges an interpretative approach in the formulation/reformulation of requests, as well as data vizualisation.*

MOTS-CLÉS : *Environnement numérique, classification, usages, interfaces, visualisation.*

KEYWORDS: *Numerical environment, clustring, uses, interfaces, visualization.*

1. Introduction

1.1. *Interprétation de documents juridiques*

Avec l'apparition des techniques numériques et de l'internet, la distance entre la population et l'information économico-juridique tend apparemment à se réduire. Sur le plan juridique, il est aujourd'hui possible d'accéder à un large pan de la réglementation et de la jurisprudence, qu'elles soient françaises ou étrangères. Mais ce rapprochement technique n'est pas pour autant signe d'une meilleure maîtrise et appropriation de l'information. Les spécialistes en sciences de l'information font le constat des écarts (des fossés) entre prouesses technologiques et appropriation des contenus par des utilisateurs de plus en plus hétérogènes. La question de l'appropriation ne se résout pas en effet par le simple ajout de métadonnées aux documents numériques comme dans le projet du Web Sémantique où l'objectif annoncé par Tim Berners-Lee (1998), initiateur du projet et directeur du W3C, est d'enrichir (notamment au moyen des technologies XML) les documents (à l'aide d'ontologies normalisées, soit automatiquement, soit en assistant leurs auteurs) avec « des informations sur leur propre sémantique qui soient directement interprétables par des agents logiciels sans la supervision d'une interprétation humaine ». Ce positionnement fait l'hypothèse que la valeur sémantique d'un passage de document n'est le fait que de son auteur (alors que c'est tout autant celui de son lecteur confronté à ses pratiques professionnelles). La réponse du Web sémantique est d'indexer les textes avec les concepts d'une ontologie partagée par une large communauté sans que celle-ci ne soit d'ailleurs clairement définie, ni surtout que soient prises en considération la diversité et l'évolution des pratiques langagières au sein de sphères d'activités hétérogènes comme peuvent l'être ici celles des acteurs du transport.

Constatant que les systèmes actuels (IHM, bases de données, ...) conduisent à une interaction Système/Utilisateur forcément appauvrie, parce qu'ancrée dans un environnement prédéfini, (Peschard, 2004) celui des réponses, sous forme de thésaurus qui réorientent la question de l'utilisateur, nous jetterons ici les bases de la conception d'un environnement numérique de travail (E.N.T.) capable de s'enrichir d'apports successifs dus aux interactions de plus en plus denses et complexes au sein de sphères d'activités. Cela nous conduit à sortir de la problématique du mot-clé, ou du figement lexical (représentation de connaissances), pour celle de la thématique des textes et de l'interprétation située. Notre démarche fait l'hypothèse que la valeur sémantique d'un passage est d'abord le fait de son lecteur (entité pouvant être collective) qui *grâce à cette étrange faculté de l'esprit qui est de relier* (Vico, 1986) tracera ses thématiques en fonction de son environnement en même temps qu'il constitue un corpus de textes par sa navigation intertextuelle.

Nous nous intéressons plus particulièrement à la conception d'un *Environnement Numérique de Travail* (E.N.T.), sorte d'extranet dédié aux usages de la filière transport et logistique. Certes, cet ENT ne recèlera guère de fonctionnalités inédites.

En revanche, l'intégration d'un ensemble de ressources et de services interopérables en son tout, dédiés non pas à une collection de cas d'usages particuliers, mais justement à une sphère d'activité (transport et logistique) large et en évolution rapide, constitue une réelle nouveauté, voire une singularité. La mise en œuvre de ce dispositif est susceptible de contribuer à des évolutions notables de l'usage de documents réglementaires et, plus généralement, des activités des acteurs de la filière transport et logistique.

1.2. *Accès au corpus et difficultés de l'appropriation des contenus*

Le corpus réglementaire est encore difficile d'accès malgré une forte demande sociale et économique tout particulièrement en transport et logistique. A ce jour, le corpus et la base documentaire de l'Institut du Droit International du Transport (IDIT[1]) sont accessibles en ligne. Cette base s'adresse à des adhérents spécialistes du droit, mais elle est difficilement utilisable par un novice dans le domaine juridique comme un transporteur, qui chercherait des informations pour la mise en place de conditions de transport de marchandises conformes à la législation en vigueur, par exemple. Cette base documentaire est associée à un thésaurus hiérarchisé « maison » pour améliorer son interrogation. Elle impose aussi la saisie manuelle de comptes-rendus (CR) de jurisprudence et de réglementation sous la forme de fiches. Cette captation de l'information et la veille présentent des difficultés majeures pour renseigner et mettre à jour le système d'information. Nous envisageons, suite à la numérisation (en cours) des collections papiers (des milliers d'articles et de décisions de justice relatifs à des risques et litiges en matière de transports), une aide à l'interprétation de contenus textuels.

Le SI de l'IDIT est conçu pour diffuser des informations aux adhérents afin qu'ils puissent gérer dans les meilleures conditions leurs entreprises et sécuriser leurs activités. Or, la fragmentation de l'information relative au droit des transports et de la logistique qui couvre des domaines très variés rend difficile son accès, d'où la nécessité d'une mise en relief (signalement pour interprétation) de celle-ci.

2. Une approche centrée utilisateur

Notre approche de l'accès aux documents se situe à l'opposé de celles défendues dans le cadre du Web Sémantique. Là où le Web Sémantique cherche à rendre le plus possible partagées de vastes ontologies qui synthétisent une connaissance devant convenir à tous les utilisateurs, nous préférons manipuler des ressources termino-ontologiques (bases de données terminologiques) propres à un utilisateur ou un petit groupe d'utilisateurs et liées à leur tâche, leurs besoins et de leurs centres d'intérêt. Il en découle une certaine *légèreté* sémantique de ces ressources, au sens

[1] L'IDIT, créé en 1969, est une association qui, aux termes de ses statuts, a pour objet l'étude de toutes les questions d'ordre juridique intéressant les transports

de (Perlerin, 2004), dans la mesure où elles ne représentent que ce qui est important du point de vue de l'utilisateur et restent ainsi de taille raisonnable (une centaine de termes) ce qui les rend moins complexes à construire, à maintenir et à enrichir.

Cette approche centrée utilisateur conduit à opérer un certain renversement scientifique relativement aux ressources qu'utilisent les modèles de TAL. Premièrement, d'un point de vue très pratique, force est de constater que des ressources très généralistes, valables pour tout type de traitement envisagé ainsi qu'à destination de tout utilisateur potentiel, ne sont pas facilement disponibles (sous forme électronique pour des traitements automatiques) et encore moins gratuites. Deuxièmement, nous soutenons que l'idée même d'une ressource généraliste est illusoire car elle dépend inévitablement du contexte qui lui préexiste. Le rapport de l'Action Spécifique 32 du CNRS/STIC en 2003 (Charlet *et al.*, 2003) va également dans ce sens en précisant un obstacle au projet du Web Sémantique : la détermination et l'ajout, même de simples méta-données, n'est pas une activité naturelle pour la plupart des personnes.

Les ressources qui sont les plus importantes pour un utilisateur dans une instrumentation informatique pour l'accès aux documents sont celles qui doivent être produites de manière endogène dans une boucle d'interaction entre un outil logiciel, un utilisateur et des corpus. L'accès personnalisé au contenu s'inscrit dans un processus interprétatif en aller-retour entre des outils, des corpus et des ressources personnelles, les uns étant conditionnés par les autres.

Dans notre approche herméneutique et énactive du langage, nous mettons l'accent sur l'interprétation plus que sur les connaissances. Ainsi la priorité est donnée aux spécificités sociolinguistiques des utilisateurs (par exemple leurs centres d'intérêt, leurs habitudes terminologiques, leurs parcours interprétatifs). Ce qui a du sens pour les utilisateurs ne se réduit pas à une représentation et encore moins à une formalisation. Ce n'est pas le résultat d'un calcul, c'est une activité au centre d'une interaction homme-machine. Ainsi, on remet en cause l'idée qu'un mot, une phrase, un texte ou un corpus ait du sens, pour défendre plutôt l'idée qu'ils font sens dans un couplage personne-système.

L'utilisateur et lui seul est capable de dire si un ensemble de documents en retour à sa requête est pertinent ou pas, au vu de sa problématique. Il doit être actif à chaque étape du processus de recherche d'information.

2.1. *Prétraitement et préparation du corpus*

La base de données de l'IDIT comporte plus de 40.000 fiches (jurisprudences, articles, réglementations, acquisitions). Chaque fiche est composée de différentes rubriques (Numéro de la fiche, Thèmes, Date de la décision, Mode de transport, Pays, Objet, Sommaire, Référence), et stockée en format texte.

Pour valider notre démarche, nous effectuerons les premiers tests sur un échantillon restreint du corpus, soient 1369 fiches ayant comme thème « *Conteneur* ».

La première étape de notre développement est de proposer une représentation des documents du corpus. Nous utilisons ici le modèle vectoriel (Salton, 1975), l'une des approches les plus courantes dans le domaine de la recherche d'information.

Nous représentons, les fiches et les requêtes dans un espace vectoriel engendré par l'ensemble des termes contenus dans les fiches. Un terme présent dans un document équivaut alors à une dimension du vecteur [1]. Ce qui permettra par la suite à l'utilisateur de regrouper des items voisins de façon à faire émerger l'information dont il a besoin.

$$T < t_1, t_2, \ldots, t_M > \quad [1]$$

Nous utilisons un algorithme standard pour la pondération des attributs des vecteurs représentatifs des documents, à savoir la fonction TFIDF [2].

$$tfidf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} . \log \frac{|D|}{\left|\{d_j : t_i \in d_j\}\right|} \quad [2]$$

Avec : - $n_{i,j}$: le nombre d'occurrence du terme t_i dans le document d_j

- $\sum_k n_{k,j}$: nombre d'occurrences de tous les termes dans le document d_j
- $|D|$: le nombre total de fiches dans le corpus
- $\left|\{d_j : t_i \in d_j\}\right|$: le nombre de fiches où le terme t_i apparaît

Nous aurons ainsi la valeur de chaque attribut équivalant à l'importance d'un terme dans un document relativement à l'ensemble des documents. L'avantage de cet algorithme est qu'il va permettre de supprimer la totalité des mots qui seront communs à toutes les fiches (Numéro, jurisprudence, thème, etc.). De plus, cette pondération va permettre à l'utilisateur de mettre en avant les termes les plus représentatifs du document en fonction de sa problématique.

A l'issue de cette étape, nous obtenons un vecteur de 11913 attributs représentatif de chaque document. Le corpus contenant 1369 documents au total, on obtient donc une matrice terme-document de 11913 lignes sur 1369 colonnes. Nous notons ici l'un des inconvénients de ce modèle de représentation. Il crée des vecteurs de très grande taille et de ce fait, difficilement utilisable dans une interaction homme-machine dynamique, notamment pour la classification mais aussi pour la reformulation de requêtes. En effet, un tel vecteur va considérablement augmenter les temps calculs, et ainsi retarder l'obtention des résultats à une requête,

ce qui n'est pas concevable pour un système que l'on veut interactif. Il est donc impératif de réduire sa taille, en limitant au maximum la perte d'information. Nous avons alors cherché à supprimer les termes qui ne portent pas d'information à un instant donné et donc qui ne sont pas discriminants pour la représentation des documents. Nous partons ainsi d'un principe assez répandu en recherche d'information qui dit que les mots les plus fréquents (articles, prépositions, auxiliaires) sont vides de sens. Les premiers utilisateurs (3 spécialistes de l'IDIT), ont défini tout mot comptant moins de quatre lettres comme étant vide de sens. En réalisant cette suppression de termes, nous sommes en mesure de passer la taille d'un vecteur représentatif d'un document à 11178 attributs valable dans le contexte de ce groupe d'utilisateurs. Malgré cette première réduction, le nombre d'attributs de chaque vecteur représentatif reste encore trop élevé. Nous proposons une nouvelle diminution, en nous appuyant sur des méthodes issues de la linguistique, notamment la lemmatisation (ou radicalisation) et sélection de mots pleins, en regroupant les variantes d'un mot.

Exemple : économie, économiquement, économiste économ

Toutefois, un problème non négligeable d'ambiguïté des termes peut se produire. En effet, le fait de réduire les termes à leur radical peut avoir comme conséquence de regrouper des mots ayant le même radical mais n'ayant pas le même sens. Par exemple, on peut faire correspondre « *transporter* » et « *transformer* » à travers le radical « *trans* ». Ces regroupements erronés ont comme conséquence la perte en précision. Cependant, cette radicalisation permet de réduire nos vecteurs représentatifs de manière significative, passant de 11178 à 7243 (Tableau 1).

	à l'origine	Stop words	Lemmatisation
Taille vecteurs	11913	11178	7243

Tableau 1. *Tailles des vecteurs suite aux réductions de dimension*

Le corpus est donc maintenant correctement préparé par et pour l'utilisateur. Il s'agit ensuite de proposer, dans le contexte de l'utilisateur, une classification dans le but de former des ensembles de documents thématiquement proches.

2.2. *Classification contextuelle*

Nous partons simplement du principe que si un document est jugé pertinent à une requête, alors les documents similaires à ce document ont de fortes chances d'être également pertinents. Nous allons donc laisser l'utilisateur regrouper les documents au sein de *clusters*, et relever ses thèmes majeurs ou ceux de sa sphère d'activité. De nombreuses méthodes ont été utilisées en classification de textes (Sebastiani, 2005). Dans la mesure où nous ne disposons pas (et ne souhaitons pas constituer) d'exemples d'items déjà classés et étiquetés, nous nous intéressons ici uniquement aux méthodes de classification non-supervisées pour catégoriser les

documents. Pour faciliter l'usage de l'ENT, et pousser à l'émergence d'information sans connaissances a priori, on intègre plusieurs méthodes exploitables par l'utilisateur.

2.2.1. *Approche par les k-means*

La première méthode de classification que nous avons intégrée dans l'ENT est les k-means (McQueen, 1967). L'intérêt de cette méthode, pour nous, vient du fait que la valeur k, correspondant au nombre de classes à créer n'est pas fixée à l'avance. Il peut donc apparaître comme un élément du contexte utilisateur. Nous avons réalisé les premiers tests de classification du corpus pour différentes valeurs de k. A l'analyse des fiches placées dans un même cluster, les utilisateurs ont ainsi mis en avant que plus k est grand et plus les clusters sont représentatifs de leurs thèmes. Toutefois, si le nombre k devient trop grand, il apparaît une sur-segmentation du corpus. Ceci risque alors de provoquer une perte du contexte pour l'utilisateur dans sa navigation intertextuelle.

L'évaluation de la pertinence dans le tableau ci-dessus correspond à un jugement utilisateur, après navigation dans les fiches contenues dans un cluster.

2.2.2. *Approche par une classification hiérarchique*

Toujours dans le but d'aider à l'émerge nce de connaissances par l'usage de l'ENT, nous proposons également une classification hiérarchique descendante, qui structure le corpus sous forme d'un arbre (Figure 2).

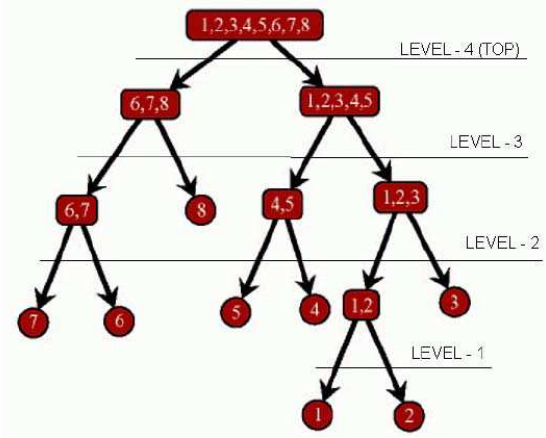


Figure 2. *Exemple de classification hiérarchique ascendante*

Cette classification aboutit à 251 clusters qui contiennent entre 1 et 20 documents chacun. Une analyse rapide de cette hiérarchie valide la répartition des fiches dans les clusters en fonction des thèmes de l'utilisateur. Et montre que dans le contexte courant, deux clusters situés aux extrémités de l'arbre sont jugés totalement

différents. L'avantage de cette seconde approche est qu'elle fournit une topologie facilement représentable et manipulable pour la visualisation et la navigation.

Maintenant que l'utilisateur a classifié ses documents, il peut s'appuyer sur cette structuration pour exprimer sa requête et affiner son besoin d'information.

2.3. *Expression du besoin et reformulation de requête*

Dés lors que le corpus est classifié, et que dans chaque classe les documents sont jugés thématiquement et sémantiquement proches par l'utilisateur, on peut proposer des outils de formulation-reformulation de requêtes dans notre ENT. Cette procédure se décompose en vectorisation de la requête utilisateur, calcul de la distance entre requête et clusters et reformulation par réinjection de pertinence.

2.3.1. *Requête initiale*

Une requête peut être définie comme étant l'expression formalisée d'un besoin d'information exprimé par l'utilisateur. Elle est dans la plupart des cas composée d'une suite de termes permettant de décrire ce besoin d'information. Cependant, pour pouvoir être comparée aux documents du corpus par un calcul de distance, la requête doit être représentée avec le même formalisme que les documents. Elle sera donc représentée par un vecteur (de 7243 attributs) identique à ceux des documents. De plus, à chaque terme de la requête sera affecté un poids, déterminé en fonction du nombre d'occurrences du terme, normalisé par le nombre de termes de la requête. Le poids pour un terme t est ainsi défini par la fonction suivante : $p_t=n/N$ où n est le nombre d'occurrences du terme dans la requête et N le nombre total de termes.

2.3.2. *Calcul de distance*

La requête étant vectorisée, nous pouvons aider l'utilisateur à établir la similarité entre sa requête et les différents documents du corpus [3]. Cette similarité apparaît comme une distance entre les deux vecteurs. Nous aurons donc pour chaque document d_i :

$$sim(q, d_i) = \overrightarrow{v(q)}.\overrightarrow{v(d_i)} \qquad [3]$$

Cependant, nous ne travaillons pas directement sur les documents mais sur des clusters de documents, en calculant un vecteur représentatif de chacun d'eux. Un cluster est alors vu comme la moyenne terme à terme des vecteurs représentatifs des documents. Nous aurons donc :

$$\vec{V}_{cluster} = \sum \frac{\overrightarrow{v_d}}{N} \qquad [4]$$

Avec $\overrightarrow{v_d}$ les vecteurs représentatifs des documents et N le nombre de documents dans le cluster.

Pour déterminer la similarité entre ses requêtes et les clusters, l'utilisateur peut exploiter un cosinus, une métrique fréquemment utilisée en fouille de texte

2.3.3. *Reformulation par injection de pertinence*

Pour affiner de façon incrémentale le besoin utilisateur, nous utilisons une retro-propagation de la pertinence sur la requête. L'idée est de générer une nouvelle requête par une combinaison linaire des éléments de la requête initiale, et de l'avis de l'utilisateur sur les documents extraits. Dans cette approche, si les *n* premiers clusters trouvés sont jugés pertinents[2], ils sont réutilisés pour reformuler la requête de l'utilisateur (Figure 3).

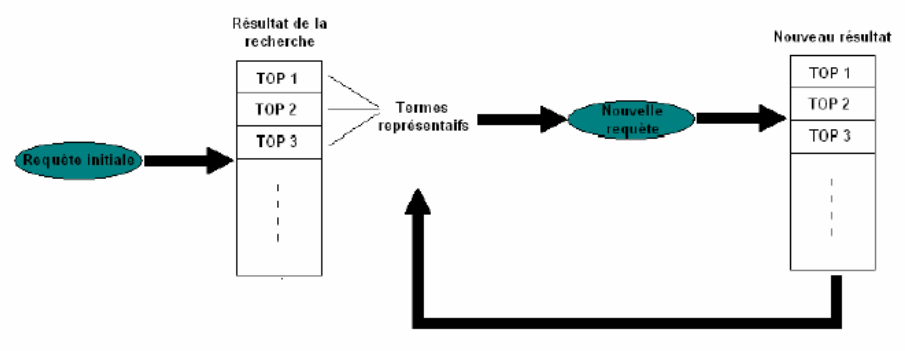


Figure 3. *Schéma de principe de la reformulation*

Les n clusters sélectionnés pour la reformulation sont les premiers clusters du classement réalisés lors de l'étape précédente. Nous ajoutons à la requête initiale de l'utilisateur les termes les plus représentatifs de ces clusters. Par précaution, et pour éviter que les termes ajoutés suite à la reformulation ne deviennent prépondérants par rapport à la requête initiale, ils sont pondérés. Toutefois, on constate les clusters placés loin dans le classement d'origine (*ie* jugés peu pertinents), peuvent remonter.

Nous ne cherchons pas ici, à automatiser le processus, ni à fournir le meilleur résultat. On a fait le choix dès le début de plonger l'utilisateur (ou un petit groupe d'utilisateurs) dans des interactions qui offrent la possibilité de notamment mieux discerner l'homogénéité thématique d'un corpus, de mettre en évidence sa densité, d'en extraire les principales tendances thématiques de chaque document et de permettre un accès rapide à tel ou tel document ou passage de document. Dans le couplage personne-système les interprétations des utilisateurs et les calculs des machines vus précédemment, ne sont pas en concurrence car les uns n'ont en aucun cas le but de supplanter les autres. Au contraire, nous les pensons comme complémentaires dans le sens où l'activité d'une machine a pour objectif de

[2] *n* étant un paramètre que l'utilisateur peut affiner en fonction de son besoin et de la visualisation des résultats

produire dans l'interaction des traces qui vont participer aux interprétations du ou des utilisateurs.

3. *Ver*s un environnement numérique de travail personnalisable

D'un point de vue expérimental, il s'agit de savoir comment un environnement numérique de travail, et le couplage qu'il induit, permettent l'émergence par énaction d'une perception sémantique du corpus et ainsi un meilleur accès aux informations (Varela, 1989).

Dans notre stratégie d'amélioration de la recherche d'information, nous proposons à l'utilisateur plusieurs approches pour naviguer dans l'ensemble des documents, visualiser, manipuler et organiser le résultat de ses recherches. Il pourra notamment s'appuyer sur l'historique de sa navigation, ses propres traces, mais aussi celles qui sont liées à sa sphère d'activité (collectif de travail). Il s'agira donc d'observer l'utilisateur dans son activité, et de lui permettre d'exploiter dynamiquement cette observation. Avec ses traces (volontaires ou involontaires), nous ne cherchons pas à modéliser un comportement pour faire de la prédiction, mais à disposer d'outils de description et d'analyse de la navigation intertextuelle en situation réelle.

La visualisation et l'analyse des résultats de la recherche sont des étapes nécessaires qui s'inscrivent dans le processus global de recherche d'information. La perception de l'information est liée à la prise de décision dans le contexte d'utilisation de l'ENT proposé. L'utilisateur se retrouve au centre d'une boucle itérative « *formulation-analyse-visualisation-reformulation* » dans une représentation globale du processus comme celle de la Figure 4 (inspirée de Kules, 2008).

Le processus est donc initialisé lorsqu'un utilisateur identifie un besoin informationnel et tente de le satisfaire en entreprenant une ou plusieurs tâches de recherche. Il prend des décisions sur la ou les stratégies à adopter, les outils à exploiter et le corpus ou partie du corpus à consulter. Chaque unité d'information découverte peut déclencher de nouvelles idées, suggérer de nouvelles directions et changer la nature même du besoin d'information. On émet alors l'hypothèse que, la gestion sous forme d'historiques de traces (incluant point de blocages et retours arrière) laissées par les différents utilisateurs peut aider à la découverte de nouvelles stratégies et de nouvelles informations.

Pour ce qui est de l'extraction d'information, chaque action implique un engagement cognitif et physique, et peut induire une évolution dynamique de l'interface ou des connaissances en émergence. Nous cherchons à faciliter le couplage et l'engagement de l'utilisateur en lui proposant des outils simples pour la manipulation/sélection/déplacement des documents résultats, ainsi que pour l'expression dynamique des requêtes. Notre démarche consiste donc à utiliser des

représentations spatiales dynamiques pour concevoir et mettre en œuvre une plate-forme générique en personnalisation de la visualisation et en intégration de modalités variées et hétérogènes.

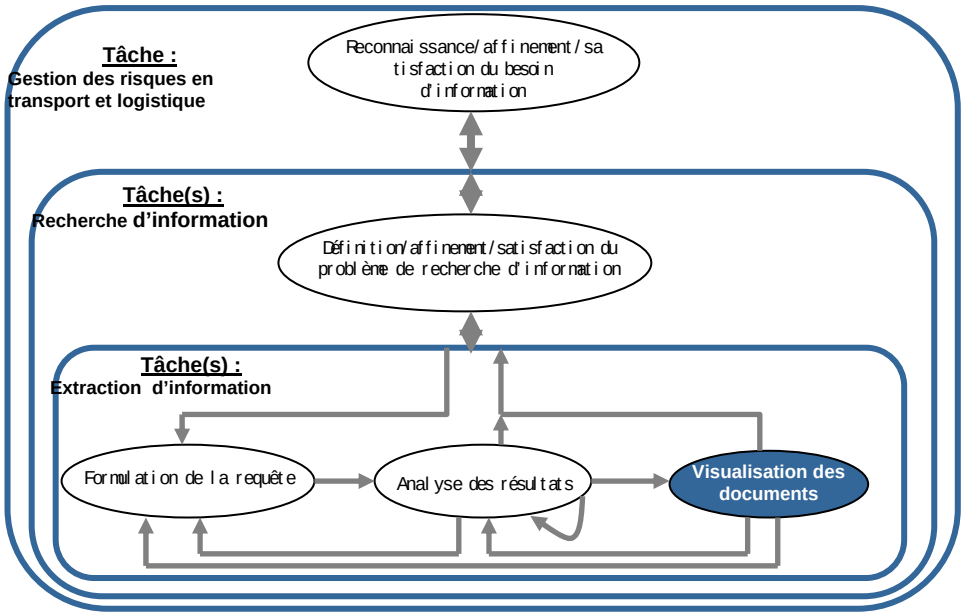


Figure 4. *Visualisation dans un processus d'aide à l'interprétation*

5. Conclusion

A travers cet article et notre projet de mise au point d'un ENT dans le domaine juridique, nous avons voulu poser d'une manière particulière la problématique de la recherche d'information. Nous avons cherché à mettre en avant la complémentarité entre un agent humain énactif et un système classique en RI. Le document électronique est ici considéré de manière indissociable à l'activité du ou des humains qui les produisent, recueillent, indexent et recherchent. Il en découle à notre avis que les approches autour de la recherche d'information ne pourront rester indépendantes des utilisateurs et sans prise en compte des paliers d'intertextualité comme le sont actuellement par exemple les moteurs de recherche ou encore certains projets dans le contexte du web sémantique. Nous proposons d'intégrer plusieurs modalités de manipulation et de navigation dans les données. Cette navigation dynamique et visualisation à différents niveaux de granularité de

l'ensemble des documents permet alors à l'utilisateur de se créer son propre parcours interprétatif.

Plus que jamais la problématique de la recherche d'information requiert des collaborations pluridisciplinaires pour mettre au point, expérimenter et évaluer les conditions d'une relation entre documents et interprétants d'où puisse émerger du sens et de nouveaux usages.

6. Bibliographie

Salton G., Wong A., and Yang C.S., « A Vector Space Model for Automatic Indexing », *Commun. ACM*, vol. 18 (11), 1975, p. 613-620.

Bates M J : «*The design of browsing and berrypicking techniques for the online search information* ». Online review, 13, 407,-431, 1989.

Berners-Lee, T. « *What the semantic web can represent?* » W3C, Disponible à : www.w3.org/designissues/rdfnot.html., 1998.

Peschard, I. « *La réalité sans représentation, la théorie de l'enaction et sa légitimité épistémologique* ». Thèse de Philosophie, 2004, Ecole Polytechnique.

Ricoeur, P. « *Du texte à l'action : essais d'herméneutique* ». Point Seuil, 1986, Paris.

Vico, G. « *Principes d'une science nouvelle* ». (trad JL. Lemoigne) Nagel 1986.

Perlerin, V. *Sémantique légère pour le document* ». Thèse d'informatique. Université de Caen. 2004 .

Charlet, J., Laublet, P., Reynaud, G. « *Web sémantique* ». Rapport de l'Action Spécifique 32 CNRS/STIC, 2003.

Fabrizio Sebastiani. « *Text Categorization* ». In Alessandro Zanasi, editor, Text Mining and its Applications to Intelligence, CRM and Knowledge Management, pages 109--129. WIT Press, Southampton, UK, 2005.

McQueen J. B. « *Some Methods for classification and Analysis of Multivariate Observations* », Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", 1967, Berkeley, University of California Press, 1:281-297.

Varela, F. « *Invitation aux sciences cognitives* ». Seuil, 1989, Paris.

Kules, B., Shneiderman, B. « *Users can change their web search tactics : Designs guidelines for categorized overviews* ». Information Processing and Management, 2008.

Jacko, J. A., Sears, A. «*The Human-Computer Interaction Handbook : Fundamentals, Evolving Technologies and Emerging Applications* ». 2nd Edition, Lawrence Erlbaum Associates, 2006.

Sebastiani, F. «*Text Categorization* ». In Alessandro Zanasi, editor,Text Mining and its Applications to Intelligence, CRM and Knowledge Management, pages 109--129. WIT Press, Southampton, UK, 2005.

Chapitre 8

Articles Jeunes Chercheurs

Une approche sémantique basée sur l'apprentissage pour la recherche d'image par contenu

Hichem Bannour

Département des Sciences d'informatique
Faculté des Sciences de Monastir
Université de Monastir, 5000 Tunisie.

Hichem.Bannour@issatso.rnu.tn
Bannour.Hichem@yahoo.com

RÉSUMÉ. *Cette dernière décennie témoigne un accroissement exponentiel des données multimédia (texte, image, son et vidéo). La recherche d'information au sein de cette masse de données, en particulier les images, devient un processus incertain. Aussi, le problème se pose au niveau de l'indexation puisque les techniques actuelles ne permettent pas de décrire efficacement le contenu des images. Dans ce papier, nous nous concentrons sur le problème de découverte de connaissance à partir du contenu des images et nous proposons une nouvelle approche pour l'indexation sémantique des images. Notre approche se base sur l'apprentissage pour associer des éléments symboliques (concepts) à des éléments de bas niveau.*

ABSTRACT. *Multimedia databases are witnessing an exponential growth. Information retrieval in these data collections, especially in images, becomes a hard task. Furthermore, there is a difficulty in indexing these data since the present techniques don't allow an efficient description of the content of images. In this paper, we focus on knowledge discovery in images database and we propose a new semantic approach for image indexation. Our approach is based on learning of distributions to map low level features to high level concepts.*

MOTS-CLÉS : *Fouille d'image, indexation et recherche d'image, RImC, attributs d'image, reconnaissance d'objet, segmentation d'image.*

KEYWORDS: *Image mining, image indexing and retrieval, CBIR, image features, object recognition, image segmentation.*

1. Introduction

Il devient très difficile et imprécis, de nos jours, de faire la recherche d'information multimédia sur le Web. Les techniques d'indexation et de référencements par mots clés ne suffisent plus à pallier aux besoins croissants des utilisateurs, de plus en plus exigeants en termes de précision face à leurs requêtes.

Pour palier ce problème la communauté scientifique s'est intéressée depuis quelques années à une nouvelle approche appelé Recherche d'Image par Contenu (RImC) et qui consiste à représenter des images par des attributs de bas niveau tels que la couleur, la texture et les formes. Seulement cette approche a été confrontée à un problème important qui est le manque de formalisme précis permettant à un utilisateur d'accéder à ces informations de bas niveau. En effet, un utilisateur cherchant une image sur le web se soucie peu de ses caractéristiques de bas niveau, et formule une requête exprimant une sémantique d'un certain ordre. Cependant, les méthodes actuelles de RImC sont incapables d'abstraction, en d'autres termes, incapable d'extraire de l'information sémantique à partir des images.

Le domaine de la RImC doit donc faire face à une caractéristique importante des images qui est le manque d'un langage de représentation explicite permettant d'en exprimer la sémantique. Cet écart entre les pixels et leur signification est appelé le fossé sémantique (Semantic gap).

Dans ce contexte, l'image mining semble apporter des solutions pour la description sémantique des images. Dans le but d'une indexation sémantique des images, l'image mining consiste en la découverte de connaissances à partir du contenu des images d'un ensemble d'apprentissage, sous la forme d'associations entre des éléments sémantiques (concepts) et des éléments de bas niveau (caractéristiques des images).

Notre objectif dans ce travail est alors d'exploiter l'information contenue dans ces données, et d'en extraire une sémantique fiable pour un besoin de recherche d'information et plus particulièrement pour une indexation sémantique automatique, ou semi automatique des images dans un système de recherche d'information multimédia. Dans notre approche, nous utilisons une représentation des caractéristiques symboliques et des caractéristiques numériques dans un même espace et détectons des relations sémantiques entre les éléments de l'espace pour produire à la fin un vecteur descripteur représentant le contenu de l'image. Notre approche est basée sur l'utilisation d'une collection d'apprentissage d'images. Cette collection permettra de découvrir les régularités qui seront utilisées ultérieurement pour indexer des nouvelles images.

Le reste de ce papier est organisé comme suit, dans la section 2 nous présentons les travaux connexes dans le domaine de recherche d'image par contenu. Nous proposons notre modèle d'indexation sémantique des images dans la section 3. La section 4 introduit les résultats attendus par notre approche, puis nous discuterons les avantages et limites de cette approche dans la section 5. Enfin, nous terminerons par la conclusion et nos perspectives dans la section 6.

2. Travaux Connexes

Avec l'explosion des données multimédia sur le web, et en particulier les images, la mise en place d'un système pertinent de recherche d'image devient une nécessité. On distingue deux approches dans la littérature pour la recherche d'image : l'approche basée sur l'annotation textuelle et l'approche basé sur le contenu.

– La première approche est la plus utilisée, et consiste à une indexation des images par le moyen de mots clés. Néanmoins, elle nécessite un effort considérable pour une bonne description de l'image et reste encombrante et limitée.

– Une deuxième approche est la recherche d'image par le contenu (RImC) qui consiste à une indexation des images par des attributs de bas niveau tels que la couleur, la texture et les formes (Cox *et al.*, 2000, Deselaers, 2003, Flickner *et al.*, 1995).

La RImC n'a pas montré son efficacité dans le domaine de la recherche d'information (RI). Un inconvénient majeur de cet axe est qu'il est sémantiquement très faible ce qui le rend non adapté a un besoin de RI. En effet, la plupart des systèmes de recherche d'images traditionnels, tel que Photobook ou QBIC (Flickner *et al.*, 1995), se limitent à une recherche en termes de similarité visuelle.

Le domaine de l'image mining tente de trouver des solutions à ce problème. Les travaux dans cet axe essayent d'attribuer une certaine sémantique aux connaissances extraites à partir des images (Lavrenko *et al.*, 2003, Rasiwasia *et al.*, 2007, Carneiro *et al.*, 2007, Barnard *et al.*, 2003, Cheng *et al.*, 2005). Cependant les approches proposées jusqu'à nos jours ne sont pas assez développées, puisque cet axe est confronté à plusieurs obstacles :

– Le niveau d'analyse croit considérablement en fonction des connaissances à extraire - voir Figure 1.

– Le fossé sémantique [1] et le fossé sensoriel [2] définit par Smeulders dans (Smeulders *et al.*, 2000).

– Les problèmes liés au traitement d'image (absence de méthode universelle pour la segmentation d'image, la détection du contour, la reconnaissance d'objets, etc.).

En vue d'une description sémantique des images, Ordonez et al. (Ordonez *et al.*, 1999) ont appliqué un algorithme de découverte de règles d'associations à des images de synthèse. Ces images ont une description très pauvre sous forme de "tâches" (traduction de blobs). Ils n'ont utilisé que deux propriétés des images : la couleur et la texture. Une tâche est donc une région de pixels connexes cohérente au niveau de la similarité de couleur et de texture. Ces taches sont obtenues en segmentant les images. Les règles d'associations sont exprimées entre des tâches des images. Les résultats de

1. The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.
2. The sensory gap is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene.

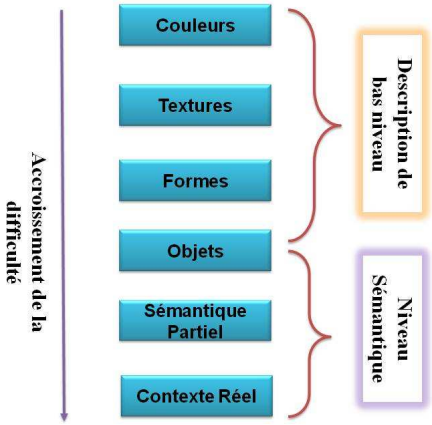


Figure 1. *Niveau d'analyse pour les systémes de RImC.*

Ordonez semblent prometteurs même si les propriétés auxquelles le mining a été appliqué sont assez simples, et donc pauvre sur le plan sémantique.

Une autre approche est de combiner des données textuelles avec des données visuelles de l'image pour les inclure dans un même espace. Un exemple de cette application est présenté dans (Parsons *et al.*, 2003) où les caractéristiques des images sont extraites sous forme de termes ou descripteurs (exemple "voiture rouge", "voiture non-rouge", "toits", "arbre", "route", etc.) pour être combinées avec les termes du texte qui accompagnent les images. Ces éléments sont représentés dans le même espace en utilisant la technique d'indexation sémantique latente (LSI). Les dimensions de l'espace vectoriel obtenu avec cette technique ne sont pas étiquetées ce qui les rend sémantiquement très pauvres et la segmentation difficilement interprétable.

Dans ce papier, nous nous concentrons sur le problème de découverte de connaissances à partir du contenu des images sous la forme d'associations entre des éléments symboliques (concepts) et des éléments de bas niveau.

3. Modèle proposé

Le principal objectif de ce travail est de développer une méthodologie pour la représentation et l'extraction des connaissances à partir des images en utilisant des techniques d'intelligence artificielle, afin de ramener la recherche d'image a une opération d'inférence utilisant les connaissances récoltées dans les images. La solution consisterait donc à décrire d'une façon automatique le contenu des images. Ceci, permettrait de dépasser le fossé sémantique, en d'autres termes le problème lié à la subjectivité des utilisateurs de la base.

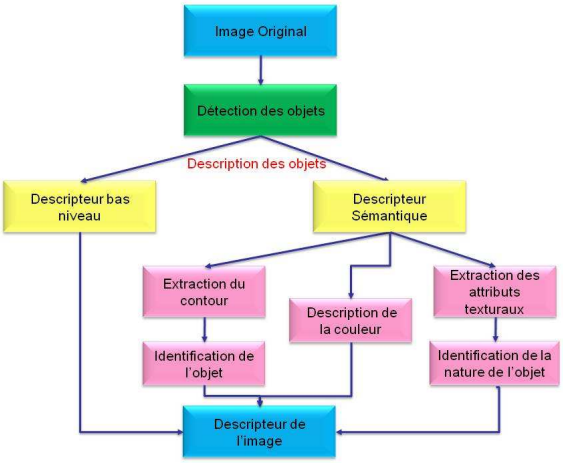


Figure 2. *Architecture du modèle proposé.*

Comme le montre la figure 2, le modèle que nous proposons utilise deux types de descripteur pour l'indexation des images : un descripteur sémantique qui permet la recherche d'image par la sémantique et un descripteur de bas niveau qui servira comme fonction d'appariement pour ordonner le résultat de la recherche. Le descripteur de bas niveau peut aussi être utilisé pour la recherche en termes de similarité visuelle ou lorsque l'utilisateur juge que le résultat de la recherche sémantique n'est pas satisfaisant.

Le principe de notre approche repose sur la description sémantique des objets contenus dans l'image. Pour aboutir au résultat souhaité plusieurs étapes sont nécessaires :

1) La première étape consiste à séparer les objets de l'image. Pour cela il est nécessaire d'utiliser un algorithme de segmentation d'image. L'algorithme de classification PFCM (Pal *et al.*, 2005) peut suffire pour obtenir un résultat satisfaisant.

2) La description de bas niveau de chaque région de l'image. Ce descripteur servira à établir un ordre de similarité pour le résultat de la recherche ou encore à faire une recherche d'image se basant sur la similarité visuelle. La méthode (Bannour *et al.*, 2009) peut être utilisée pour définir ce descripteur.

3) La description sémantique de chaque objet de l'image.

a) Afin d'identifier l'objet, il faut dans un premier lieu détecter le contour de l'objet (Canny, 1986), puis utiliser un algorithme d'apprentissage pour la reconnaissance de forme (Carpenter *et al.*, 1987). Ce descripteur a pour rôle de générer des labels comme : table, avion, personne, etc.

b) Nommer la couleur de l'objet (Mojsilovic, 2005). Ex. vert, noire, rouge, etc.

c) Pour identifier la nature de l'objet, on commence par extraire les attributs texturaux de l'objet (Haralick, 1979), puis on utilise un algorithme d'apprentissage pour la reconnaissance du modèle (Carpenter *et al.*, 1987). Ce descripteur a pour rôle de générer des descriptions comme : objet en bois, en marbre, en brique, etc.

4) Enfin, construire un vecteur descripteur de l'image qui va regrouper la description de bas niveau et la description sémantique de l'image.

4. Résultats

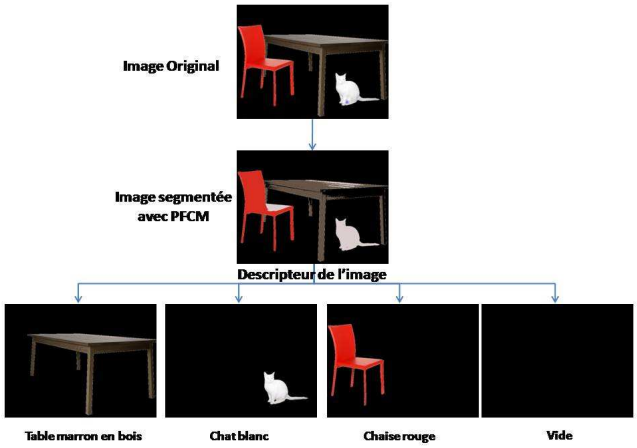


Figure 3. *Résultat de la description sémantique par le modèle proposé.*

La figure 3, illustre le résultat attendu par notre approche pour la description sémantique des images. L'indexation se base sur la description automatique des objets contenus dans l'image (appellation, couleurs et natures). La figure montre le résultat de la segmentation de l'image par PFCM, puis la description de chaque objet (Ex. objet 1 : la reconnaissance de la forme de l'objet a retourné le label "table", la reconnaissance de la texture a retourné la nature de l'objet "bois", l'identification de la couleur a donné la couleur "marron"). Une description détaillée du descripteur de bas niveau ainsi que les résultats obtenus avec est illustré dans (Bannour *et al.*, 2009).

5. Discussion

Actuellement le modèle que nous proposons est en phase de développement. Nous avons déjà implémenté la partie description de bas niveau, ainsi que la partie reconnaissance de forme pour l'annotation des images. Le descripteur de bas niveau, comme nous l'avons déjà indiqué, permet de jouer le rôle de fonction d'appariement lors de la

recherche sémantique d'image, il permet aussi la recherche d'image au niveau similarité visuelle. Les résultats obtenus pour la recherche visuelle des images sont reporté dans (Bannour *et al.*, 2009). Les résultats de la recherche sémantique basée sur la reconnaissance de formes seront bientôt publiés.

Dans le cas d'une recherche sémantique des images, le modèle que nous proposons permet de traiter trois types de requêtes : les requêtes par mots clés, les requêtes par croquis et les requêtes par l'exemple. Dans le cas de la recherche d'image par l'exemple, l'image exemple sera indexée et les mots clés obtenus seront utilisé pour être comparé à ceux des images de la base. Dans le cas d'une requête par croquis, la recherche sera basée sur les formes reconnus dans le croquis et dans les images de la base. Pour la recherche par mot clés, la requête sera comparée aux mots clés des images de la base.

Cependant, les performances de notre approche dépendent du résultat de la segmentation, mais aussi de celle de la reconnaissance des formes et de la reconnaissance du modèle de textures. Un autre problème auquel nous devons faire face, est l'insuffisance de la forme pour identifier certains objets. Par exemple la forme ne permet pas de reconnaitre des concepts comme "ciel", "mer", "herbes", etc., problème auquel on peut probablement pallier en se basant sur la texture aussi pour l'identification de certains objets.

Néanmoins, les premiers résultats que nous avons obtenus sur la base de COREL restent satisfaisants et promoteurs, d'autant que le modèle que nous proposons reste ouvert à plusieurs améliorations possibles. Par exemple, nous envisageons dans un futur proche d'enrichir la description des objets en définissant la relation entre eux. Une amélioration possible serait de définir des prépositions de lieu entre les objets d'une image : sur, sous, dans, devant, à côté de, entre, etc. (Ex. chat sous la table) ; ou aussi les subordonnées comparative : plus grand que, plus petit que, etc.

Une autre utilisation très intéressante de notre approche serait la génération automatique d'ontologies pour la description sémantique des images.

6. Conclusion

Un système de recherche d'image adapté aux besoins des utilisateurs doit être capable d'abstraction, abstraction par rapport aux simples pixels dont sont constituées les images, c'est-à-dire capable d'extraire de la sémantique d'image. Cependant, le fossé entre les attributs de bas niveau et la construction des connaissances sémantiques est le principal obstacle dans la construction d'une sémantique fiable pour la recherche d'image. Dans ce papier nous avons proposé une approche permettant de découvrir des informations sémantiques à partir des traits de bas niveau d'une image. Notre approche s'intéresse à la description sémantique des objets d'une image donnée. Comme perspective à ce travail, nous proposons de finir d'implémenter le modèle proposé, puis de l'enrichir en décrivant les relations entres les objets d'une image.

7. Bibliographie

Bannour H., Ayeb B., Hlaoua L., « Toward Content Based Image Retrieval : Global versus Local Image Description », CORIA 2009 : Sixième édition de la Conférence en Recherche d'Information et Applications, Belambra de la Presqu'ile de Giens, France, May, 2009.

Barnard K., Duygulu P., Freitas N., Forsyth D., Blei D., Jordan M. I., « Matching words and pictures », *Journal of Machine Learning Research (JMLR)*,, vol. 3, p. 1107-1135, 2003.

Canny J., « A Computational Approach to Edge Detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, n° 6, p. 679-698, 1986.

Carneiro G., Chan A. B., Moreno P. J., Vasconcelos N., « Supervised learning of semantic classes for image annotation and retrieval », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, p. 2007, 2007.

Carpenter G., Grossberg S., « A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine », *Computer Vision, Graphics, and Image Processing*, vol. 37, p. 54-115, 1987.

Cheng S. C., Chou T. C., Yang C. L., Chang H. Y., « A semantic learning for content-based image retrieval using analytical hierarchy process », *Expert Systems with Applications*, vol. 28, n° 3, p. 495-505, April, 2005.

Cox I. J., Miller M. L., Minka T. P., Papathornas T. V., Yianilos P. N., « The bayesian image retrieval system, pichunter : Theory, implementation, and psychophysical experiments », *IEEE Transactions on Image Processing*, 2000.

Deselaers T., Features for Image Retrievals, Diploma thesis, RWTH AachenUniversity, Aachen, Germany, 2003.

Flickner M., Sawhney H., Niblack W., Ashley J., Huang Q., Dom B., Gorkani M., Hafner J., Lee D., Petkovic D., Steele D., Yanker P., « Query by Image and Video Content : The QBIC System », *IEEE Computer Society*, vol. 28, n° 9, p. 23-32, 1995.

Haralick R. M., « Statistical and structural approaches to texture », *Proc. IEEE*, vol. 67, p. 786-804, 1979.

Lavrenko V., Manmatha R., Jeon J., « A model for learning the semantics of pictures », *in NIPS*, MIT Press, 2003.

Mojsilovic A., « A computational model for color naming and describing color composition of images », *IEEE Transactions on Image Processing*, vol. 14, n° 5, p. 690 - 699, May, 2005.

Ordonez C., Omiecinski E., « Discovering association rules based on image content », *In Proceedings of the IEEE Advances in Digital Libraries Conference*, Mai, 1999.

Pal N., Pal K., Keller J., Bezdek J., « A Possibilistic Fuzzy c-Means Clustering Algorithm », *Fuzzy Systems, IEEE Transactions on*, vol. 13, n° 4, p. 517-530, Aug., 2005.

Parsons O., Carpenter G. A., « ARTMAP neural networks for information fusion and data mining : map production and target recognition methodologies », *Neural Networks*, vol. 16, n° 7, p. 1075-1089, September, 2003.

Rasiwasia N., Moreno P., Vasconcelos N., « Bridging the Gap : Query by Semantic Example », *Multimedia, IEEE Transactions on*, vol. 9, n° 5, p. 923-938, Aug., 2007.

Smeulders A. W., Worring M., Santini S., Gupta A., , Jain R., « Content-based image retrieval at the end of the early years », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 12, p. 1349-1380, December, 2000.

Recherche contextuelle d'information dans un environnement mobile

Ourdia Bouidghaghen

IRIT-Equipe SIG/RI
118 Route de Narbonne
31062 Toulouse cedex 04
bouidgha@irit.fr

RÉSUMÉ. *La recherche contextuelle d'information (RCI) a pour objectif de mieux répondre aux besoins de l'utilisateur en lui délivrant l'information adaptée à son contexte spécifique de recherche. Cet article présente une approche de RCI dans le cas d'un environnement mobile, où le contexte spatio-temporel de l'utilisateur et son contexte cognitif, sont à la fois considérés pour lui délivrer de l'information pertinente. Nous proposons de construire des profils situationnels basés sur l'historique de recherche annoté par le contexte spatio-temporel pour personnaliser les résultats de recherche. De plus, le contexte spatio-temporel sera exploité pour mieux répondre à des requêtes sensibles au temps ou à la localisation.*

ABSTRACT. *The goal of contextual information retrieval (CIR) is to better meet the user's needs by delivering him information adapted to his specific search context. This paper presents a CIR approach for the case of a mobile environment where the spatio-temporal context of the user and his cognitive context are taken into account to deliver him some relevant information. We suggest building situational profiles based on the research history annotated by the spatio-temporal context to personalize the search results. Furthermore, the spatio-temporal context will be exploited to better address time or location sensitive queries.*

MOTS-CLÉS : *recherche contextuelle d'information, contexte mobile, profil situationnel, localisation, temps.*

KEYWORDS: *contextual information retrieval, mobile context, situational profile, location, time.*

Ourdia Bouidghaghen

1. Introduction

La surabondance de l'information et sa large accessibilité à travers le web, ont engendré la dégradation de la qualité des résultats retournés par les moteurs de recherche d'information (Dervin *et al*., 86), (Shamber, 94). En clair, le problème n'est pas tant la disponibilité de l'information, mais sa pertinence relativement à un contexte d'utilisation spécifique. C'est pourquoi les travaux en recherche contextuelle d'information (RCI) ont vu le jour ces dernières années (Ingwersen *et al*., 05), dans le but d'optimiser la pertinence des résultats de recherche, en impliquant deux étapes liées : définition du contexte du besoin en information de l'utilisateur, puis adaptation de la recherche en le prenant en considération dans le processus de sélection de l'information. Diverses taxonomies du contexte ont été proposées dans la littérature pour mieux comprendre les facteurs contextuels qui sont nécessaires pour une meilleure application du contexte dans les systèmes de RI (Göker *et al*, 08), et (Tamine-Lechani *et al*., 09) où une taxonomie est définie selon les 5 dimensions spécifiques du contexte qui ont été explorées dans la littérature de la RCI : la *dimension dispositif* (Göker *et al*, 02), la dimension t*âche/problème* (Jansen *et al.,* 07), la dimension *contexte du document* (Xie, 08), la dimension *spatio-temporelle* (Tao *et al*., 03), la dimension *contexte utilisateur* qui comprend à son tour deux sous dimensions : *le contexte social* (Smyth *et al*., 06) et *le contexte personnel* qui traite des sous dimensions du *contexte démographique* (Hupfer *et al*., 06), du *contexte psychologique* (Kim, 08) et du *contexte cognitif* (Timothy *et al*., 05), (Tamine *et al*., 08), (Daoud *et al*.,09).

Les recherches actuelles en RCI se sont focalisées sur la modélisation et l'exploitation du contexte cognitif, notamment les centres d'intérêt de l'utilisateur. Nous nous proposons de présenter dans cet article une approche de RCI qui intègre trois dimensions à la fois : les centres d'intérêt, la localisation et le temps. Cette orientation est en effet motivée par la prolifération de la technologie mobile (PDAs, téléphones mobiles, ...) qui a rendu l'accès à une grande masse de documents sur le web, possible à toute place et à tout moment. Ce nouveau cadre d'utilisation accentue le besoin et la nécessité de prendre en considération des informations du contexte pour améliorer la précision de recherche. En effet, vu les contraintes et particularités techniques des appareils mobiles (des difficultés de saisie de requêtes, zone d'affichage limitée, ...), nous assistons à une pratique de recherche différente de celle des requêtes traditionnelles de bureau. Des études sur les logs des requêtes des mobinautes (Kamvar *et al*., 07) montrent que les requêtes des utilisateurs sont plus courtes (donc plus ambigües), qu'il y a moins de requêtes par session et moins d'utilisateurs qui consultent plus loin que la première page des résultats. De plus, d'après les études dans (Sohn *et al*., 08), 72 % des besoins informationnels des utilisateurs mobiles sont liés à des facteurs contextuels notamment la localisation et le temps.

Dans ce cadre, notre objectif est de proposer des techniques permettant de supporter un processus de RCI dans ce nouvel environnement. Plus précisément, nous envisageons d'exploiter d'autres sources d'évidence que le contexte cognitif de l'utilisateur afin de mieux cerner les besoins spécifiques des utilisateurs mobiles. Pour cela, on vise à enrichir le contexte de recherche par des annotations du contexte spatio-temporel, notamment la localisation de l'utilisateur et le temps lors de son activité de recherche. Notre contribution consiste à exploiter à la fois le temps et la localisation de l'utilisateur avec différentes représentations possibles pour répondre à des requêtes sensibles à la localisation et/ou le temps. De plus, nous exploitons le contexte spatio-temporel pour caractériser des situations dans lesquelles se trouve un utilisateur interrogeant le système de RI. L'idée est de construire pour chaque situation identifiée, un profil regroupant les centres d'intérêt appris sur la base des documents consultés dans cette situation.

Cet article est organisé comme suit : la section 2 présente une synthèse de travaux connexes. La section 3 présente notre approche pour la représentation et l'exploitation du contexte dans un environnement mobile. On termine, en section 4, par une conclusion qui résume notre contribution et présente notre perspective de recherche.

2. Recherche contextuelle d'information dans un environnement mobile

Les moteurs de recherche traditionnels considèrent peu le contexte de la recherche et ne sont pas adaptés à l'environnement mobile. Des travaux récents tentent d'améliorer les performances de recherche dans cet environnement. Une première catégorie de travaux a abordé les questions liées à l'adaptation de la recherche aux contraintes imposées par les fonctionnalités limitées des appareils mobiles. Des approches sont proposées pour adapter de la visualisation de la liste des résultats (Sweeney *et al.,* 06) et pour faciliter la saisie des requêtes (Schofield *et al.,* 02).

Une autre catégorie de travaux a porté sur l'exploitation du contexte de l'utilisateur mobile pour améliorer la précision des résultats de recherche. Dans (Yau *et al*., 03) les auteurs appliquent des techniques de data mining sur un historique d'usage, composé de données du contexte, d'actions et de données liées aux actions, pour construire les profils d'un utilisateur qui reflètent ses comportements, ses intérêts et ses intentions dans chaque situation. Quand une requête est émise par l'utilisateur mobile, elle est interceptée et modifiée sur la base de ses profils et du contexte actuel.

Dans (Panayiotou *et al.,* 06) les auteurs exploitent l'importance du temps et de l'expérience (au travail, en vacances, etc) dans la personnalisation d'un portail de recherche de services web pour un utilisateur mobile. Ils proposent de construire un profil dynamique où les centres d'intérêt sont pondérés selon des zones temporelles apprises par l'étude de la routine journalière de l'utilisateur et ses activités dans

chaque zone. De plus, pour modéliser le changement des préférences de l'utilisateur selon ses expériences, l'association des poids d'importance aux concepts du profil est établie pour chaque nouvelle expérience de l'utilisateur. Ces profils basés temps évoluent et sont maintenus sur la base des feedbacks utilisateurs.

Dans (Hattory *et al*., 07) les auteurs traitent le problème de l'ambiguïté des requêtes en utilisant une méthode d'expansion basée sur la localisation. Leur méthode consiste a récupérer les coordonnées spatiales de l'utilisateur, les transformer en des mots contextuels (noms de places, d'activités liés à cette place) en utilisant un système d'information géographique et des techniques d'apprentissage des *weblog*, puis à pondérer ces mots contextuels sur la base de la comparaison de la probabilité globale du mot contextuel dans l'ensemble des documents du corpus cible et sa probabilité locale dans les documents retournés par la requête originale, en fin, la requête originale est étendue par les mots contextuels ayants les poids les plus élevés.

Dans (Ala-Siuru *et al*., 06) les auteurs présentent une méthode qui combine l'identificateur de la cellule GSM à laquelle est connecté l'utilisateur et les adresses MACs des dispositifs *Bluetooth* à côté, pour caractériser une situation de l'utilisateur mobile. Un raisonnement par cas, peut être conduit alors, sur la base du contexte courant et des situations apprises, pour inférer le profil adéquat du mobile.

Cet article présente notre approche pour une recherche d'information contextuelle dans un environnement mobile. En comparaison aux travaux précédents, notre approche peut être caractérisée par :

- la combinaison de l'adaptation à la localisation, au temps et aux centres d'intérêt de l'utilisateur.
- l'exploitation à la fois de représentations brutes (récupérées des capteurs mobiles) et sémantiques (des concepts d'ontologies) du contexte.
- la construction de profils situationnels à base de l'historique de recherche annoté par le contexte spatio-temporel.

3. Adaptation aux centres d'intérêt de l'utilisateur mobile, à sa localisation et au temps

3.1. *Approche générale*

Le schéma général de notre approche de RCI mobile est représenté dans la figure 1. L'utilisateur est engagé dans une situation (au travail, fait du tourisme,...) Lorsqu'il a besoin d'information, il soumet une requête à un moteur de recherche traditionnel, qui retourne une liste de documents. Pour adapter les résultats de recherche aux centres d'intérêt de l'utilisateur mobile, à sa localisation et au temps, un processus de contextualisation est mis en oeuvre, il consiste à construire une représentation de ces éléments contextuels de l'utilisateur et des documents. Puis un processus d'appariement permet de définir une fonction d'appariement pour chaque

type de contexte et de combiner les différents scores contextuels. Les documents seront réordonnés selon ce score contextuel en plus de leur score initial. L'utilisateur sélectionne les résultats qu'il juge pertinents pour sa situation, et l'historique est alors mis à jour. Dans la suite, nous explorons les éléments clés de notre approche.

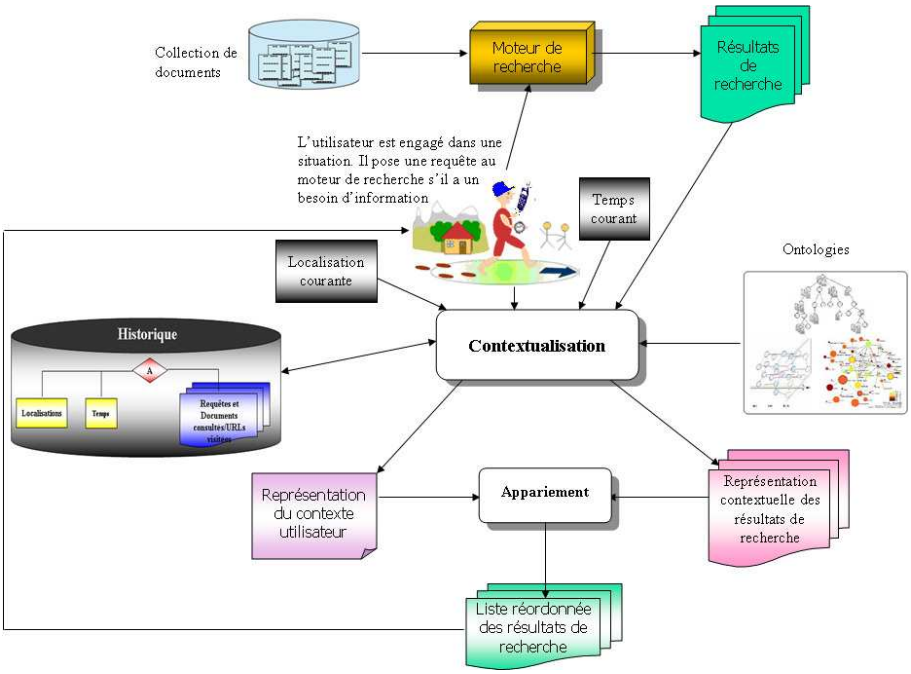


Figure 1. *Schéma général de notre approche.*

3.2. *Processus de contextualisation*

Il consiste à créer les représentations du contexte : centres d'intérêt de l'utilisateur, localisation et temps pour l'utilisateur et pour le document.

3.2.1. Les centres d'intérêt

Les centres d'intérêt de l'utilisateur ont été identifiés comme le facteur contextuel le plus important qui réduit la difficulté d'une requête ambigüe dans une tâche de recherche ad hoc (Park, 94). Dans (Tamine *et al.*,08) une distinction est faite entre des centres d'intérêt à long terme désignant les domaines d'intérêt généraux de l'utilisateur, et des centres d'intérêt à court terme reflétant un domaine d'intérêt spécifique relatif à une session de recherche. Notre but est d'affiner la notion de centre d'intérêt vers des besoins situationnels de l'utilisateur. En effet, la mobilité induit des besoins informationnels typiquement conséquents de la situation dans laquelle se trouve l'utilisateur. Nous pensons qu'une partie des informations qui

aident à clarifier une situation dont découle le besoin en information, sont les aspects contextuels de son moment d'émission, à savoir le temps et la localisation, exemple : "être la nuit à la maison", "l'été à la plage", etc. Cependant, il est clair qu'une situation ne se caractérise pas par des pures coordonnées géographiques et des points temporels. Par exemple la date n'est pas forcément l'attribut à utiliser pour identifier une situation, des attributs tels que le moment de la journée, le jour de la semaine, la saison, ...se montrent plus pertinents pour la caractérisation d'une situation (le même raisonnement s'applique à la localisation). C'est pour cela que nous proposons d'associer les informations du contexte récupéré par les capteurs du mobile à des concepts sémantiques (extraits d'ontologies temporelles et spatiales) pour pouvoir récupérer toutes les propriétés qui décrivent au mieux les facettes temporelles d'une date et d'une heure et les facettes spatiales des pures coordonnées géographiques. Une situation sera déterminée par une classification sémantique d'une agrégation de points du contexte spatio-temporel. La construction des profils se fera ensuite sur la base de l'historique des recherches attachées aux situations identifiées. Comme dans (Daoud *et al.*, 08), on se basera sur une représentation sémantique des profils par des concepts d'ontologie.

3.2.2. *La localisation*

Pour pouvoir représenter la localisation de l'utilisateur au moment de la requête et caractériser sa situation (nom et type de place), et de plus réaliser une indexation spatiale des documents, un modèle pour la représentation de localisations est nécessaire. L'information géographique peut être représentée à divers niveaux de granularité et sous diverses formes. Pour permettre une bonne représentation de l'information géographique et sa manipulation, la tendance est actuellement vers des approches sémantiques avec des ontologies spatiales. Comme dans (Chen *et al.*, 04) ou dans le projet SPIRIT [1], nous proposons de se baser sur une base de données spatiale et un thesaurus spatial pour représenter et raisonner sur les données géographiques.

3.2.3. *Le temps*

Pour définir les aspects temporels liés au temps de la requête et à la situation de l'utilisateur (matin, soir, *weekend*,...), mais aussi aux documents (heures et/ou dates de disponibilité), un modèle pour la représentation du temps est nécessaire. L'information temporelle est une information complexe, elle est continue et peut être représentée à divers niveaux de granularité. Pour permettre une bonne représentation de l'information temporelle et sa manipulation, la tendance est actuellement vers des approches sémantiques avec des ontologies temporelles. L'ontologie OWL-Time (Pan, 07) est aujourd'hui une référence pour la représentation et le raisonnement sur le temps.

---

1. Spatially-Aware Information Retrieval on the Internet:http://www.geo-spirit.org/index.html

3.3. Processus d'appariement

Une fois que le contexte est représenté, il est exploité dans le processus de RI afin d'améliorer la pertinence de l'information retournée. Notre exploitation consiste à réordonner les résultats de recherche en tenant compte de plusieurs scores contextuels: un score de personnalisation, un score géographique et un score temporel, en plus du score initial d'un document. A chaque fois que l'utilisateur soumet une requête au moteur de recherche, sa localisation et son temps courants sont récupérés. Ils seront utilisés d'une part, pour identifier la situation courante en vue de choisir le profil adéquat pour la personnalisation des résultats de recherche. Et d'autre part, pour calculer un score de pertinence géographique et/ou temporelle des documents retournés.

4. Conclusion et perspectives

Cet article présente une contribution à la définition d'un contexte de recherche dans le cadre d'un utilisateur mobile et son exploitation dans un processus de RI. Un aperçu des approches de la littérature en RCI et mobile est tout d'abord présenté. Ensuite, on a présenté des éléments de notre approche où les éléments du contexte considéré sont explicités avec leur exploitation dans un processus de RI. Nos perspectives de recherche portent en premier lieu sur l'adoption d'une stratégie de caractérisation de situations à partir du contexte spatio-temporel de recherche. En second lieu, sur la proposition d'une méthode de calcul et de combinaison des scores (initial, personnalisation, géographique et temporel) de pertinence des documents.

5. Bibliographie

Ala-Siuru P., Rantakokko T., «Understanding and recognizing usage situations using context data available in mobile phones», *ubiPCMM06, 2006*.

Chen H., Perich F., Finin T., Joshi A., «soupa: standard ontology for ubiquitous & pervasive applications», *Int. Conf. on mobile & ubiquitous systems: networking and services*, 2004.

Daoud M., Tamine L., Boughanem M., «Using a concept-based user context for search personalization», *In Proc of ICDMKE'08,* pages 293-298. IAENG, 2008.

Daoud M., Tamine L., Boughanem M., Chebaro B. «*A Session Based Personalized Search Using An Ontological User Profile*». In *ACM Symposium on Applied Computing (SAC 2009)*, ACM, p. 1031-1035, mars 2009.

Dervin B., Nilan M., «Information needs and uses», *ARIST, William, M.E. Eds*, p. 3-33, 1986.

Göker A., Myrhaug H., «User context and personalization», *in ECCBR workshop on Case Based Reasoning and Personalization*, Aberdeen, 2002.

Hattori S., Tezuka T., Tanaka K., «Context-aware query refinement for mobile web search», *Proc. of the 2007 International Symposium on Applications and the Internet Workshops.*

Hupfer M., Detlor B., «Gender and Web information seeking: A self-concept orientation model: Research Articles», *Journal of the American Society for Information Science and Technology 57(8),* 1105-1115, 2006.

Ingwersen P., Jarvelin K., «The Turn: Integration of information seeking and information retrieval in context», *Springer,* 2005.

Jansen B., Booth D., Spink A., «Determining the User Intent of Web Search Engine Queries», *Proceedings of the 16th international conference on World Wide Web,* 1149 -1150, 2007.

Kamvar M., Baluja S., «Deciphering trends in mobile search», *Computer40(8),* 58-62, 2007.

Kim K., «Effects of emotion control and task on web seraching bihavior», *Information processing and management 44(1),* 373-385, 2008.

Pan F., «Representing complex temporal phenomena for the semantic web and natural language», Ph.D thesis, University of Southern California, December 2007.

Panayiotou C., Samaras G., «Mobile User Personalization with Dynamic Profiles: Time and Activity», *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops,* LNCS 4278, pp.1295-1304.2006.

Park T., «Toward a theory of user-based relevance: A call for a new paradigm of inquiry». *J. Am. Soc. Inf. Sci.*, 45(3):135–141, 1994.

Schofield E., Kubin G., «On interfaces for mobile information retrieval», *Mobile HCI 2002*, LNCS 2411, pp. 383–387, 2002.

Shamber L., «Relevance and information behavior», *ARIST, William, M.E. Eds*,p. 3-48, 1994.

Smyth B., Balfe E., «Anonymous personalization in collaborative web search», *Information Rerieval 9(2),* 165-190, 2006.

Sohn T., Li K. A., Griswold W. G., Hollan J. D., «A Diary Study of Mobile Information Needs», *CHI 2008,* April 5–10, 2008, Florence, Italy.

Sweeney S., Crestani F., «Effective search results summary size and device screen size: Is there a relationship?», *Information processing and management*, 42(4): 1056-1074, 2006.

Tamine-Lechani L., Boughanem M., Daoud., «Evaluation of contextual information retrieval effectiveness : Overview of Issues and Research », *Knowledge and Information Systems Journal*, to appear 2009.

Tamine L., Boughanem M., Zemirli N., «Personalized document ranking: Exploiting evidence from multiple user interests for profiling and retrieval», *JDIM*, 6(5), 2008.

Tao Y., Mamoulis N., Papadias D., «Validity information retrieval for spatio-temporal queries», *Advances in Spatial and Temporal Databases: 8th International Symposium,* (LNCS 2750), 159-178, 2003.

Timothy M., Sherry C., Robert M., «Hypermedia learning and prior knowledge: domain expertise vs. system expertise», *J. of Computer Assisted Learning* 21(12), 53-64, 2005.

Xie H., «Users' evaluation of digital libraries (DLs): Their uses, their criteria, and their assessment», *Information processing and management 44(3),* 1346-1373, 2008.

Yau S., Liu H., Huang D., Yao Y., «Situation-Aware Personalized Information Retrieval for Mobile Internet», *In Proc. of the 27th Annual International Computer Software and Applications Conference*, p. 638–645, Nov, 2003.

Recherche d'information textuelle et phonétique pour le contrôle de l'étiquetage automatique d'émissions dans un flux télévisuel

Camille Guinaudeau

IRISA/INRIA
Campus de Beaulieu
35042 RENNES Cedex
France
Camille.Guinaudeau@irisa.fr

RÉSUMÉ. *En 2007, Naturel (Naturel, 2007) a proposé un système qui associe automatiquement une étiquette, c'est-à-dire un titre, à des émissions issues du découpage d'un flux TV. Cependant, ce système ne permet pas de vérifier la correction des associations étiquette-émission. Nous proposons dans cet article de contrôler cet étiquetage en nous basant sur les transcriptions textuelle et phonétique de la bande sonore contenue dans le flux. Nous montrons que des méthodes de recherche d'information permettent d'associer à chaque émission une description, issue d'un guide de programmes TV, description qui est ensuite comparée avec l'étiquette originale de l'émission. La technique proposée permet de contrôler un peu plus de 45% des émissions étudiées et de diminuer de nombre d'erreurs de l'étiquetage original de 3,5%.*

ABSTRACT. *In 2007, Naturel (Naturel, 2007) developed a method which, given a segmented video stream, associated a label with each segment. However, this method did not automatically check the accuracy of the results obtained. In this paper we propose to control these results, by taking each segment, and associating the corresponding phonetic or textual transcription of the soundtrack with descriptions extracted from a TV guide. Using techniques inspired from information retrieval methods, a description is linked to each segment, which can then be compared with the label associated by Naturel's method. This new method allows us to make a decision for 45% of the segments, and to lower the original labeling error rate by 3.5%.*

MOTS-CLÉS : *transcription automatique de la parole, recherche d'information textuelle, recherche d'information phonétique, multimédia, étiquetage des segments de flux TV*

KEYWORDS: *automatic speech recognition, textual information retrieval, phonetic information retrieval, multimedia, TV stream segments labeling*

Camille Guinaudeau

1. Introduction

Le nombre toujours grandissant de collections de documents télévisuels disponibles (et de documents multimédia en général) pose désormais le problème de la navigation dans ces flux, de leur interrogation, *etc.*, ce qui nécessite un accès à leur contenu sémantique, pouvant impliquer, dans un premier temps, leur structuration. Une des étapes de structuration consiste à découper un flux TV en émissions successives. Un tel flux est composé de deux types de segments : les inter-programmes (publicités, *jingles*, bandes-annonces), également notés IP, et les programmes (émissions de variétés, reportages, films, *etc.*). En 2007, Xavier Naturel (Naturel, 2007) a proposé une méthode permettant de découper automatiquement en programmes et inter-programmes les flux TV correspondant à plusieurs semaines d'enregistrement, et d'associer à chacun des segments obtenus une étiquette, c'est-à-dire une information sémantique qui se limite, dans ce travail, au titre de l'émission. Les IP étant généralement des programmes répétés, l'une des idées directrices de (Naturel, 2007) est d'utiliser ces répétitions pour réaliser la distinction entre les deux types de programmes. L'association d'une étiquette à chaque segment est ensuite faite grâce à un alignement entre un guide de programmes électronique, de type Télé Magazine –contenant les titres des émissions accompagnées de leurs heures de diffusion –et la segmentation obtenue. Cet appariement est mis en place par un algorithme d'alignement dynamique temporel (*Dynamic Time Warping* –DTW) qui calcule la distance entre la segmentation et le guide en prenant en compte la similarité de la longueur des segments ainsi que celle des horaires de diffusion. Cette tâche est complexe car un guide de programmes n'est pas complet. En effet, les IP n'y sont pas mentionnés, ainsi que certains programmes tels que la météo. De plus, des retards et des modifications de dernières minutes opérées par les chaînes par rapport au guide de programmes compliquent la tâche d'étiquetage. Les résultats de la segmentation sont globalement bons, mais, en ce qui concerne l'étiquetage, le processus associe parfois à plusieurs segments consécutifs une même étiquette. En étudiant ces cas de plus près, on constate qu'il ne s'agit pas d'un problème de sur-segmentation (où un programme serait découpé en plusieurs segments) mais bien d'un problème d'étiquetage. De plus, le processus d'étiquetage ne peut pas gérer les changements dans la grille de diffusion car il ne considère pas les informations sémantiques contenues dans les émissions mais uniquement leurs horaires de diffusion. Il est donc nécessaire de vérifier que les étiquettes liées aux segments correspondent aux programmes effectivement diffusés.

Dans cet article, nous proposons une méthodologie permettant de contrôler automatiquement, voire d'améliorer, l'étiquetage des segments proposé par (Naturel, 2007). Lors de notre contrôle, nous prenons en compte les informations de sens portées par les segments d'émissions. En effet, nous cherchons à caractériser le contenu des segments en travaillant sur les transcriptions –textuelles et phonétiques –de la bande sonore de ces segments, les paroles prononcées dans les programmes étant fortement représentatives de ce qu'ils renferment. Une telle caractérisation peut être mise en oeuvre de différentes façons. Certaines études, s'appuyant sur le fait qu'on utilise un certain vocabulaire pour parler d'un sujet puis que l'on change d'unités lexicales pour passer à un autre thème, se fondent sur la notion de cohésion lexicale. Par exemple, dans (Ferret *et al.*, 2001), les auteurs proposent un module qui extrait des signatures thématiques à partir de segments thématiquement homogènes obtenus préalablement. Ces signatures sont constituées de l'ensemble des mots maintenant élevé le niveau de cohésion lexicale et correspondent de ce fait à une représentation des thèmes abordés dans les textes. D'autres travaux mettent en place des méthodes issues de la recherche

d'information (RI). Dans (Lecorvé *et al.*, 2008) par exemple, les auteurs caractérisent les documents grâce aux mots dont le poids *tf * idf* est le plus important. Ce sont des méthodes similaires à ces derniers travaux que nous allons utiliser dans cet article.

La méthodologie proposée ici consiste à associer automatiquement à la transcription des segments résultant du travail présenté dans (Naturel, 2007) une description textuelle extraite d'un guide télévisuel grâce à des méthodes de RI. Le guide de programmes utilisé contient exactement les mêmes émissions que celui ayant servi lors de l'étape d'étiquetage dans le travail de (Naturel, 2007) ; il est cependant plus complet dans la mesure où chacun des programmes est associé à une description composée du titre de l'émission et, sauf exception, d'un résumé de son contenu qui peut aller jusqu'à 250 mots pour les plus longs. La description liée grâce à notre technique à chaque segment transcrit est ensuite comparée à l'étiquette fournie par Naturel afin de décider si cet étiquetage semble correct ou non. Lier transcription et description est toutefois difficile. En effet, les descriptions des programmes sont parfois très courtes et peu informatives (si limitées au seul titre de programme ou à un très bref résumé) et les transcriptions de la bande sonore des segments sont parfois très éloignées de ce qui a été prononcé dans la réalité. Le but de cet article est tout d'abord de montrer que des techniques de RI sont utilisables dans le contexte de la transcription de la télévision ; en effet, il n'existe pas, à notre connaissance, de travaux appliquant ces méthodes sur un tel matériau. Nous démontrons également qu'en adaptant des techniques de RI, il est possible de travailler sur du texte dégradé –les transcriptions de certains programmes télévisuels ayant un taux d'erreurs pouvant aller jusqu'à 80% –tout en obtenant des résultats encourageants. Nous réussissons en effet à contrôler l'étiquetage de près de la moitié des segments.

Nous décrivons dans un premier temps la méthode que nous avons mise en place pour contrôler l'étiquetage proposé par Naturel. Nous présentons ensuite les premiers résultats, obtenus sur les deux journées-test des 10 et 11 mai 2005, avant de conclure sur les travaux à mener afin de les améliorer.

2. Méthodologie

Le principe de la méthode que nous proposons pour contrôler l'étiquetage se décompose en deux étapes principales (*cf figure* 1), étapes présentées dans la suite de cette section. La première consiste à attacher à chaque segment reconnu dans le flux TV par Naturel une description issue du guide de programmes. L'association segment/description se base sur la similarité entre le contenu du segment, obtenu grâce aux transcriptions textuelle et phonétique de sa bande sonore, et celui de la description, et prend donc en compte des informations lexicales et sémantiques. Dans la seconde étape, la description associée par notre méthode est comparée, pour chacun des segments, avec l'étiquette proposée par (Naturel, 2007) afin de valider ou non cette dernière et éventuellement de la remplacer.

2.1. *Association segment/description*

L'association entre les segments de programmes et les descriptions est mise en place grâce à deux méthodes de recherche d'information appliquées respectivement sur les transcriptions textuelles et phonétiques de la bande sonore contenues dans les segments. Les résultats de ces deux recherches sont combinés *a posteriori*.

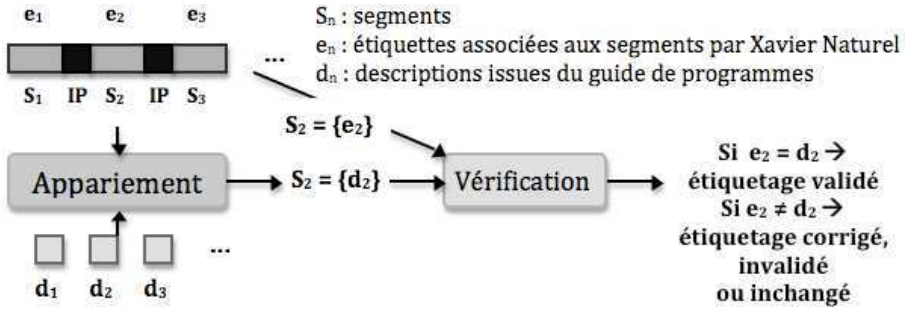


Figure 1. *Architecture globale de la méthode de contrôle de l'étiquetage*

2.1.1. *Recherche textuelle*

Pour lier descriptions et transcriptions textuelles des segments, ces deux types d'informations sont représentés, après lemmatisation, par des vecteurs de mots pondérés par un score *tf * idf*. La fréquence documentaire inverse est calculée à partir d'articles journalistiques extraits du journal *Le Monde* entre 1987 et 2003, soit 800 000 articles[1]. Une similarité entre les vecteurs représentant les descriptions et ceux représentant les transcriptions est calculée grâce à une mesure angulaire de type cosinus. La valeur de cette mesure est ensuite utilisée pour calculer, pour chaque couple segment transcrit/description, un score final qui prend en compte d'autres éléments que nous décrivons dans la suite de ce paragraphe. Notre méthode présente deux originalités : la première est la prise en compte, dans le calcul du poids *tf * idf* des mots contenus dans la transcription, de l'indice de confiance qui leur est associé. À chaque mot qu'il produit, le système de transcription automatique de la parole joint, en effet, une valeur qui traduit la confiance qu'il porte en sa transcription. Ainsi, dans notre technique, les mots ayant un indice de confiance faible sont discrédités par le biais d'un système de malus appliqué à leur poids suivant l'idée de (Lecorvé *et al.*, 2008). La seconde originalité est que l'ensemble des transcriptions des programmes est tour à tour considéré comme un ensemble de documents et un ensemble de requêtes. De la même manière, les descriptions de programmes sont vues soit comme des documents, soit comme des requêtes. Nous obtenons ainsi pour un segment transcrit s la liste classée, par ordre décroissant de similarité, de toutes les descriptions possibles, et pour une description d la liste classée, par ordre décroissant de similarité, de tous les segments possibles. Ces deux « passes » se justifie dans le calcul du score final: nous souhaitons, en effet, que pour chaque couple segment/description, la description soit celle qui corresponde le mieux au segment et *vice versa* ; or le score associé aux couples s/d et d/s n'est pas symétrique car il prend en compte la valeur du cosinus mais également la position du couple dans la liste résultat et la différence entre le score de ce couple et celui du couple suivant (*cf.* (Guinaudeau, 2008) pour plus de détails). Pour finir pour chaque segment s, l'association retenue –c'est-à-dire celle pour laquelle la description d cor-

1. Bien que le vocabulaire de ce corpus soit parfois assez éloigné de celui prononcé dans les émissions télévisées –ce qui influence sans doute la valeur des scores *tf * idf* –le choix de son utilisation s'explique par le fait qu'il a été employé lors de l'apprentissage du modèle de langue du système de transcription. Les mots qu'il contient sont donc reconnus par le système.

respond le mieux au programme contenu dans le segment –est celle pour laquelle le score final, égal à la somme des scores des couples s/d et d/s, est le plus élevé.

2.1.2. *Recherche phonétique*

Cependant, comme le montrent (Cardillo *et al.*, 2002, Logan *et al.*, 2002), appliquer une recherche textuelle seule pour explorer des transcriptions de documents sonores se heurte au problème des mots hors vocabulaire. En effet, les systèmes de reconnaissance automatique de la parole utilisent, en règle générale, pour produire des transcriptions, un dictionnaire phonétique qui permet d'associer des mots aux sons entendus. Ils ne peuvent donc pas retourner des termes n'appartenant pas au dictionnaire. Or ces mots dits hors vocabulaire peuvent posséder un fort contenu sémantique. C'est en particulier le cas des noms propres, souvent absents des dictionnaires. Nous choisissons donc de traiter différemment les descriptions issues du guide télévisuel contenant des noms propres, repérés grâce à l'outil LIA_PHON (Béchet, 2001), de celles n'en possédant pas, qui ne subissent, elles, que le seul dispositif explicité ci-dessus.

Pour chaque description d'émissions contenant des noms propres, nous mettons tout d'abord en place une recherche phonétique permettant de chercher à repérer, à l'intérieur de la bande sonore d'un segment, les noms propres présents dans cette description. Le système de reconnaissance de la parole produit, dans son processus de transcription automatique de la bande sonore, une transcription phonétique qui traduit sous forme de phonèmes les sons prononcés dans l'émission. Les noms propres, quant à eux, sont phonétisés à partir des descriptions grâce à l'outil LIA_PHON qui transforme chaque mot de la description en suite de phonèmes par un système de règles. La recherche phonétique permet de retrouver une petite suite de phonèmes –ici un nom propre –dans une séquence de phonèmes plus grande –la transcription phonétique de la bande sonore contenue dans le segment. Nous ne cherchons toutefois pas à repérer dans la transcription phonétique exactement la suite de phonèmes qui constitue le nom propre. En effet, les noms propres n'apparaissant pas dans le dictionnaire du système de reconnaissance de la parole, ils font l'objet d'erreurs de transcription. De plus, la transcription phonétique de la bande sonore est perturbée par les marques d'hésitation ainsi que les accents des locuteurs, ce qui modifie la prononciation des noms propres recherchés. Notre but est donc de rechercher dans la transcription phonétique la séquence de phonèmes qui minimise le plus la distance avec la phonétisation du nom propre. Cette distance est calculée grâce à une adaptation de la distance d'édition qui autorise le nom propre phonétisé à être situé n'importe où dans la transcription phonétique (*cf.* (Muscariello *et al.*, 2009) pour plus de détails sur la distance d'édition utilisée). La méthodologie que nous proposons est la suivante : pour chacun des noms propres contenus dans une description, nous récupérons, pour tous les segments de programmes, le coût minimal engendré par la transformation de la chaîne de phonèmes correspondant au nom propre phonétisé en une partie de la chaîne de phonèmes correspondant à la transcription phonétique de la bande sonore du segment. Ce score minimal est ensuite additionné avec les scores obtenus pour chacun des noms propres de la description. Cependant, les descriptions qui contiennent des noms propres possèdent également de nombreux autres mots tout aussi porteurs de sens. C'est pourquoi nous appliquons aussi sur ces descriptions, parallèlement à la recherche phonétique, une recherche textuelle identique à celle employée pour les descriptions ne contenant pas de noms propres.

2.1.3. *Combinaison des deux types de recherche*

Finalement, pour chacun des couples segment/description, les scores de ces deux recherches sont combinés *a posteriori* –celui issu de la recherche phonétique et additionné à celui obtenu par les deux « passes » de la recherche textuelle multiplié par un facteur 200 –et les paires segment/description retenues sont celles dont le score global est le plus élevé.

2.2. *Vérification de l'étiquetage de Xavier Naturel*

La phase d'appariement expliquée ci-dessus, appliquée à un corpus de deux jours d'enregistrement de télévision, nous permet d'associer une description à 60 segments sur 133. Nous souhaitons, dans cette seconde étape, contrôler automatiquement la qualité de l'étiquetage proposé dans (Naturel, 2007). Pour cela, nous comparons pour chaque segment du flux télévisé l'étiquette fournie par Naturel avec la description liée à ce segment par notre technique. Si le titre contenu dans notre description et l'étiquette correspondent, l'étiquetage est considéré comme correct. Si, au contraire, la description que nous avons attachée à un segment est différente de l'étiquette de Naturel, nous comparons alors les horaires de début de diffusion correspondant aux deux programmes désignés respectivement par l'étiquette et la description avec l'heure du début du segment. Si l'heure de début du programme désigné par l'étiquette est la plus proche de celle de début de diffusion du segment, l'étiquetage est considéré correct, sinon il est dit faux et on remplace l'étiquette par notre description. Cependant, si la différence entre l'heure de début du programme désigné par l'étiquette ou par la description et l'heure de début de diffusion du segment est supérieure à une demi-heure, on considère que l'étiquetage est faux sans proposer de nouvelle étiquette.

3. Résultats

Les résultats que nous fournissons ici concernent globalement tant la méthode de recherche textuelle « pure » que celle « hybride », combinant recherche phonétique et textuelle pour les descriptions contenant des noms propres. Notre technique de vérification nous permet de contrôler automatiquement l'étiquetage d'un peu plus de 45% des segments repérés par Naturel dans les deux journées-test. Ce pourcentage peut s'expliquer par le fait qu'environ 30% des segments de notre corpus contiennent des programmes –tels que la météo ou les programmes interstitiels [2] –qui ne possèdent pas de description dans le guide des programmes. Nos résultats se répartissent comme présentés dans la figure 2.

Si l'étiquetage de seulement 45% des segments a pu être contrôlé par notre méthode, ces résultats sont tout de même prometteurs. En effet, des expériences successives nous ont montré que les paires segment/description fournies par la combinaison étaient un peu meilleures que celles fournies par une recherche textuelle seule. D'une part, le nombre de couples pertinents retournés augmente légèrement (plus 2 pour les deux journées-test). D'autre part, le nombre de fausses alarmes, c'est-à-dire d'asso-

2. Les programmes interstitiels sont des programmes d'une minute environ tels que « Un jour, un arbre » ou « Conso mag ».

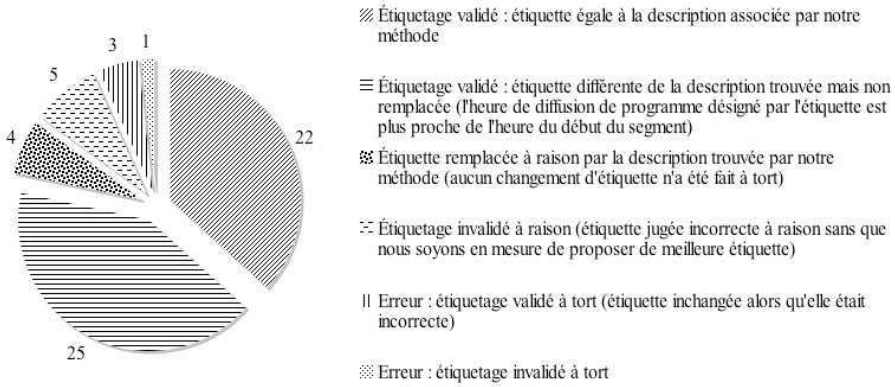


Figure 2. *Résultats de la vérification des étiquettes fournies par Xavier Naturel*

ciations considérées à tort comme pertinentes, par rapport au nombre d'associations pertinentes trouvées diminue pour les deux journées. Cependant ces résultats sont encore perfectibles. Nous étudions dans la partie suivante les différentes techniques que nous pourrions mettre en place afin d'accroître les performances obtenues.

4. Discussions et perspectives

La méthodologie de contrôle que nous avons proposée et qui s'avère conduire à des résultats encourageants repose sur une combinaison originale de techniques de RI textuelle et phonétique, et tire d'ailleurs profit de cet assemblage. Il reste cependant à étendre la couverture de notre technique à tous les programmes puisqu'elle ne contrôle actuellement qu'environ la moitié des segments produits par (Naturel, 2007). Pour ce faire, diverses pistes sont envisageables.

Une première perspective, à court terme, consiste à faire fonctionner notre méthode sur un corpus plus étendu. En effet, nous constatons de grandes disparités au niveau des résultats sur les deux journées des 10 et 11 mai 2005. Travailler sur un corpus plus important nous permettrait d'avoir une vision plus complète des performances de notre système et de traiter un ensemble plus large de types d'émissions. En plus du corpus de mai 2005 constitué de trois semaines d'enregistrement segmentées et étiquetées, un nouveau corpus de 6 mois d'enregistrements en continu des chaînes TF1 et France 2 vient d'être construit à l'IRISA.

Nous devons également travailler sur une catégorisation automatique des programmes télévisuels. Nous avons, dans cet article, proposé deux modes de recherche différents en fonction de la présence ou non de noms propres dans les descriptions mais nous n'avons pas pris en compte les disparités au niveau des émissions. Or, l'utilisation de méthodes différentes, adaptées à chacun des types d'émissions, nous permettrait sans doute de contrôler l'étiquetage d'un plus grand nombre de segments.

Certains travaux, (Fischer *et al.*, 1995) par exemple, proposent des techniques permettant de décider de la classe d'un programme – météo, journal télévisé ou sport

–en se basant sur la longueur des scènes, la couleur dominante du fond d'écran, la pureté des silences, *etc*. En utilisant des indices vidéo et audio, nous pourrions ainsi améliorer les résultats du système en utilisant, pour chaque programme, une méthode plus adéquate.

Enfin, un dernier élément qui explique peut être partiellement nos résultats est que nous supposons dès le départ que la segmentation de Xavier Naturel est correcte, ce qui se révèle assez juste de façon générale ; cependant certains segments contiennent plusieurs programmes et donc généralement plusieurs thèmes, ce qui gêne le calcul de similarité entre les descriptions et la transcription textuelle de ces segments, car les divers thèmes se parasitent dans la représentation à l'aide de vecteurs de mots pondérés. Nous pourrions donc réfléchir à l'utilisation d'une méthode étudiant la cohésion lexicale du segment afin de le découper en sous-parties si cette cohésion passe sous un seuil critique et de caractériser son thème en utilisant les termes qui ont permis de maintenir cette cohésion. Ceci suppose toutefois que les sous-parties soient suffisamment longues pour que l'on puisse effectivement distinguer une cohérence au niveau du vocabulaire.

Le découpage en émissions est une étape essentielle à la structuration de flux télévisés mais n'est cependant pas suffisant à la navigation à l'intérieur de ceux-ci. Notre objectif, à plus long terme, est de regrouper les thèmes similaires à travers les émissions et les chaînes, dans le but de pouvoir faire une recherche dans le flux vidéo sur un sujet particulier, ou de mener une étude comparative du traitement d'un même sujet entre différentes chaînes. Un prolongement de notre travail consistera donc à mettre en place une segmentation et un étiquetage des émissions en sous-sections abordant plusieurs sujets –les différents reportages d'une émission d'investigation par exemple.

5. Bibliographie

Béchet F., « LIA_PHON : un système complet de phonétisation de textes », *Traitement automatique des langues*, vol. 42, n° 1, p. 47-67, 2001.

Cardillo P. S., Clements M., Miller M. S., « Phonetic Searching vs. LVCSR : How to Find What You Really Want in Audio Archives », *International Journal of Speech Technology*, vol. 5, n° 1, p. 9-22, 2002.

Ferret O., Grau B., « Utiliser des corpus pour amorcer une analyse thématique », *Traitement automatique des langues*, vol. 42, n° 2, p. 517-545, 2001.

Fischer S., Lienhart R., Effelsberg W., « Automatic Recognition of Film Genres », *3rd International ACM Conference on Multimedia*, 1995.

Guinaudeau C., « Contrôle automatisé de contenu télévisuel », 2008. Rapport de stage de Master 2 recherche de l'université de Caen Basse Normandie, Caen, France.

Lecorvé G., Gravier G., Sébillot P., « On the Use of Web Resources and Natural Language Processing Techniques to Improve Automatic Speech Recognition Systems », *6th International Language Resources and Evaluation*, 2008.

Logan B., Moreno P., Deshmukh O., « Word and Sub-word Indexing Approaches for Reducing the Effects of OOV Queries on Spoken Audio », *2nd International Conference on Human Language Technology Research*, 2002.

Muscariello A., Gravier G., Bimbot F., « Variability Tolerant Audio Motif Discovery », *International Multimedia Model Conference*, 2009.

Naturel X., Structuration automatique de flux vidéos de télévision, PhD thesis, Université de Rennes 1, France, 2007.

Aggregated search: From information nuggets to aggregated documents

Arlind Kopliku

Institute de Recherche en Informatique de Toulouse, UMR 5505 CNRS, SIG-RFI
118 route de Narbonne F-31062 Toulouse Cedex 9
Arlind.Kopliku@irit.fr

ABSTRACT. *The aggregated search assembles in one interface information from different sources. It deals with different types of content (text, video, image, etc) and granularities of retrieval. It aims to assemble the retrieved content forming an aggregated result. This is in contrast with the common approach which provides a list of documents as the search result.*
Today we are able to retrieve content of different types and granularities, but little work has been done for their aggregation. Being a new area of research formalization of aggregated search is still missing. This paper treats the problem in a high level of abstraction. It presents the state of the art and decomposes the problem, listing issues and providing examples and formalization. This work aims to be a base of reflection and a reference for future work.

RÉSUMÉ. *Le but de la recherche agregée est de rassembler des informations provenant de plusieurs sources en une seule interface. Elle doit ainsi gérer des problématiques liées aux différents types de contenu (texte, vidéo, image, etc) ainsi qu'à la granularité des résultats. La formation d'un contenu agrégé à partir de différents types de contenus retrouvés contraste avec l'approche commune en RI consistant à renvoyer à l'utilisateur une liste ordonnée de résultats. Si nous sommes aujourd'hui capables de retrouver de l'information de différents types et de différente granularité, très peu de travaux existent concernant leur agrégation. La recherche agrégée étant un domaine de recherche récent, elle manque encore de formalisation. Ce papier se propose de traiter la recherche agrégée à un niveau d'abstraction élevé. Il présente tout d'abord l'état de l'art, puis décompose le problème en listant et formalisant les différentes problématiques. Ce travail doit servir de base de réflexion et de référence pour de futurs travaux sur le domaine.*

KEYWORDS: *aggregation, aggregated search, unified search*

MOTS-CLÉS : *agrégation, recherche agrégée, recherche unifée*

1. Introduction

Aggregated search comes in contrast with classical search paradigm, where search systems list a number of documents as an answer to a free language query. In the latter, the user has to scan the returned list and search within one or some of the retrieved documents. This can be time consuming especially if the needed information is only a small portion of a document or when the needed information resides in more documents. Aggregated search proposes fictitious built documents which should contain, organize and connect useful information.

Today we are able to retrieve information of different types and at different granularity such as paragraph, chapter, document, etc, but there is little work providing means to combine them. The information need can be composed of sections of text belonging to different sources as well as it can contain some images, videos, etc. Aggregated search tries to identify the necessary content, organize it and present it to the user in such a way to facilitate his information search. Aggregation can also provide a richer view of the existing information.

Existing solutions are very limited. However, some search engines have already started to merge information of different types into the main result search page. The contents are simply listed or placed into fixed visualization spaces. To our knowledge, scientific publications are even more limited. Most of the work is at the proposal level or it is about specific issues in specific contexts.

Our aim in this paper is not to provide a solution, neither to list existing ones. Aggregated search is very recent and needs formalization. Thus, we aim to present the problem in general and decompose it, listing the main issues. This way we provide a mean of reflection and reference for future work.

This paper is organized as follows. Section 2 describes the state of the art and why a lot of work is to be done. Section 3 presents the aggregation search problem. It describes the aggregation process providing the necessary phases and it provides examples and reflections on the types of aggregation. Section 4 is about conclusions and future work.

2. State of the art

Aggregated search intersects many areas. It starts from the retrieval of contents and it ends with their filtering, selection and organization. A lot of work exists on the retrieval side. But content aggregation has not been largely studied. A lot of work is to be done and formalization is almost missing.

A good study starting point can be focused retrieval (Fuhr *et al.*, 2008). Focused retrieval deals with the granularity issue. Aggregated search could use it to provide the input to assemble and organize. In fact, in the XML retrieval context it has been shown than returning several elements together trigger a stronger user satisfaction than returning a single element. An interesting case study comes from the INEX Relevant in Context task(Fuhr *et al.*, 2008). Here, instead of returning elements

separately, relevant elements are grouped by document. Still, it does not consider grouping from different sources.
During the ACM SIGIR 08 conference (Lalmas *et al.*, 2008), a workshop was held especially for aggregated search. Sushmita, Lalmas, Tombros (S.Sushmita *et al.*, 2008) propose to visualize as search results, digest pages which are thought as fictitious documents built from clustering the documents returned by a search engine. Some others focus in specic domains such as social science and medicine (Ou *et al.*, 2008; Wan *et al.*, 2008). The first one considers a collection of social science articles. It extracts and organizes important concepts. The second article focuses on the importance of result organization to the user utility.
The industry already comprises aggregated search features. We can find it in specific contexts such as product search or location search. For example, *Wize.com* offers product search with results that are obtained as an aggregation from several sources (Shilman, 2008). Google's local search[1] adds phone numbers, images and web pages when available in addition to the map result.
Aggregation seems to be the trend in web search, too. This trend is often referred as *unifed search* , but other names are in use such as *blended vertical search* or *universal search*. The new approach adds to the list of web pages presented in the main search results page vertical content such as maps, images, news, etc. There are two main approaches at this moment. In the first one, the content appears inline with the HTML listings and all of it is ranked according to a scoring algorithm. Google Universal search reflcts this approach. The other approach involves a new search results layout with standardized "holes" for each type of content. *Ask3D* [2] , *Kosmix* [3], *Yahoo's Alpha*[4] and *Google's SearchMash*[5] are all taking this approach.

Existing approaches remain very limited. In general, almost all solutions predefne the way the content should be organized: some use predefned content placement and some other simply order by relevance. Relations between the retrieved contents are not considered. However, sometimes some information has to be shown before another even if it is less relevant (logical connection, chronological order, etc). Moreover, some contents should be grouped together (similar content, alternative lecture, A explains B, A is a photo of B, etc). Studying relationship between results would not only help better visualize the contents, but also provide supplementary useful information which is not deemed as relevant in the beginning.

1. http://maps.google.com
2. http://www.ask.com
3. http://www.kosmix.com
4. http://au.alpha.yahoo.com/
5. http://www.searchmash.com

3. From information nuggets to aggregated documents

3.1. *Definition*

In the ACM SIGIR 2008 workshop on aggregated search(Lalmas *et al.*, 2008) the following definition was given:

Definition *Aggregated search is the task of searching and assembling information from a variety of sources and placing it in a single interface.*

In fact, the user might be satisfied with a part of a document or some parts. Sometimes, his information need might be the composition of some sections from different documents. He might also want some images, videos, etc. Let's consider an example with the query "Pink Floyd". The classical ad-hoc approach would list a list of web pages, probably repetitive and partially relevant. Google's universal search would list contents of different types augmenting the diversity of information. Aggregated search would try to deal not only with redundancy and diversity but it would also organize the information. It can start with a description of the group, general data, some images, videos, albums, etc. Figure 1 better illustrates the approaches.

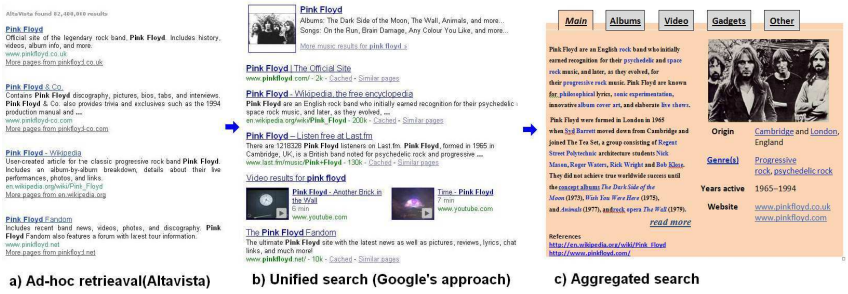


Figure 1. *Comparing search paradigms assuming the query "Pink Floyd" for all three cases. The aggregated result is hand built.*

Aggregated search does not look for single documents. It tries to build them. It merges content together. We will define the aggregated document as its search objective.

Definition *An aggregated document is the result of assembling information from a variety of sources.*

If aggregated documents are built by aggregation, we are interested in their components. We will categorize these components by type. The term nugget will represent a component of a certain type. The following definition better describes it.

Definition *The information nugget is a unit of information of one type of content (text, image, video, news, etc).*

In the context of textual information, it can represent a partition of the text. In the

case of images, it is the image itself. In the case of video, it can be the entire video or a part of it. It is clear now that we want to build aggregated documents assembling information nuggets.

3.2. *Aggregate structure*

Aggregated search has to deal with the organization of relevant content. It is not enough to gather the information but it is also fundamental to organize it for its final visualization to the user. We would like to define as aggregate structure all the information which describes the content, the order of visualization,and the preferences in visualization of the aggregated document. We want to keep out the aesthetic considerations such as background color or font size.
Definition *The aggregate structure describes the content and visualization preferences of the aggregated document. Stylistic considerations (such as background color, font size) are left out.*
The following examples illustrate some partial structure of some aggregated information:

–3 images, 2 videos

–a list of 4 reviews, 3 ratings

–passage A, passage B, passage C, visualization order A,B,C

In ad-hoc information retrieval the answer to a query is a list of documents. In aggregated search it should be possible to add aggregate structural information to the query. Someone could ask for images and videos only, or reviews and ratings only. But user studies (Nielsen, 2003) tell users are generally lazy. Though, they do not use additional options. Nevertheless, this can be very useful in times. Imagine a travel service in the web who wants to search for hotels and it is interested in hotels with at least 3 photos, 1 map, the address, phone number, number of stars, reviews, etc.

3.3. *Abstract phases*

In this section we define 3 abstract components of the aggregated search. They do not have to be implemented separately. They can be merged and ordered in many ways. Nevertheless, we consider they are part of the process. We present a selection, a filtering and an organization phase.
The **selection phase** is about selecting the information which is potentially useful. There are many possibilities. The list of the K top documents returned from a search engine is one choice.
The **filtering phase** is about filtering useless information. The selection and filtering phase correspond to the intuition that the aggregated search might show the most useful information and the less useless information. Filtering can be applied before selection. But, it seems more plausible placing it after selection or merging them together. Filtering could remove information redundancy. It could also deal with ads.

The aggregated search allows assembling information coming from different sources. Because the aggregated result components can be parts of documents, we loose the original structure. Here comes the **organization phase**. It is necessary to relate the retrieved content. The retrieved content has to be organized in order to be visualized in a sensed manner. Some content is more sensed to be listed; some has to be grouped and so on. Organization information can be also gathered at selection or filtering time.

3.4. *Types of content*

It is important to distinguish between different types of content because the retrieval and the aggregation process depend strongly on the types of content. There are many choices. Some types include some others and some times there is intersection. Below there is a brief list of types of content:

By type of media: text, video, image

Units of information: book, article, chapter, title, web page, word

Contextual use: blog, review, rating

Field: field, email, phone number, address

Link: anchor, hyperlink, reference

We can distinguish between media types such as text, video, image and hypertext. But we can also drill into different granularities within the same media such as book, chapter and paragraph in text. The type of information can be used in some cases such as for news, reviews, comments, etc.
We also define a special type of content field. Fields are composed of the filed name and the field value. Emails, phone numbers are examples of fields.

3.5. *Aggregation*

The aggregated result can be considered as composed of information (set of nuggets) and organization (aggregate structure). It is necessary though to be able to relate the retrieved content for the visualization. We will define first a list of logical relations, which can later be combined for visualization. Below, there is a short list of relation which is not exhaustive.
Association: We have an association when a content is related with another. Because the final aim is visualization, this property should reflect the probability of two or more contents to appear together. Association should try to define almost certain

relations between two entities like for example: this map is about this hotel.
Group: Some items share some properties. These properties can be used to group the content. We can put in the same group information of the same context. We can also group based to the type of content. Grouping can be done at different levels. It can be used for visualization, but as for association it does not force a specific visualization.
Order: Some content should be ordered. The ordering criteria can be various. We can order some content in chronological order, by relevance, etc. Relations of order give a preference to one item with respect to other items.
We can now define the most common types of visualization and how the logical relations can be used here. Visualization relations are part of the aggregation structure. They define visualization preferences. They can also be merged with each other.
List: Lists are a frequent type of visualization. The content here is simply listed in a consecutive order. The items might be ordered by some property or be presented randomly. Depending on the case, we can use the order or group logical relations. The listing can be vertical or horizontal.
Table: The tables are often used for visualization. They can be used for some logical organization of the data as well as for aesthetic considerations.
Block: We define as a block the visualization of content within a rectangular surface. The structure within can be a table. Blocks can be built using the group logical relation as well as some other properties. We can put within the same block information of the same context or same type of content, etc. Blocks can also be nested together or listed.
Links: Links can contain a brief description of a content and a link. They are very useful in information retrieval.
Menu: Menus are linked content by a list of links. Menus are useful to organize the visualization space. They can be viewed as linked blocks.
Partial inclusion: Sometimes an information can be very long. We might want to partially introduce it. This way the user can decide if reading more or not. A *"read more"* option can be provided or a link to the source.
Summary: A summary is a shortened version of the original source. The main purpose of such a simplification is to highlight the major points from the genuine (much longer) subject, e.g. one or more documents, a movie, an event, etc.

4. Conclusions and future work

This paper addresses aggregated search as a prominent area of interest. It presents the problem in general and then decomposes it trying not to lose generality. We provide phases and considerations necessary to build such a system. Although things are moving at enterprise level, there is still few published scientific research addressing aggregated search directly. That is why this paper contributes by presenting the problem definition, the actual state of art and identifying some of the issues and subproblems.

The paper starts at a high level of abstraction then it drills down into lower level. It proposes three abstract phases as essential components of aggregated search namely selection, filtering and organization. But, no constraints and specific implementations are proposed. There are in fact many possible solutions. Proposing a part of them is not the goal of this article which aims generality.
Aggregated search arises from the availability of information of different types and sources. Different types of content are retrieved and assembled differently. Here, we present an overview of the possible aggregations and we provide some types of relations which can be used to organize the aggregated result. This should help to reflect on specific solutions.
To solve the aggregated search problem, the following research issues can be considered. Focused retrieval and vertical searches can be used to feed the input to the system. Further filtering can involve redundancy removal, such as near-duplicate detection. Query interpretation is essential to understand the user need, but it can also indicate some aggregation hints. In order to aggregate the retrieved content, it is also necessary the study of the relations between content as well as the study of intelligent content organization (placement) algorithms. Finally, evaluation is also an open issue. User studies can be used but they are time consuming and not preferable. Devising a good evaluation system and realizing evaluation benchmarks are important challenges for the domain.

5. References

Fuhr N., Kamps J., Lalmas M., Malik S., Trotman A., "Overview of the INEX 2007 Ad Hoc Track", pp. 1–23, 2008.

Lalmas M., Murdock V. (eds), *SIGIR Workshop on Aggregated Search*, ACM, 2008.

Manning C. D., Raghavan P., Schutze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Nielsen J., *Designing Web Usability*, 9th edn, Dwyer, David, Indianapolis (USA), 2003.

Ou S., Khoo C. S. G., "Aggregating search results for social science by extracting and organizing research concepts and relations", *in* Lalmas *et al.* (2008), 2008.

Shilman M., "Aggregate documents: making sense of a patchwork of topical documents", *DocEng '08: Proceeding of the eighth ACM symposium on Document engineering*, ACM, New York, NY, USA, pp. 3–7, 2008.

S.Sushmita, M.Lalmas, A.Tombros, "Using digest pages to increase user result space: Preliminary design", *in* Lalmas *et al.* (2008), 2008.

Wan S., Paris C., Krumpholz A., "From Aggravated to Aggregated Search: Improving Utility through Coherent Organisations of an Answer Space", *in* Lalmas *et al.* (2008), 2008.

6IR : Un index paramétrable pour les requêtes ramifiées

Y. Péron

Laboratoire Valoria
Université de Bretagne Sud
Bât. Yves Coppens – B.P. 573
56017 Vannes Cedex
youen.peron@univ-ubs.fr

RÉSUMÉ. *Cet article contient une présentation de notre travail en cours de développement dans le domaine de la recherche d'informations dans des bases de documents semi-structurées. Nous cherchons à construire un système d'interrogation – dénommé 6IR pour Structure based IndeX Information Retrieval – qui fournisse une liste de documents similaires au contenu et à la structure d'une requête ramifiée. L'extraction des documents est basée sur l'identification de points communs entre leur structure et celle de la requête. Nous détaillons le processus d'indexation qui consiste à extraire des documents de la base tous les points d'accrochage exploitables dans le processus d'interrogation. Nous montrons comment parvenir à maîtriser l'explosion combinatoire de la taille de l'index en paramétrant la taille des points d'ancrage et les propriétés qui en découlent pour les documents candidats obtenus lors du processus d'interrogation.*

ABSTRACT. *This paper contains a presentation of our work in progress in the domain of information retrieval in base of semi-structured documents. We try to build a querying engine – called 6IR for Structure based IndeX Information Retrieval – which provides a list of documents similar in content and structure of a twig query. The extraction of documents is based on the identification of structure pattern. We detail the indexing process that consists of extracting all the patterns of the documents of the base useable for the process of interrogation. We show how to control the combinatorial explosion in the size of the index by setting the size of the patterns and the properties that followed on the documents obtained during the interrogation.*

MOTS-CLÉS : *recherche d'information, XML, requête ramifiée, fragmentation de document, indexation de document, recherche dans une collection*

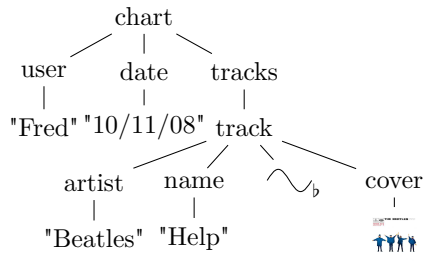
KEYWORDS: *information retrieval, XML, twig query, document fragmentation, document indexing, search in a collection*

1. Introduction

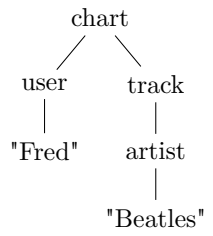
L'utilisation massive du standard XML pour la représentation de l'information implique de nouveaux problèmes liés au stockage, à l'indexation et à la recherche d'informations. Les documents XML ont une structure d'arbre (arbre DOM) qui leur permet d'embarquer du contenu non structuré et hétérogène. Par exemple l'arbre de la figure 1a regroupe des information aussi diverses qu'une date, un mot-clé, une série temporelle ou une image. Les modèles de bases de données classiques ont une structure trop rigide pour gérer l'hétérogénéité ; à l'opposé les listes inverses utilisées pour la recherche plein texte ne prennent pas en compte la structure des documents. Or il est naturel d'exprimer les requêtes, exactes ou approchées, sur les bases de documents XML avec des arbres reliant plusieurs critères de recherche. Par exemple le langage FleXPath (Amer-Yahia *et al.,* 2004) permet de définir des requête de la forme 1c qui correspond à l'arbre de la figure 1b. On qualifie ces requêtes de ramifiées *(twig query)*. Retrouver rapidement l'ensemble des documents D d'une collection $\mathcal{C}$ correspondant à une requête ramifiée R est une opération incontournable dans le processus d'interrogation des bases de documents XML.

Les auteurs de (Catania *et al.,* 2005) ont proposé une classification des systèmes de gestion des collections de documents XML selon trois types d'approches :

– **Décomposition des documents en chemins :** pour chaque feuille du document on construit le chemin qui relie cette feuille à la racine de l'arbre afin de créer une table avec comme clé la feuille et comme valeur le chemin qui lui est associé. L'approche de (Abiteboul, 1997) est très efficace pour les requêtes ne contenant qu'un seul chemin. Cependant un post-traitement est nécessaire



(a) Arbre DOM d'un document XML.



(b) Requête ramifiée.

$$//chart[.user[.contains(\text{"}Fred\text{"})]\ and\ .track[.artist[.contains(\text{"}Beatles\text{"})]]]$$

(c) Requête FleXPath

Figure 1. *Exemple de document* XML

si l'on veut répondre à des requêtes ramifiées comme dans (Popovici *et al.,* 2006).

– **Décomposition de la requête en jonctions :** Les approches (Bruno *et al.,* 2002) et (Jiang *et al.,* 2003) indexent les nœuds d'un arbre en fonction de leurs valeurs ou de leurs tags. Après interrogation de l'index, un algorithme vérifie que les relations de parentés concordent avec les relations de la requête. Les limitations de ces approches est de décomposé la requête ramifiée en plusieurs sous requêtes de type ancêtre–descendant.

– **Représentation séquentielle de la requête :** en utilisant une représentation séquentielle des arbres, les systèmes (Wang *et al.,* 2003) et (Rao *et al.,* 2004) utilisent des algorithmes de recherche de sous-séquences pour répondre à une requête. Une telle approche évite la décomposition de la requête en sous-requêtes et donc les jointures nécessaires à la fusion des résultats des sous-requêtes. Cependant la linéarisation des arbres engendre des faux-positifs que l'on doit filtrer.

Actuellement, on peut constater que les systèmes de résolution de requête ramifiée favorisent les relations ancêtre–descendant, que ce soit dans le choix du découpage de l'arbre, ou dans le choix de la représentation de la structure des document XML. Dans les deux premières approches citées ci dessus le contenu non-structuré sert de point d'accroche pour retrouver des réponses potentielles, puis le système vérifie la correspondance entre la structure de la requête et celle du document candidat. Ces approches basées sur l'indexation du contenu non-structuré impliquent que les documents XML contiennent des informations de même type (paragraphes, mots-clés, séries-temporelles).

Par rapport aux approches précédentes, nous cherchons un moyen pour indexer les documents XML qui permette à la fois :

– des comparaisons spécifiques à chaque type de contenu présent dans un document (image, texte, série temporelle) ;

– de mieux intégrer les relations de voisinage dans la structure.

Nous proposons dans cet article une approche qui, dans un premier temps, utilise la structure des documents comme points d'ancrage d'une requête et, dans un second temps, utilise une distance spécifique à chaque nœud pour en comparer le contenu avec celui de la requête.

Le reste de l'article est organisé de la manière suivante. Dans la section 2 nous décrivons l'ensemble de notre approche en nous focalisant sur le problème de la construction de l'index et la solution que nous proposons pour maîtriser l'explosion combinatoire qui en résulte. Dans la section 3 nous discutons de la pertinence de l'index ainsi construit et des possibilités offertes par sa paramétrisation. Nous terminons à la section 4 par un état des lieux de l'avance de nos travaux et les expérimentations en cours pour valider et enrichir notre approche.

2. L'approche 6IR

Étant donné une base de documents XML, on cherche à construire un système d'interrogation qui fournisse une liste de documents similaires au contenu et à la structure d'une requête ramifiée. L'extraction des documents candidats est basée sur l'identification de points communs entre leur structure et celle de la requête. Le processus d'indexation consiste à extraire des documents de la base tous les points d'accroches exploitables dans le processus d'interrogation. Ces points d'accroches sont des sous-arbres XML de taille très réduite (quelques nœuds). Nous détaillons dans cette section le processus d'indexation et son exploitation lors du processus d'interrogation de la base.

2.1. *Processus d'indexation de la base*

On s'intéresse dans un premier temps à évaluer N_P, le nombre total de points d'ancrage à extraire d'un ensemble de $|\mathcal{C}|$ documents d'une collection $\mathcal{C}$. Le nombre de points d'ancrage extraits d'un document D correspond au nombre de sous-arbres contenus dans l'arbre XML de ce document. Si on note $|D|$ la taille en nombre de nœuds d'un document D, $|D|_{max}$ la taille du plus grand document de la base et $\binom{n}{k}$ le coefficient binomial de k parmi n, alors le nombre total de points d'ancrage N_P de la collection $\mathcal{C}$ est borné par la relation :

$$N_P \leq \sum_{D \in \mathcal{C}} \left(\sum_{k=0}^{|D|} \binom{|D|}{k} \right) \leq |\mathcal{C}| \times 2^{|D|_{max}}$$

L'explosion combinatoire du nombre de points d'ancrage potentiels qui en résulte nous incite à fragmenter les documents pour réduire le facteur $|D|_{max}$ ainsi qu'a limiter et fixer la taille des points d'ancrages (facteur k). C'est pourquoi lors de l'indexation nous procédons en deux phases successives :

1) une phase de fragmentation qui consiste à réduire le nombre de nœuds dans l'arbre XML de chaque document ;

2) une phase d'extraction des points d'ancrage dont la taille (le nombre de nœuds) est fixée à une valeur arbitraire k.

2.1.1. *La fragmentation des documents et la table $\mathcal{T}_F$*

La fragmentation des documents consiste à extraire de l'arbre XML d'un document un ensemble de sous-arbres dont la structure est simplifiée. L'extraction est basée sur la structure des documents XML dont on suppose connaître soit les DTD soit les XML Schema : Tout document contenant une liste d'éléments provoque la génération d'une liste de fragments de documents contenant une occurrence de chaque élément. Ce remplacement est supervisé, de manière,

d'une part à limiter la multiplication du nombre de fragments, et d'autre part à privilégier certaines structures de requêtes adaptée à l'application visée.

Chaque fragment extrait d'un document est enregistré dans une table des fragments nommée $\mathcal{T}_F$. La clé d'une entrée est un identifiant composé d'un entier suffixé par le numéro de l'occurrence de l'élément dans la liste qui a servie à la fragmentation. La valeur associée à cette clé est une référence qui permet de retrouver le document dans le système de fichiers de la base XML.

2.1.2. *L'extraction des points d'ancrage et la table $\mathcal{T}_A$*

Maintenant que la structure des documents est simplifiée, on en extrait les points d'ancrage qui vont servir d'entrées dans l'index de la manière suivante.

On fixe arbitrairement la valeur de k, le nombre de nœuds de chaque point d'ancrage. Le choix de la valeur de k et ses conséquences sont discutées au paragraphe 3.

On construit de manière exhaustive l'ensemble des sous-arbres de taille k à partir de chaque fragment. Le nombre de points d'ancrage ainsi obtenu pour un fragment de taille n est borné par $\binom{n}{k}$. Ce nombre étant relativement important même pour de petites valeurs de k [1], nous optimisons l'utilisation de la mémoire en codant les arbres par des séquences de Prüfer (Prufer, 1918) comme dans les systèmes décrits dans (Rao *et al.,* 2004) et (Agarwal *et al.,* 2007).

Les points d'ancrage ainsi obtenus sont mémorisés dans une table $\mathcal{T}_A$. La clé d'une entrée est le codage de Prüfer du point d'ancrage. La valeur associée est l'identifiant du fragment (2.1.1) dont est extrait le point d'ancrage ; ou plus précisément, comme un point d'ancrage peut être présent dans plusieurs fragments, la valeur associée est l'ensemble des identifiants des fragments contenant ce point d'ancrage.

2.2. *Processus d'interrogation de la base*

On soumet au système 6IR une requête ramifiée que l'on décompose en points d'accroche suivant un algorithme identique à celui utilisé pour l'indexation (cf paragraphe 2.1.2) et avec la même valeur du paramètre k (taille des points d'accroche). Chaque point d'accroche issu de la décomposition est codé sous la forme d'une séquence de Prüfer et sert de point d'entrée dans la table $\mathcal{T}_A$. La lecture de la valeur associée à cette entrée dans la table $\mathcal{T}_A$ est un ensemble d'identifiants de fragments. L'intersection de tous les ensembles obtenus pour tous les points d'accroche de la requête fournit (via la table des fragments $\mathcal{T}_F$) la liste des documents candidats, similaires du point de vue de *leur structure* à celle de la requête.

---

1. Comme indiqué dans le paragraphe 3.3, les valeurs de k considérées vont de 1 à 3

Il reste ensuite à comparer *le contenu* de la requête avec celui des documents candidats. Nous opérons en deux phases successives.

1) On aligne la structure des arbres XML des documents candidats avec celle de l'arbre de la requête. La phase d'alignement est un problème connu ; on utilise la solution donnée dans (Bruno *et al.,* 2002).

2) On compare les données non structurées de la requête qui sont alignées avec celles des documents candidats.

La phase de comparaison du contenu peut ainsi dépendre du type d'information enregistré dans un nœud. Par exemple, s'il s'agit de mots clés, on peut utiliser un calcul de distance basé sur l'algorithme de Levenshtein (Levenshtein, 1966). Si le contenu est formé de paragraphes, une solution basée sur un TF-IDF sera mieux adaptée. De manière générale, notre approche s'adapte à tout type de données non structurées présentes dans les documents XML ; il faut disposer d'un algorithme de calcul de distance sur le type considéré.

3. Discussion

Dans la partie 2 on a expliqué comment extraire un ensemble de documents structurellement proches de la requête. On discute dans cette partie de la pertinence des documents extraits. Cette discussion va dans un premier temps s'intéresser aux gains apportés par l'index sur l'espace de recherche. Dans un deuxième temps nous allons discuter des conséquences de la fragmentation. Enfin nous étudierons l'influence du paramètre k (taille des points d'accroches) sur la qualité du voisinage structurelle considéré.

3.1. *Pertinence de l'index*

On peut se demander si les résultats d'un index basé que sur la structure réduit de façon conséquente l'espace de recherche. En effet si les documents ont tous la même DTD, alors les nœuds qui composent les arbres XML auront un nombre de labels très limité avec des relations identiques (celles guidées par la DTD). Les combinaisons possibles pour extraire les points d'ancrages seront elles aussi limitées. La probabilité qu'un fragment contienne un point d'accroche sera par conséquent très élevé, et l'espace de recherche sera peu réduit.

Pour répondre à ce problème, on propose de diversifier les éléments de structures en rajoutant des signatures issues du contenu non-structuré dans la structure. Par exemple dans l'arbre de la figure 1a on peut utiliser la signature « première lettre du mot-clé » pour rajouter un nœud ayant comme label « F » entre le nœud « user » et le contenu « Fred ».

En contrepartie la taille des fragments est augmentée et la recherche sur le contenu non-structuré est conditionnée par l'égalité entre la signature des données de la requête et du fragment. Le choix d'une fonction de signature peut s'avérer imposible pour des données plus importantes qu'un simple mot clé, comme un paragraphe ou une série temporelle.

3.2. *Conséquence de la fragmentation*

En découpant les documents en fragments de document, on supprime les liens entre les données contenus dans chaque fragment. De même que les approches basées sur la décomposition des arbres XML en chemins considèrent chaque chemin indépendant, l'approche 6IR considère que les fragments ne sont pas corrélés.

Cette hypothèse se justifie par le fait que la fragmentation, guidée par les DTD est supervisée par un administrateur pour répondre au mieux à l'application visée en se basant sur la sémantique des balises XML.

3.3. *Influence de la taille des points d'accroches*

Dans cette dernière partie de la discussion on s'intéresse aux conséquences du choix du facteur k. La qualité des fragments obtenus avec l'approche 6IR augmente avec la taille des points d'accroches.

En effet si on choisit des points d'accroche de taille $k = 1$ les points d'accroche sont des nœuds. Tous les nœuds de la requête ramifié sont contenus dans les fragments retournés par l'index mais la structure qui les relie peut être totalement différente. En prenant la valeur $k = 2$ on conserve le lien de parenté entre les nœuds cependant l'ordre des nœuds n'est pas conservé et il est possible de trouver des cas de faux positifs (les mêmes cas que dans le système VIST (Wang *et al.,* 2003). À partir d'une taille $k = 3$ on conserve la structure avec les triplets de la forme (fils,fils,parent) l'ordre des nœuds est respecté et les faux positifs de VIST sont éliminés.

L'inconvénient majeur d'augmenter la taille des points d'ancrage est de multiplier leur nombre et d'augmenter sensiblement la table $\mathcal{T}_A$.

4. Conclusion

Nous avons présenté dans cet article une approche qui indexe les documents XML en fonction d'éléments de petite taille de leur structure. La nouveauté dans cette approche est de ne pas indexer le contenu mais la structure ce qui permet de comparer les données non structurées et hétérogène avec une distance adéquate.

Nous avons developpé un prototype de l'application 6IR qui valide les mécanismes de fragmentation, d'extraction de points d'accroche, d'utilisation de signature des mots clés et l'interrogation de l'index. Cependant il n'a pas encore été testé sur une grande masse de documents contenant plusieurs types de contenu. Actuellement seule la distance de Levenshtein est utilisée.

Nous avons vu dans la partie discussion que l'indexation de la structure prenait tout son sens pour une valeur de k (taille des points d'accroches) égale à 3. Il nous parait intéressant d'appliquer cette méthode pour l'interrogation d'une collection de triplets RDF qui sont de plus en plus utilisés pour diffuser les informations du web sémantique.

5. Bibliographie

Abiteboul S., *Querying Semi-Structured Data*, Springer, 1997.

Agarwal N., Oliveras M. G., Chen Y., « Approximate Structural Matching over Ordered XML Documents », *Database Engineering and Applications Symposium, 2007. IDEAS 2007. 11th International*, p. 54-62, 2007.

Amer-Yahia S., Lakshmanan L. V. S., Pandit S., « FleXPath : flexible structure and full-text querying for XML », *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, ACM New York, NY, USA, p. 83-94, 2004.

Bruno N., Koudas N., Srivastava D., « Holistic twig joins : optimal XML pattern matching », *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, ACM Press New York, NY, USA, p. 310-321, 2002.

Catania B., Maddalena A., Vakali A., « XML document indexes : a classification », *Internet Computing, IEEE*, vol. 9, n° 5, p. 64-71, 2005.

Jiang H., Wang W., Lu H., Yu J. X., « Holistic twig joins on indexed XML documents », *VLDB '2003 : Proceedings of the 29th international conference on Very large data bases*, VLDB Endowment, p. 273-284, 2003.

Levenshtein V. I., « Binary Codes Capable of Correcting Deletions, Insertions and Reversals », *Soviet Physics Doklady*, vol. 10, p. 707, 1966.

Popovici E., Menier G., Marteau P., « Recherche approchée d'information dans une base de documents semi-structurés », *3ème Conférence en Recherche d'Information et Application*, p. 53-64, 2006.

Prufer H., « Neuer Beweis eines Satzesuber Permutationen », *Archiv fur Mathematik und Physik*, vol. 27, p. 142-144, 1918.

Rao P., Moon B., « PRIX : indexing and querying XML using prufer sequences », *Data Engineering, 2004. Proceedings. 20th International Conference on Data Engineering (ICDE'04)*, vol. 0, p. 288-299, 2004.

Wang H., Park S., Fan W., Yu P. S., « ViST : a dynamic index method for querying XML data by tree structures », *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, ACM Press New York, NY, USA, p. 110-121, 2003.

Classement collaboratif de manuscrits

Pierre-Edouard Portier

Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
pierre-edouard.portier@insa-lyon.fr

RÉSUMÉ. *Pour chaque projet d'édition numérique de manuscrits, après que le corpus ait été constitué, les chercheurs commencent par le classer. Cette opération demande de grands efforts d'interprétation, elle n'est pas neutre mais contribue à la construction du point de vue du chercheur sur son objet d'étude. Ainsi, plusieurs classements peuvent être proposés pour un même sous-ensemble de l'archive. Or il n'existe pas de plateforme informatique spécifique pour assister les chercheurs dans cette opération délicate. Nous en proposons une sous la forme d'un service Web et d'une IHM qui prennent en compte les spécificités de la tâche de classement et peuvent profiter à tout projet qui étudie un corpus de documents numérisés dans un des domaines des Humanités.*

ABSTRACT. *Every electronic edition of manuscripts begins by ordering the studied corpus. This operation is of a highly hermeneutic nature and contribute to the construction of each researcher viewpoint. For a same fragment of the corpus, multiple orderings can be proposed. However, there is no digital plateform aimed at assisting researchers in this task. That is why we developed a Web service and a graphical interface dedicated to this operation and beneficial to any Humanities project.*

MOTS-CLÉS : *bibliothèques numériques, XML, classement, Haskell*

KEYWORDS: *digital libraries, XML, ordering, Haskell*

Pierre-Édouard Portier

1. Introduction

De nombreux projets d'édition critique de manuscrits débutent par un classement du corpus des textes étudiés. Souvent les textes, de natures très diverses, sont initialement désordonnés. L'histoire des mains par lesquelles ils passèrent peut être complexe et un héritier les aura par exemple dérangés, ... Dans le cadre de notre projet de thèse financé par la région Rhône-Alpes, nous travaillons avec des chercheurs de l'ENS-LSH qui étudient les archives manuscrites du philosophe J.T.Desanti [1]. Fort de cette expérience, nous découvrons que l'opération de classement fait appel à la faculté d'interprétation. Le classement est ainsi toujours nécessaire *et* particulier. C'est pourquoi son instrumentation sera profitable. En effet, pour être propre à un interprétant elle demande d'être accompagnée de possibilités de travail collaboratif, pour être universelle elle demande à ce que son résultat soit partageable. La numérisation du corpus permet de travailler dans le domaine symbolique où il est facile de construire des classements concurrents de sous-ensembles de l'archive. L'existence d'un standard pour l'édition électronique de textes, la TEI ( ), permet d'assurer partage et réutilisation des résultats du classement. Notre travail consiste à déterminer et assurer les conditions pour permettre à plusieurs utilisateurs de travailler au classement d'un corpus d'archives manuscrites et pour offrir un accès unifié à une représentation standard des classements produits. Nous proposons une solution sous forme d'un service Web qui respecte le patron d'architecture dit REST ( ) afin d'inscrire les objets résultats du classement au sein même de la structure du Web et permettre de réutiliser ces objets dans une grande variété de contextes (publication, partage avec d'autres projets d'édition électronique, ...). Nous avons aussi développé un programme client écrit dans le dialecte Smalltalk Squeak ( ) qui offre une grande liberté d'interaction à l'utilisateur afin de simuler au mieux les opérations du chercheur à sa table (physique) de travail. Dans la suite, nous présenterons les travaux existants les plus proches de notre proposition, nous décrirons ensuite le service Web puis l'IHM.

2. État de l'art

2.1. *Propriétés nécessaires pour un système de classement d'archives manuscrites*

Après avoir interrogé et observé les chercheurs en sciences humaines avec lesquels nous travaillons, nous avons pu déterminer cinq caractéristiques nécessaires à tout système de classement :

– préservation de l'ordre initial de l'archive telle que trouvée avant numérisation

– système d'annotations évolué qui permette la création de relations n-aires (afin de pouvoir exprimer des assertions du type : "cette collection de pages est *une version alternative* de cette autre page")

1. environ 300 documents pour 30 000 pages

– environnement collaboratif où plusieurs utilisateurs peuvent proposer des classements concurrents.

– résultats du classement facilement publiables

Nous passons maintenant en revue les travaux existants qui répondent à un ou plusieurs des points précédents.

– Collate ( ) est un système Web de travail collaboratif orienté documents. Il ne permet pas le reclassement mais possède un système d'annotations collaboratives élaboré qui permet la construction d'une forme de discours. Cependant les annotations ne sont ici que des relations 1-aire. De plus, la publication des ressources n'est pas prise en compte.

– BAMBI ( ) et son successeur Diphilos ( ) sont des systèmes hypermedia pour la transcription de manuscrits. Les images de manuscrits sont entrées dans une base de données au début d'un projet et la construction de classements concurrents n'est pas possible. Cependant, la technologie SGML/HyTime utilisée permet la création de liens bidirectionnels entre pages, mais ne permet pas de gérer des relations n-aires. Finalement, il n'y a pas de possibilités simples de publier les ressources.

– Une partie du projet DEBORA ( ) consiste en une bibliothèque numérique avec des fonctionnalités de travail collaboratif. Y est introduit la notion de livre virtuel : la représentation d'un chemin à travers les pages de l'archive. Mais ces chemins ne sont pas eux-mêmes des ressources à part entière et ne peuvent pas entrer dans un processus collaboratif qui permettrait de les échanger, les annoter, etc.

– HyperNietzsche ( ) (aujourd'hui Nietzschesource) est un système pionnier de bibliothèque numérique. La problématique du reclassement est prise en considération très sérieusement mais le choix est fait de réaliser le classement une seule fois et par un petit comité d'experts. Comme pour DEBORA, un mécanisme de chemins existe qui a les mêmes défauts. TALIA ( ) est la suite d'HyperNietzche et utilise les technologies du Web sémantique, ce qui permet de représenter des relations n-aires.

– BRICKS ( ) est une architecture P2P de gestion de réseaux de bibliotèques numériques accompagné d'un ensemble d'applications construites au dessus de cette architecture. Elle introduit les deux notions de collections physiques et logiques. Les collections logiques contiennent des liens vers des objets de collections physiques. Mais un objet, en tant qu'il appartient à une collection logique, ne peut pas être annoté. RDF est utilisé pour créer des relations entre objets, il est donc possible de créer des relations n-aires.

Finalement, nous n'avons pas trouvé de solution qui réponde à l'ensemble des critères que nous avons énoncés plus haut. Ainsi, nous présentons maintenant une solution complète et générique à cette problématique.

3. Interface d'accès aux archives manuscrites

Dans cette partie nous décrivons un service Web qui permet de gérer des collections de pages manuscrites. Il assure une représentation toujours correcte de l'archive dans un sous-ensemble du langage XML défini par la TEI.

3.1. *Un standard : la TEI*

La TEI ( ), Text Encoding Initiative, est un consortium qui développe et maintient un standard pour la représentation des textes électroniques. Ses recommandations constituent une expertise dont peut profiter tout projet d'édition électronique. Elles sont exprimées sous la forme modulaire et extensible d'un schéma XML documenté.

Pour le classement d'archives manuscrites nous utilisons cinq balises de la TEI (graphic, teiCorpus, TEI, facsimile et surface ... voir figure 1 pour un exemple d'utilisation).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <teiCorpus n="1">
3    <TEI>
4      <facsimile>
5        <surface n="1">
6          <graphic url="pochettes/0200/0001_high.jpeg"/>
7          <graphic url="pochettes/0200/0001_low.jpeg"/>
8        </surface>
9        <surface n="2">
10         <graphic url="pochettes/0200/0002_high.jpeg"/>
11         <graphic url="pochettes/0200/0002_low.jpeg"/>
12       </surface>
13       <surface n="3">
14         <graphic url="pochettes/0200/0003_high.jpeg"/>
15         <graphic url="pochettes/0200/0003_low.jpeg"/>
16       </surface>
17       <surface n="4">
18         <graphic url="pochettes/0200/0004_high.jpeg"/>
19         <graphic url="pochettes/0200/0004_low.jpeg"/>
20       </surface>
21     </facsimile>
22   </TEI>
23
24
25
26
```

```
28  <teiCorpus n="2">
29    <TEI>
30      <facsimile/>
31    </TEI>
32    <teiCorpus n="1">
33      <TEI>
34        <facsimile>
35          <surface n="1">
36            <graphic url="pochettes/0200/2_1/0001_high.jpeg"/>
37            <graphic url="pochettes/0200/2_1/0001_low.jpeg"/>
38          </surface>
39          <surface n="2">
40            <graphic url="pochettes/0200/2_1/0002_high.jpeg"/>
41            <graphic url="pochettes/0200/2_1/0002_low.jpeg"/>
42          </surface>
43          <surface n="3">
44            <graphic url="pochettes/0200/2_1/0003_high.jpeg"/>
45            <graphic url="pochettes/0200/2_1/0003_low.jpeg"/>
46          </surface>
47          <surface n="4">
48            <graphic url="pochettes/0200/2_1/0004_high.jpeg"/>
49            <graphic url="pochettes/0200/2_1/0004_low.jpeg"/>
50          </surface>
51        </facsimile>
52      </TEI>
53      <teiCorpus n="2">
```

Figure 1. *Partie d'un fichier XML TEI : les lignes 3 à 22 représentent un groupe de 4 pages qui entourent d'autres groupes, la description du premier de ces groupes commence à la ligne 28*

3.2. *Une architecture : REST*

La figure 2 est un modèle, sous forme de diagramme de classes UML, des ressources offertes par le service Web que nous avons développé. Notre service respecte le schéma d'architecture dit REST ( ). Le principe à la base de cette architecture est d'avoir un nombre illimité de ressources avec pour chacune un identifiant unique (par exemple une URL) et au plus les quatre opérations définies par le protocole HTTP : GET, PUT, DELETE et POST. De plus à chacune de ces opérations est associée une sémantique générique.

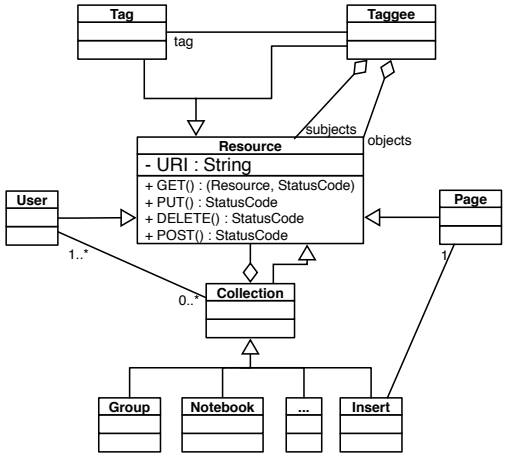


Figure 2. *modèle de domaine du service Web*

– La méthode GET doit servir à obtenir une représentation de la ressource, elle doit être sûre (ne pas modifier la ressource) et idempotente (elle produit toujours le même effet qu'elle soit itérée une ou plusieurs fois).

– La méthode PUT doit servir à modifier la représentation d'une ressource identifiée, si l'identifiant n'existe pas la ressource est créée. Cette opération est idempotente.

– La méthode DELETE doit servir à supprimer une ressource et être idempotente.

– La méthode POST doit être réservée aux opérations non sûres et éventuellement non idempotentes qui ne relèvent pas des trois premières méthodes.

Nos cinq types de ressources les plus essentiels sont l'utilisateur, le groupe, la page, la collection et l'insert. La collection est une agrégation de ressources, le groupe une collection d'utilisateurs, l'insert une sous-collection associée à la page de la collection où elle est insérée.

Le besoin de représentation de relations n-aires est satisfait grâce aux deux types de ressources *Tag* et *Taggee*. Un Tag est un terme. Un Taggee est une relation nommée d'un Tag et à laquelle des ressources peuvent participer soit en tant que sujets soit en tant qu'objets. Par exemple, l'assertion : "Les pages A et B sont des brouillons pour la page C" peut se modéliser au moyen d'un Taggee nommé du Tag "brouillon-pour" et avec trois participants : les pages A et B en tant que sujets, la page C en tant qu'objet.

Cette architecture en place, supposons que l'utilisateur bob veuille remplacer la troisième page de sa seconde collection par la première page du second des inserts qui se trouvent à la deuxième page du premier cahier de l'archive. Les opérations nécessaires sont GET suivi de PUT, informellement :

```
p := GET http://serveur.org/cahiers/1/pages/2/inserts/2/pages/1
PUT http://serveur.org/users/bob/collections/2/pages/3 p
```

Ou bien, supposons que l'utilisateur bob veuille ajouter la troisième page de la première collection de l'archive original à sa quatrième collection. Cette opération correspond à un GET suivi d'un POST, informellement :

```
p := GET http://serveur.org/collections/1/pages/3
POST http://serveur.org/users/bob/collections/4/pages/ p
```

3.3. *Traitement sûr des documents XML*

Comme rappelé dans ( ) il existe trois familles de solutions pour traiter des documents XML : les API XML telles que SAX ou DOM, les langages spécialisés tels que XSLT ou XDuce ( ), les isomorphismes entre types de données XML et types d'un langage de programmation. Les avantages de cette dernière solution associée à un langage de programmation fonctionnel pur (sans effets de bord) et fortement typé (nous choisissons Haskell) sont l'absence de phase d'analyse syntaxique et l'assurance offerte par le compilateur de tranformations correctes : un document valide se transforme en un document valide.

En combinant l'utilisation d'un sous-ensemble du langage de balisage défini par la TEI, une architecture REST et une correspondance entre types XML et types du langage Haskell nous avons développé un service Web de classement d'archives manuscrites qui est sûr, repose sur un standard établi et est "universellement" accessible car inscrit dans l'architecture même du Web.

4. IHM pour le classement d'archives manuscrites

4.1. *Utilisateurs cibles*

Nos utilisateurs sont des chercheurs en sciences humaines qui s'approprient un corpus documentaire en le classant. Ainsi, l'IHM doit offrir une grande liberté d'interactions pour simuler au mieux les opérations habituelles de ces chercheurs à leur table de travail. Les résultats de ces classements pourront devenir des objets consultables par d'autres types d'utilisateurs (simples lecteurs, philologues, etc.) au travers éventuellement d'une autre interface mais toujours servis par l'architecture décrite plus haut.

4.2. *Approche dynamique*

Nous nous sommes tournés vers le système de développement d'IHM appelé Morphic ( ) initialement développé pour le langage orienté objet par prototypes Self de Sun

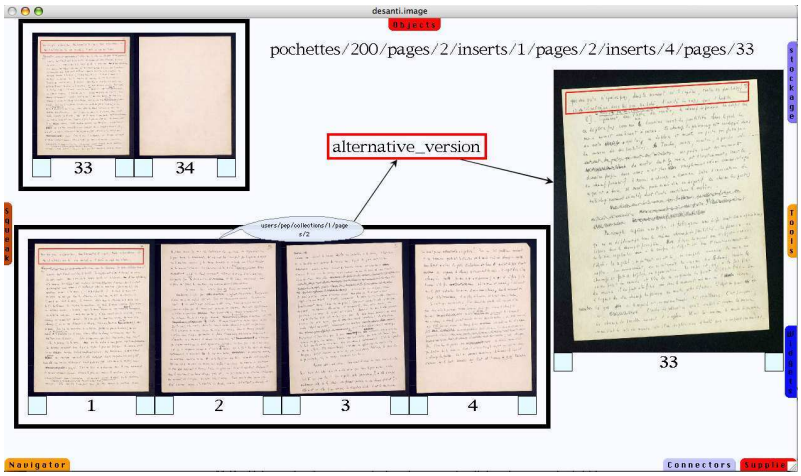


Figure 3. *Copie écran de l'IHM : en haut à gauche un navigateur ouvert sur l'ensemble de pages d'un insert d'une pochette ; en bas à gauche un navigateur ouvert sur une collection nouvellement créée pour classer les pages de l'insert précédent par glisser-déposer d'une collection dans l'autre ; sur la droite une page dont la collection nouvellement créée est une version alternative ; au centre la relation "version alternative" ; en haut un objet URL avec lequel beaucoup d'opérations sont possibles par simples glisser-déposer*

puis repris dans le dialecte Smalltalk Squeak ( ). Nous y avons trouvé les deux caractéristiques qui nous étaient nécessaires : interactivité et réactivité. La figure 3 est une copie écran de l'application. Le système est hautement interactif puisque pour accéder aux objets de l'interface, les examiner, changer leurs propriétés, il suffit d'une interaction directe avec leurs représentations graphiques sans qu'il soit nécessaire de passer par une représentation intermédiaire. Nous obtenons ainsi la vitesse de développement dont nous avions besoin.

5. Conclusions

Les chercheurs en sciences humaines et sociales travaillent principalement sur des corpus de textes. Ces derniers nécessitent presque toujours un classement. Cette opération est très délicate et requiert toute l'attention et l'imagination des chercheurs. Il n'existait pas d'outil numérique spécifique pour assister les chercheurs dans ce classement pourtant systématiquement nécessaire. Nous avons développé cet outil et l'avons conçu suffisamment générique et interopérable pour répondre aux besoins des chercheurs des diverses disciplines (philosophie, littérature, etc.) qui traitent des corpus textuels et pour leur permettre de partager facilement les résultats de leur travail. Finalement, nous sommes en train d'inscrire ce programme au sein d'une plateforme que

nous développons et qui est destinée à assister les chercheurs du domaine des Humanités dans l'édition numérique de textes ; elle comprend, entre autres fonctionnalités, un module évolué pour la transcription.

6. Bibliographie

[ATA 03] ATANASSOW F., CLARKE D., JEURING J., « Scripting XML with Generic Haskell », rapport, 2003, Utrecht University.

[BER 07] BERTONCINI M., « On the Move Towards the European Digital Library : BRICKS, TEL, MICHAEL and DELOS Converging Experiences », *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings*, vol. 4675 de *Lecture Notes in Computer Science*, Springer, 2007, p. 440-441.

[BOZ 97] BOZZI A., CALABRETTO S., « The Digital Library and Computational Philology : The BAMBI Project », *ECDL*, vol. 1324 de *Lecture Notes in Computer Science*, Springer, 1997, p. 269-285.

[BOZ 04] BOZZI A., « The DIPHILOS workstation for critical apparatus management : some experiments on medieval provencal texts », *Textual Criticism and Genetics Confronting Methods, Louvain-la-Neuve*, 2004.

[BUR 07] BURNARD L., BAUMAN S., « TEI P5 : Guidelines for Electronic Text Encoding and Interchange », , 2007.

[D'I 07] D'IORIO P., « Nietzsche on New Paths : The HyperNietzsche Project and Open Scholarship on the Web », *Maria Cristina Fornari, Sergio Franzese (Ã©ds.), Friedrich Nietzsche. Edizioni e interpretazioni, Pisa ETS*, 2007.

[FIE 00] FIELDING R. T., « Architectural styles and the design of network-based software architectures », PhD thesis, 2000.

[HAH 08] HAHN D., NUCCI M., BARBERA M., « The Talia library platform - Rapidly building a digital library on Rails », *4th Workshop on Scripting for the Semantic Web*, 2008.

[HOS 00] HOSOYA H., PIERCE B. C., « XDuce : A typed XML processing language (preliminary report », *In Proc. of Workshop on the Web and Data Bases (WebDB*, Springer-Verlag, 2000, p. 226–244.

[ING 97] INGALLS D., KAEHLER T., MALONEY J. M., WALLACE S., KAY A., IMAGINEERING W. D., « Back to the future : The story of Squeak, A practical Smalltalk written in itself », *In Proceedings OOPSLA 97, ACM SIGPLAN Notices*, ACM Press, 1997, p. 318–326.

[MAL 95] MALONEY J. H., SMITH R. B., « Directness and Liveness in the Morphic User Interface Construction Environment », *In Proceedings of User Interface and Software Technology (UIST 95) ACM*, ACM Press, 1995, p. 21–28.

[NIC 00] NICHOLS D. M., PEMBERTON D., DALHOUMI S., LAROUK O., BELISLE C., TWIDALE M. B., « DEBORA : developing an interface to support collaboration in a digital library », *European Conference on Digital Libraries*, Springer, 2000, p. 239–248.

[STE 04] STEIN A., KEIPER J., BEZERRA L., BROCKS H., THIEL U., « Collaborative Research and Documentation of European Film History : The COLLATE Collaboratory », *In International Journal of Digital Information Management*, 2004, p. 30–39.

Extraction des connaissances à partir du Web pour la recherche des images géoréférencées

Houda BOUAMOR[1]

LIMSI-CNRS et Université Paris-Sud 11

B.P. 133, 91403 Orsay Cedex

houda. bouamor@limsi.fr

RÉSUMÉ. Les bases de données géoréférencées connaissent un rôle croissant dans une grande variété de domaines d'application. La création manuelle de ces bases de données est cependant une opération coûteuse. Cela a suscité un intérêt pour l'automatisation de leur construction, par exemple, par l'exploitation des informations géographiques présentes sur le Web. Dans ce travail, nous présentons une nouvelle approche automatique pour la construction d'une base de données géoréférencées multilingues et à large échelle en se basant principalement sur l'encyclopédie collaborative Wikipédia pour identifier les noms géographiques, catégoriser ces noms, trouver leurs coordonnées géographiques et les classer selon une estimation de leur pertinence. La base de connaissances obtenue a été intégrée dans ThemExplorer, une application de recherche d'images géoréférencées.

ABSTRACT. Geolocalized databases are becoming necessary in a wide variety of application domains. The manual creation of such databases is an expensive operation which stimulated the interest for the automation of their construction, by mining geographic information from the Web. In this article, we present and evaluate a new automated approach for creating a geographical database. Our technique is based on Wikipedia to identify geographical names, categorize them, find their geographical coordinates and rank them. Then this database has been integrated in ThemExplorer, an application for geographic image retrieval.

MOTS-CLÉS : Bases de données géoréférencées, Wikipédia, extraction d'information, fouille de données, ThemExplorer.

KEYWORDS: Geographic databases, Wikipedia, Information extraction, data mining, ThemExplorer

[1] Ce travail a été réalisé dans le cadre d'un stage de Master 2 au CEA/LIST/ LIC2M

1. Introduction

La structure minimale d'un thésaurus géographique est défini par Hill [4] de la façon suivante : chaque entité de la base doit être renseignée avec les informations suivantes : son nom (***Musée du Louvre***), sa localisation géographique représentée par les coordonnées (***48.862 N, 2.336 E***) et sa classe parent regroupant cette entité avec d'autres éléments de même nature (***Musée***). Les travaux de Hill ont permis de définir un schéma standard pour les bases de données géographiques et de mettre en place un système de classification hiérarchique des catégories géographiques. Or, la construction manuelle de bases de données géographiques, telles que Alexandria [4] ou Geonames[2] s'avère un processus long et fastidieux ce qui motive donc l'intérêt pour des travaux visant à automatiser ce processus.

Dans cet article, nous décrivons une méthode permettant de créer automatiquement une base de connaissances multilingue et à large échelle pour le domaine géographique en exploitant l'encyclopédie Wikipédia, une source d'informations riche en contenu semi-structuré qui a été utilisée dans plusieurs travaux de recherche (par exemple [2,5]). Un des objectifs de ce travail est l'amélioration de ThemExplorer [7], application de recherche d'images géoréférencées du CEA LIST, par l'intégration des données acquises.

Cet article est structuré de la façon suivante : dans la section 2, nous donnons un aperçu des travaux liés à notre approche. Dans la section 3, nous exposons notre méthode de création automatique du thésaurus géographique et la section 4 présente les différentes évaluations effectuées afin de valider notre approche.

2. État de l'art et approche suivie

Il existe un fort besoin d'accès à des informations géographiques, tel que montré par l'étude de Sanderson et Han [9] qui révèle que jusqu'à 37% des requêtes soumises à un moteur de recherche concernent des informations géographiques. Aujourd'hui, on dispose de plusieurs grandes sources d'informations géographiques créées manuellement telles que Alexandria et Geonames. Geonames contient plus de 8 millions de noms correspondant à environ 6,5 millions de lieux. Le nom, le type et les coordonnées sont **renseignés** pour chaque entité. Chaque entité appartient à une classe géographique bien déterminée ce qui offre la possibilité de mettre en place une navigation thématique. Les principales limites de Geonames concernent la variation de la couverture des différentes régions du monde et l'impossibilité d'afficher les résultats par ordre de pertinence. Or, comme souligné par Toriani et *al.* [10], ce classement est essentiel pour l'exploitation efficace des thésaurus géographiques en recherche d'information.

La constitution de bases de données géographiques à partir de corpus non structurés est illustrée par les travaux de Rattenbury et *al.* [8]. Leur approche vise à

[2] http://www.geonames.org/

séparer les noms géographiques des autres informations associées aux photographies géoréférencées de la base Flickr. Leur évaluation mesure une précision de 82% et un rappel de 50%. Buyukkoten et *al.* [3] proposent eux une méthode permettant l'identification et l'exploitation des informations géographiques à partir des sites Web afin de permettre aux moteurs de recherche d'exploiter ces informations à des fins de classement.

Le projet Gazetiki [6] est une base d'informations géographiques construite automatiquement par combinaison d'informations extraites de sources hétérogènes : Wikipédia, Panoramio et AllTheWeb. Son but est d'enrichir et compléter Geonames en utilisant un modèle du domaine permettant une intégration facilitée des deux ressources.

Notre travail se situe dans la continuité de celui de Gazetiki, consiste également à enrichir les informations de Geonames en ajoutant des données géographiques ainsi que des catégories relatives aux entités contenues dans Wikipédia. Alors que Gazetiki ne porte que sur l'anglais, notre projet à une visée multilingue et couvre six langues. Le processus d'extraction des coordonnées et de la valeur de pertinence des entités est similaire à celui de Gazetiki. L'identification des entités et leur catégorisation se fait de manière différente : la première se base sur la reconnaissance des articles géoréférencés de Wikipédia, la seconde sur l'extraction de plusieurs catégories candidates à partir des différentes parties d'un article, puis d'un processus de validation de la catégorie finale comme Ahern et *al.*[1], nous extrayons des fragments de pages à partir de Wikipédia afin de les catégoriser, mais nous exploitons également d'autres sources du Web. La limitation à un domaine précis nous permet de réaliser des analyses plus spécifiques et plus complexes des documents que dans d'autres travaux [2,8].

3. Extraction des informations à partir de Wikipédia et construction de la base des connaissances

Les articles de Wikipédia sont constitués de textes écrits en langage naturel, et comportent d'autres types d'informations structurées : les **Infobox***,* les informations sur les catégories, les coordonnées géographiques, et des liens vers les pages écrites en d'autres langues. Nous avons téléchargé les versions archivées de Wikipédia en six langues : *français, anglais, italien, espagnol, allemand* et *néerlandais,* afin de les utiliser pour construire notre gazetteer multilingue final. Notre travail consiste donc à extraire les éléments du tuple décrivant une entité géographique : *nom, localisation, catégorie, valeur de pertinence*.

La valeur de pertinence associée à chaque objet permet de classer les entités les plus populaires pour les présenter en priorité à l'utilisateur, ce qui est utile pour les régions à forte densité de monuments comme ***Paris*, *New York*...** Cette valeur a été calculée de la même façon que dans Gazetiki [6] .

3.1. Sélection des articles géoréférencés de Wikipédia

L'identification des articles géoréférencés repose sur la présence du couple {latitude, longitude}. Nous remarquons que le format des coordonnées géographiques n'est pas homogène notamment parce qu'elles sont souvent introduites par des non géographes. Le tableau 3.1 présente le nombre d'articles géoréférencés par langue.

	Anglais	**Français**	**Italien**	**Espagnol**	**Allemand**	**Néerlandais**
Nombre d'articles géoréférencés	242 142	76 477	88 513	45 534	96 405	122 915

Tableau 3.1: Nombre d'articles géoréférencés dans Wikipédia (Avril 2008) pour chacune des langues

Par ailleurs, certains articles ne sont écrits que dans une seule langue. Pour l'anglais, par exemple, il existe **107 611** articles parmi les **242142** qui ne sont pas traduits, **53438** ont une traduction en français, **48464** en italien, **28031** en espagnol, **56429** en allemand et **81305** en néerlandais.

3.2. Identification

Dans Wikipédia, chaque article a un nom unique qui est son titre, celui-ci comporte le nom de l'entité (ex : ***Château de Versailles***), accompagné dans certains cas du lieu où elle est située (ex : ***Manta (Ecuador)***) et dans d'autres, de sa catégorie (*ex :**Luzon (eiland)***). Le processus d'extraction du nom de l'entité passe par deux étapes. Dans la première, on extrait le nom dans la langue principale (celle dans laquelle l'article est écrit) en analysant la partie titre. Dans la seconde, on analyse les traductions, de cet article, afin d'extraire le nom de cette entité dans les autres langues. Le résultat est une entrée de la base de données comportant les noms des entités géographiques dans les six langues. La base de données ainsi créée sera enrichie par d'autres informations telles que la localisation de chacune des entités.

3.3. Localisation

Généralement, les coordonnées géographiques sont représentées par le couple (***latitude, longitude***) qui précise la position spatiale de l'objet. Mais dans Wikipédia, ce couple n'est presque jamais renseigné explicitement ou ne figure pas dans un format bien défini. Ce travail consiste à extraire toutes les coordonnées géographiques des entités représentées et à les convertir dans un format standard. Pour cela, nous avons extrait manuellement 31 motifs, parmi eux 7 sont communs aux six langues, un exemple de ces motifs est le suivant : *"Coor dms" : coordonnées en degré, minutes et secondes et direction N|S et E|O.*

3.4. Catégorisation

A partir de Wikipédia, nous avons établi un dictionnaire des catégories géographiques en anglais ainsi que leurs traductions dans les cinq autres langues. Pour un nombre limité de concepts, il n'existe pas de traduction, donc nous avons saisi la traduction manuellement. Ce dictionnaire nous permet de reconnaître la catégorie à extraire et d'enrichir la liste des catégories de *ThemExplorer*.

Notre méthode de catégorisation est apparentée à celle de Popescu et *al.* [6], mais nous utilisons plusieurs parties de la structure de l'article pour extraire des classes parents candidates et nous mettons en place une procédure de vote.

Les noms des objets géographiques contiennent souvent une référence explicite à leur catégorie, par exemple ***Tour* Eiffel**, **Golden Gate *Bridge***. Nous affectons temporairement cette catégorie à l'entité. Néanmoins, cela produit des erreurs pour des termes comme ***Cathedral of Learning*** qui n'est pas une cathédrale mais un gratte-ciel. De plus, cette méthode est inefficace pour les noms qui n'incluent aucune référence à leur classe, comme ***London Eye*** ou ***Parthenon.***

On analyse, ensuite, le contenu de la première phrase décrivant l'objet géographique. Celle-ci est habituellement une définition contenant une référence explicite à la catégorie cherchée. Par exemple, pour ***Notre Dame de Paris***, la première phrase est : ***Notre Dame de Paris is a Gothic cathedral on the eastern half of the Ile de la Cité.*** L'attribution de la catégorie est faite en deux étapes, nous cherchons la première apparition du verbe ***to be*** et retenons la partie à droite du verbe : ***a Gothic cathedral on the eastern half of …*** Puis, toutes les éléments du dictionnaire sont comparés au contenu de cette partie. Nous retenons comme deuxième catégorie temporaire celle qui apparaît la première.

Puis, on extrait les catégories prédéfinies par Wikipédia, situées en fin d'article et on les retient comme autres catégories temporaires.

Une procédure de vote est finalement mise en place pour choisir la catégorie la plus pertinente parmi toutes les catégories temporaires.

3.5. Extraction des entités en se basant sur le vocabulaire géographique

Il existe dans Wikipédia des articles définissant des entités géographiques mais qui ne sont pas géoréférencées. Pour découvrir ces entités, nous utilisons notre dictionnaire comme base pour la recherche des articles non géoréférencés. Cette méthode nous permet d'identifier le nom et la catégorie de chaque entité, mais pour les intégrer dans la base finale, il faut trouver leurs coordonnées géographiques.

4. Résultats et Évaluations

4.1. Résultats

Nous obtenons au final une base de connaissances multilingue contenant environ **700 000** entités : *noms, catégories, localisation et valeur de pertinence.*

Nous avons comparé nos résultats avec ceux de Geonames, celui-ci intègre des références à des articles Wikipédia, sans effectuer une analyse de leur contenu autre que l'extraction des coordonnées géographiques. Comme le montre la figure 5.1, notre approche assure une meilleure couverture dans l'extraction d'articles pertinents pour le domaine géographique dans la plupart des langues. La différence observée s'explique par le traitement de versions différentes de Wikipédia mais aussi par la grande diversité de motifs traités dans notre travail.

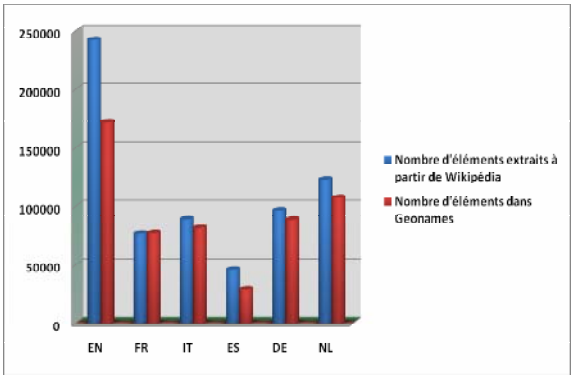


Figure 4.1- Nombre d'entités géographiques extraites de Wikipédia par notre approche et par Geonames.

4.2. Analyse de l'extraction d'articles géoréférencés

Dans Wikipédia, nous avons été confrontés à la polysémie de certains mots utilisés comme déclencheurs pour identifier les articles géoréférencés (ex : ***l*ongitud** *:* en espagnol désigne la longitude, mais aussi la longueur dans d'autres contextes) ; les faux positifs ont été éliminés car ils ne comprennent aucun des critères que nous avons définis. Parmi les résultats obtenus, on a observé la présence d'articles décrivant des lieux tels que ***Byrgius (crater)*** qui est un cratère de la lune. Les coordonnées de ce cratère sont des coordonnées sélénographiques renseignées dans un Infobox comportant ses caractéristiques.

4.3. Évaluation de la catégorisation

La qualité de la catégorisation a été évaluée semi-automatiquement. Nous avons choisi un échantillon de **1200** entités dans les six langues (200 par langue). Les résultats (tableau 4.2) montrent que la catégorisation réalisée est correcte dans plus de **95%** des cas, ce qui représente un taux de succès très satisfaisant pour une méthode complètement automatique. On observe une faible variabilité de la qualité de la catégorisation avec le changement de la langue (valeur minimale de 94% et maximale de 98%).

	EN	FR	DE	IT	ES	NL
Erreurs (/200)	12	4	8	12	10	6
Précision	94%	98%	96%	94%	95%	97%

Tableau 4.2- Résultats de l'évaluation de la catégorisation

4.4. Analyse des erreurs de catégorisation

Les erreurs sont causées principalement par des définitions complexes. Par exemple, le verbe ***to be*** est parfois suivi par une référence à la position géographique de l'objet et non par sa classe parent : **X est situé à l'est de Y et est un Z**. Dans ce cas, au lieu d'extraire Z, il est possible de trouver un élément du vocabulaire géographique dans Y qui sera extrait par notre algorithme. Nous avons aussi remarqué quelques erreurs de catégorisation en raison de particularités linguistiques des catégories en l'absence de certains termes. Par exemple, la langue allemande est une langue agglutinante, donc les catégories sont plus difficiles à extraire.

4.5. Évaluation de l'extraction d'entités non géoréférencées

Nous avons extrait un grand nombre de nouvelles entités décrites dans des articles de Wikipédia non géoréférencés. En nous basant uniquement sur le champ « catégorie » de l'Infobox, nous obtenons **12 117** (41%) nouvelles entités avec leur bonne catégorie. Nous avons ensuite analysé la première phrase des articles ainsi que la partie relative à la catégorisation pour trouver d'autres entités qui possèdent des informations liées à leur catégorie géographique. On a évalué cette méthode d'extraction en choisissant au hasard **200** entités. L'extraction est correcte à **92.5%**. Les erreurs proviennent surtout des articles de catégorisation qui ont des titres comportant le terme ***category*** (ex : ***category : cities in Ecuador)***. De plus, nous avons remarqué que quelques articles sont relatifs à des personnes. Leur existence s'explique par l'apparition du mot ***states*** dans la définition de l'entité ou dans la partie de catégorisation.

5. Conclusion et Perspectives

Le travail qu'on a présenté dans cet article a pour objectif de construire une base de connaissances géographiques multilingue et à large échelle. Nous avons réussi à ajouter plus de 700 000 entités à l'application existante de recherche d'images géolocalisées sur Internet ThemExplorer. Par ailleurs, et les évaluations montrent des résultats encourageants. En nous inspirant des travaux de Popescu et *al.* [6], nous avons extrait pour chaque entité : son nom, sa catégorie géographique, ses coordonnées et une mesure de pertinence et ce à partir de Wikipédia.

Une première perspective de ce travail est d'appliquer la méthode d'extraction d'articles non géoréférencés et de catégorisation pour les autres langues. Une fois les noms des entités et leurs catégories extraits, on essaiera de découvrir leurs coordonnées géographiques à partir de Panoramio et de calculer leur mesure de pertinence pour les ajouter à la base finale. Une deuxième perspective concerne

l'évaluation d'une catégorisation multilingue des entités. Les résultats obtenus pour une catégorisation monolingue montrent une précision de plus de **95%**, mais nous sommes convaincus qu'il est possible de les améliorer en utilisant plusieurs langues. Une troisième perspective consiste en l'application de ces méthodes sur d'autres sources d'information, comme Flickr qui contient un très grand nombre d'images géoréférencées.

Références

[1] S. Ahern, M. Naaman, R. Nair, J. Yang. 2007. WorldExplorer: Visualizing Aggregate Data from Unstructured Text in Geo-Referenced Collections. *In Proc. of JCDL (Vancouver, Canada, June 2007).*

[2] S. Auer, C. Bizer, G. kobilarov, J. Lehmann, R. Cyganiak et Z. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. *In Proc.of ISWC 2007 (Busan, Korea, November 2007)*

[3] O. Buyukkoten, J. Cho, H. Garcia-Molina.1999.Exploiting Geographical Location Information of Web Pages. *In WebDB'99, (1999).*

[4] L. L. Hill, J. Frew, et Q. Zheng. 1999. "*Geographic Names: The Implementation of a Gazetteer in Georeferenced Digital Library*". In *CNRI D-Lib Magazine (January 1999).*

[5] J. Kazama et K. Torisawa .2007. Exploiting Wikipedia as External Knowledge for Named Entity Recognition. *In Proc.of the 2007 Joint Conference on Empirical Methods in Natural Language Processing et Computational Natural Language Learning, p. 698-707, (Prague, June 2007).*

[6] A. Popescu, G. Grefenstette et P.A. Moëllic.2008. Gazetiki: Automatic Creation of a Geographical Gazetteer, *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries.*

[7] A. Popescu, P.A. Moëllic et I. Kanellos.2008.ThemExplorer: Finding and Browsing Geo-Referenced Images, *International Workshop on Content-Based Multimedia Indexing, 2008. CBMI 2008.*

[8] T. Rattenbury, N. Good et M. Naaman. 2007. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. *In Proc.of SIGIR 2007 (Amsterdam, The Netherlands, July 2002).*

[9] M. Sanderson, Y.Han.2007. Search Words and Geography. *In GIR'2007, November 9, 2007, Lisbon, Portugal.*

[10] C.Toriani, S.Battle et S.Cayzer. 2006. Sharing, Discovering and Browsing Geotagged Pictures on the Web. *3rd Italian Semantic Web Workshop on Semantic Web Applications and Perspectives.*

Index des auteurs

Ahmed-ouamer, R. 433
Aïtelhadj, A. 301
Amghar, T. 115
Amini, M.-R. 83
Arour, K. 131
Ayache, S. 67
Bannour, H. 445,471
Bechir, A. 445
Beigbeder, M. 373, 397
Bellot, P. 251, 285
Besacier, L. 67
Bosc, P. 235
Bouamor, H. 519
Boughanem, M. 409, 433
Bouidghaghen, O. 479
Chappelier, J.-C. 267
Claveau, V. 235
Daille, B. 33
Daoud, M. 409
De Loupy, C. 285
Detyniecki, M. 83
Dyke, G. 397
Eckard, E. 267
El Ayari, S. 385
El-beze, M. 251
Fakeri-Tabrizi, A. 83
Fautsch, C. 19
Gallinari, P. 83
Gaussier, E. 99
Géry, M. 333
Girardot, J.-J. 397
Glotin, H. 421
Goeuriot, L. 33
Grau, B. 5, 385
Guinaudeau, C. 487
Hammache, A. 433
Hlaoua, L. 445
Hubert, G. 169
Hurault-Plantet, M. 351
Jacquemin, B. 351
Ke, G. 203
Kevers, L. 151
Kopliku, A. 495
Labiche, J. 457
Lamprier, S. 115
Largeron, C. 333
Lavalley, R. 251
Le, V. B. 67
Lecanu, J. 457
Levrat, B. 115
Ligozat, A.-L. 385
Lund, K. 397
Maisonnasse, L. 99
Marsala, C. 83
Mezghiche, M. 301, 363
M'hamed, M. 363
Moens, M. F. 2
Moriceau, V. 5
Morin, E. 33, 219
Mothe, J. 169
Mulhem, P. 67, 99, 319
Naulleau, E. 351
Peña Saldarriaga, S. 219
Peron, Y. 503
Pham, T.-T. 99
Pinel-Sauvagnat, K. 51
Pivert, O. 235
Portier, P.-E. 511
Quafafou, M. 421
Quénot, G. 67

Ralalason, B. 169
Ramanonjisoa, B. 169
Saidali, Y. 457
Saubion, F. 115
Savoy, J. 19, 185
Slimani, Y. 131
Smeaton, A. F. 1
Souam, F. 301
Tamine-Lechani, L. 409
Tan, T.-P. 67
Tannier, X. 5
Thollard, F. 333
Tollari, S. 83
Torjmen, M. 51
Trupin, E. 457
Ughetto, L. 235
Verbyst, D. 319
Viard-Gaudin, C. 219
Waszak, T. 285
Yeferny, T. 131
Zidouni, A. 421
Zweigenbaum, P. 203

Sixième Conférence Francophone Recherche d'Information et Applications

Nos partenaires

Université du Sud Toulon-Var

Orange Innovation

Laboratoire des Sciences de l'Information et des Systèmes

Centre National de la Recherche Scientifique

GDRI^3

Association Francophone de Recherche d'Information et Applications

Toulon Provence Méditerranée

Conseil régional PACA

ISBN 2-9524747-1-0
EAN 9782952474719
Editeur : LSIS-USTV