Modèle de langue par type de doxel pour l'indexation de documents structurés

Philippe Mulhem(1) et Jean-Pierre Chevallet(2)

Laboratoire d'Informatique de Grenoble (LIG)
(1) LIG - CNRS, (2) LIG - Université de Grenoble (Pierre Mendès France),
385 Av. de la bibliothèque, 38041 Grenoble Cedex 9
{Philippe.Mulhem,Jean-Pierre.Chevallet}@imag.fr

RÉSUMÉ. *Nous présentons dans cet article, une utilisation d'un modèle de langue pour une recherche d'information sur des documents structurés. Nous présentons également un usage de ressources exogènes et endogènes pour l'indexation et les requêtes. Les ressources endogènes sont les syntagmes extraits du corpus lui même, et les ressources exogènes proviennent de liens (forward links) extraits de Wikipedia. Nous montrons qu'un modèle de langue avec un lissage de type Dirichlet est plus adapté à la recherche d'information sur des documents structurés qu'un lissage par interpolation linéaire de Jelinek-Mercer. Finalement, nous utilisons cette correspondance probabiliste dans un schéma de type Fetch and Browse, allant du document complet vers des parties du document (doxels). Ces propositions nous ont permis d'obtenir 4 mesures en tête de l'évaluation officielle de la campagne INEX 2009, sur les huit mesures proposées cette année.*

ABSTRACT. *We present in this paper a language model for information retrieval on structured documents. We also present a use of endogenous and exogenous resources for indexing and querying. The endogenous resources are phrases taken from the corpus itself, and resources extracted from Wikipedia external links (forward links). We show that a language model with Dirichlet smoothing is more suited to information retrieval on structured documents than smoothing by linear interpolation (Jelinek-Mercer). Finally, we use this matching in a probabilistic scheme of type Fetch and Browse, from the full document to parts of the document (doxels). These proposals have produced 4 top tracks according to the INEX 2009 evaluation campaign, among the eight tracks proposed this year.*

MOTS-CLÉS : *documents structurés, ressources endogènes et exogènes, modèle de langue*

KEYWORDS: *structured documents, endogenous and exogenous ressources, language model*

1. Introduction

L'approche à base de modèles de langue pour la recherche de documents textuels a été proposée dès la fin des années 90 (Ponte *et al.*, 1998). Elle se base sur l'hypothèse qu'un document représente un modèle probabiliste de langue et qu'une requête est pertinente lorsqu'elle peut être *produite* par ce modèle. Ces modèles ont prouvé leur performance (Zhai *et al.*, 2001) face aux autres modèles. Bien évidemment, la taille des documents empêche d'estimer de manière fiable les probabilités d'un modèle de langue à partir d'un seul document. Pour cette raison, une autre source d'information est utilisée en complément du document et permet de *lisser* les probabilités des mots du document, utilisé comme termes d'indexation. Généralement, ce lissage utilise la collection elle même. Dans le cas où les documents à indexer sont structurés, se pose alors la question du choix de la source à utiliser pour ce lissage. En effet, les documents structurés sont représentés par une arborescence de *doxels*. Un doxel[1] est une sous partie d'un document structuré. Par exemple, dans le cas de documents codés en XML, un doxel est le sous arbre associé à une balise XML. Dans un Système de Recherche d'Information (SRI) sur des documents structurés, chaque sous arbre (doxel) est alors potentiellement une réponse. Ainsi, chaque doxel d'un document est aussi potentiellement un modèle de langue pour une requête. Dans ce cas, deux choix deviennent possibles pour le lissage : utiliser la collection entière, ou utiliser une partie de la collection en rapport avec le doxel considéré. La première solution correspond à l'usage habituel du modèle de langue pour une collection non structurée. La seconde solution tient compte de la structure des documents : nous proposons de l'étudier dans cet article.

Nous proposons donc de tenir compte de la structure des documents dans un modèle de langue en réalisant un lissage par *type de doxel*. Dans le cas de documents XML, le type d'un doxel est simplement le nom de sa balise. Notre approche des documents structurés s'intègre dans le cadre de *Fetch and Browse* (Chiaramella, 2007) lors du traitement des requêtes. Dans un premier temps le système se concentre sur les documents complets, et dans un second sur les parties de documents qui sont d'après le systèmes les plus pertinents.

Un doxel représente une partie structurelle d'un document (paragraphe, section, etc.). Il peut également représenter une partie non structurelle comme l'ancre d'un lien par exemple. Une relation de composition entre doxels dénote l'inclusion d'un doxel dans un autre. Nous appelons *type* t du doxel D le marqueur XML correspondant à D. Considérons un document XML :

```
<A> This is an <B> example </B> of <C> XML </C> document </A>
```

1. Il est possible de définir un doxel (à l'image d'un pixel) comme une partie non composée d'un document structuré, c'est à dire les feuilles d'un arbre de composition. Ce n'est pas notre choix dans cet article.

Ce document contient 3 doxels : le premier est délimité par la marqueur A, il est donc de type A, les deux autres sont de type B et C. Le doxel de type A est composé des deux doxels de type B et C.

Nous abordons également dans ce travail, l'usage de ressources extraites de la collection (endogènes), et extérieures à la collection (exogènes) pour construire des termes d'indexation à base de syntagmes et pour étendre les requêtes. L'utilisation de syntagmes nominaux pour la recherche d'information a prouvé son efficacité dans certains contextes, tels que les documents médicaux (Chevallet *et al.*, 2007). Nous avons expérimenté l'enrichissement de documents et de requêtes dans deux directions : tout d'abord, en utilisant l'annotation de Ralph Schenkel (Schenkel *et al.*, 2007) pour élargir le vocabulaire de syntagmes nominaux annoté et deuxièmement par l'utilisation des liens extraits de wikipedia pour étendre les requêtes utilisateur. Les expérimentations ont été menées sur la collection d'INEX 2009.

L'article est organisé comme suit : nous présentons le modèle de langue avec lissage sur le type de doxels en section 2. Dans la section 3 nous décrivons notre approche de type Fetch and Browse. La section 4 présente la construction des ressources endogènes et exogènes pour une indexation par syntagmes, et l'expansion des requêtes. Les résultats obtenus avec la collection INEX 2009 sont présentés en partie 5. Nous concluons en partie 6.

2. Modèle de langue pour l'indexation de Doxels

Un modèle de langue est une distribution de probabilité sur des séquences de mots. Le score RSV (Relevance Status Value) se calcule par la probabilité de produire la requête Q à partir d'un modèle de langue θ_D du document : $RSV(Q, D) = p(Q|\theta_D)$. Pour simplifier l'estimation de cette probabilité, on considère un modèle de langue *unigrame* : les mots w d'un document D sont produits aléatoirement indépendamment les uns des autres. Il s'agit alors d'un modèle multinomial sur les mots w du vocabulaire d'indexation V avec $c(w, Q)$ le nombre d'apparition du terme w dans la requête Q :

$$p(Q|\theta_D) = \prod_{w \in V} p(w|\theta_D)^{c(w,Q)} \quad [1]$$

Une manière simple d'estimer la probabilité d'un terme w sachant un modèle de document θ_D est d'estimer la probabilité maximale (maximum likelihood) qui se calcule par le rapport entre nombre d'occurrences de w dans D, $c(w, D)$ et la taille de D, $|D|$:

$$p_m(w|\theta_D) = \frac{c(w, D)}{\sum_{w \in V} c(w, D)} = \frac{c(w, D)}{|D|} \quad [2]$$

Malheureusement, la faible taille du document D n'assure pas un nombre d'occurrences non nul pour tout w : il faut ajouter de l'information d'une autre provenance. Une technique appelée *lissage* (smoothing) mélange la probabilité $p(w|C)$

du terme w estimé sur la collection C avec celle calculée avec le document. Par exemple le lissage de Jelinek-Mercer propose une combinaison linéaire d'ordre λ : $p_\lambda(w|\theta_D) = (1-\lambda)p_m(w|\theta_D) + \lambda p(w|C)$.

Le lissage introduit un biais extérieur au document D. On comprend facilement qu'il doit rester *discret* : si le document D est d'une taille importante, alors l'estimation $p_m(w|\theta_D)$ devient fiable et λ doit être le plus petit possible. Inversement, λ doit augmenter si la taille du document diminue.

Dans le cas d'une indexation structurée, deux problèmes apparaissent. D'une part, les doxels sont de taille très différentes : de la taille d'un document complet à la petite taille d'une partie de documents (quelque mots). Le facteur de lissage λ devrait alors être variable en fonction du doxel. D'autre part, réaliser un lisage systématiquement par rapport à la collection C tend à uniformiser tous les modèles, particulièrement les doxels de petites taille : ils ne différent que sur quelques mots. Nous proposons alors de réaliser un lissage par type de doxel D_t de type t sur la collection C_t composée uniquement de doxels de même types :

$$p_{\lambda_t,t}(w|\theta_{D_t}) = (1-\lambda_t)\frac{c(w,D_t)}{|D_t|} + \lambda_t p(w|C_t) \quad [3]$$

Cette approche nécessite de fixer autant de paramètres λ_t qu'il y a de types de doxels, ce qui n'est pas facile à obtenir de manière opérationnelle. Une autre solution consiste alors à faire dépendre le facteur de lissage λ_t de la taille du doxel $|D_t|$ tel que ce facteur soit d'autant plus faible que le document est de grande taille. Nous remplaçons alors tous les facteurs λ_t par un facteur unique dépendant de la taille $|D_t|$ du doxel et d'une seule constante μ : $\lambda_t = \frac{\mu}{|D_t|+\mu}$. L'équation [3] se transforme alors :

$$p_{\mu,t}(w|\theta_{D_t}) = (1-\frac{\mu}{|D_t|+\mu})\frac{c(w,D_t)}{|D_t|} + \frac{\mu}{|D_t|+\mu}p(w|C_t) \quad [4]$$

$$= \frac{c(w,D_t)+\mu p(w|C_t)}{|D_t|+\mu} \quad [5]$$

La formulation [5] correspond en fait à un lissage de type Dirichlet (Zhai, 2009) mais appliqué à chaque type de doxels pris comme une collection particulière. Lorsque nous utilisons la probabilité maximale pour estimer $p(w|C_t)$, l'interprétation de cette formule est la suivante : la grandeur μ représente alors une pseudo occurrence non entière du terme w dans le document $c_\mu(w,D_t)$ dont le total correspond à une taille μ qui augmente d'autant la taille du document : $\mu = \sum_{w\in V} c_\mu(w,D_t)$. L'estimation

$p(w|C_t)$ s'exprime alors comme la probabilité maximale de w selon cette pseudo occurrence :

$$p(w|C_t) = \frac{c_\mu(w, D_t)}{\sum_{w \in V} c_\mu(w, D_t)} \quad [6]$$

En remplaçant dans la formule [3], nous obtenons alors la formulation symétrique :

$$p_{\mu,t}(w|\theta_{D_t}) = \frac{c(w, D_t) + c_\mu(w, D_t)}{\sum_{w \in V} c(w, D_t) + \sum_{w \in V} c_\mu(w, D_t)} \quad [7]$$

Nous proposons alors d'expérimenter ce type de lissage qui nous paraît plus adapté à des documents structurés.

3. Evaluation hiérarchique de type *Fetch & Browse*

Le calcul de correspondance entre une requête et un doxel en utilisant le modèle de langue présenté précédemment, est plongé dans un cadre plus global de type *Fetch & Browse* (FB). Il s'agit d'un processus de recherche hiérarchique en plusieurs étapes qui raffine la réponse en sélectionnant des doxels plus bas dans la structure des documents donc de plus petite taille. Pour ce travail, nous nous sommes limité à un processus de raffinement ne faisant intervenir que deux types différents de doxels. Le cadre FB se compose de deux étapes :

Fetch : Cette étape réalise une recherche sur des doxels de taille importante. Elle produit un classement C des doxels dont le type t est dans un ensemble T. En pratique, nous avons utilisé le type unique racine des documents XML : $\{article\}$.

Browse : Cette étape est composée de deux parties : 1) un classement selon un autre ensemble de types de doxels T' sans intersection avec T. 2) un réordonancement et filtrage de ce résultat en fonction de l'étape Fetch. Ce filtrage se réalise selon des critères de relations structurelles entre doxels. Nous avons expérimenté les relations d'inclusion ou de chevauchement. Un doxel est inclus dans un autre s'il fait partie de sa structure. Deux doxels D et D' se chevauchent si l'on peut trouver un troisième doxel D'' strictement inclus dans D et D'. Le filtre supprime du classement C les doxels qui ne respectent pas le critère choisi. D'autre part, ce filtre respecte l'ordre initial et donne toujours la priorité sur les résultats en tête de liste : nous supprimons tous les doxels qui ont une intersection non vide avec un doxel déjà apparu en meilleure position dans la liste des résultats.

Nous avons expérimenté plusieurs configurations de notre évaluation hiérarchique. Pour décrire ces configurations, nous adoptons la notation suivante :

Fetch : Cette étape utilise un modèle de recherche d'information M comme un modèle de langue (LM), un prétraitement P sur la requête comme une extension

des requêtes. Elle se réalise sur des doxels de type $t \in T$ avec un vocabulaire V. Cette étape est donc identifiée par : $[M, P, T, V]$. Cela représente le classement C obtenu à cette étape.

Browse : Cette étape est composée d'une deuxième recherche notée de manière similaire : $[M', P', T', V']$, puis des opérations de reclassement et de filtrage utilisant le classement de la première étape : $[M, P, T, V]$. Les filtres sont notés comme des compositions de fonctions de modification de classements : $FB([M, P, T, V], [M', P', T', V'])$.

Pour nos expérimentations, nous avons testé des vocabulaires différents (avec sans sans syntagmes), ainsi que l'usage des liens de propagation pour l'expansion de requête.

4. Extraction et usage de ressources endogènes et exogènes

Nous utilisons le résultat de l'approche YAWN (Schenkel *et al.,* 2007) fourni avec le corpus INEX 2009[2]. Nous ne gardons que les syntagmes qui sont décrits par YAWN parce que nous considérons qu'ils sont dignes de confiance pour être de bons termes d'indexation. En fait, l'extraction des syntagmes identifiés par les balises YAWN constitue une *ressource endogène* que nous exploitons ensuite pour détecter toutes les occurrences de ces syntagmes dans le corpus pour les transformer en termes d'indexation atomiques. Dans la suite, nous nous référons au vocabulaire initial (mots-clés[3] seulement) par K, le vocabulaire de syntagmes[4] par P, et l'union de ces deux vocabulaire par $K\&P$.

Les documents originaux de Wikipedia possèdent une information qui n'a pas été reprise dans le corpus d'INEX : les liens de redirection. Ces liens permettent de rediriger une entrée de page sur une autre. Nous avons téléchargé la dernière version de Wikipédia en anglais, et nous avons extrait tous ces liens de redirections qui associent un terme à un ensemble de termes équivalents. Cela constitue notre *ressource exogène* qui est un graphe d'association non pondéré et non typé entre termes. Cette ressource nous permet de répondre correctement à une requête courte comme `VOIP` en ajoutant à cette requête la version développée de cet acronyme : `Voice over Internet Protocol`. Cette ressource ne modifie pas le modèle de recherche d'information, mais modifie uniquement les requêtes.

5. Experimentations et résultats

Nous avons travaillé sur la collection d'INEX 2009 qui est une transformation du contenu de Wikipédia. Cette collection contient un très grand nombre de balises, donc

2. http ://www.inex.otago.ac.nz
3. Keyword en anglais
4. Phrase en anglais

de types de doxels. Nous avons sélectionné manuellement un petit ensemble T de type de doxels : $T = \{article, ss, es1, es2, es3, es4\}$.

Nous avons indexé les doxels l'aide du modèle de langage avec lissage Dirichlet (noté LM) comme décrit dans la section 2. La construction des collections de doxels a été réalisée avec le système XIOTA (Chevallet, 2004) et l'indexation a été réalisée avec le système Zettair[5]. Les indexations se réalisent en fonction du vocabulaire K, ou $K\&P$. Dans nos expériences, nous avons travaillé sur deux séries de doxels potentiels : le doxels article complet, et les doxels de type T.

Nous avons intégré dans nos expérimentations des données de référence fournies par les organisateurs de la tâche. Il s'agit d'une indexation des mots isolés (vocabulaire K) avec le modèle probabiliste $BM25$ (Robertson *et al.*, 1994). Dans cette indexation, les doxels sont de type $articles$.

La tâche Ad Hoc d'INEX 2009 est composée de 4 tâches différentes :

Thorough : fournit une liste de documents ou de doxels par ordre de pertinence. Dans cette tâche basique, il n'y a aucune contrainte sur les doxels proposés par le SRI.

Focused : se focalise sur les doxels les plus petits mais qui répondent encore à la requête. Dans cette tâche, seuls les doxels sans chevauchement ni recouvrement sont autorisés ;

Relevance in Context : fournit le doxel pertinent le plus petit mais dans le contexte de son document. Elle nécessite donc de renvoyer les doxels regroupés par documents, sans recouvrement et en suivant l'ordre des doxels dans leur document. Le document tout entier n'a pas à faire partie de la liste des réponses.

Best in Context : simule le meilleur point d'entré dans un document. Elle impose donc de ne renvoyer qu'un seul doxel maximum par document.

Nous avons réalisé 3 expérimentations pour chacune de ces tâches. Le corpus d'INEX 2009 comporte environ 2,4 millions de documents pour un total d'environ 60 Go de texte. Nous décrivons dans suite le détails des traitements effectués, accompagné d'une courte analyse des résultats obtenus.

5.1. *Evaluation Thorough*

Pour la tâche Thorough, nous avons réalisé les trois expérimentations suivantes :

T1 $[LM, none, K, T]$: le modèle de langue (LM) sur les doxels (T), avec les mots clés (K), sans traitement FB, ni traitement des requêtes (noté $none$). Cette expérience est considérée comme une base de comparaison.

T2 $FB([LM, none, K, \{article\}], [LM, none, K, T - \{article\}])$: mise en œuvre du processus Fetch & Browse. Les articles sont indexés et interrogés séparément et forment le premier classement (Fetch). Les doxels de types différents de $article$ sont indexés, interrogés séparément puis regroupés pour former le

5. http ://www.seg.rmit.edu.au/zettair

Tableau 1. *Résultats Thourough*

Run	ip[0.00]	ip[0.01]	ip[0.05]	ip[0.10]	MAiP (rang / 30)	MAiP Doc. (rang / 30)
T1	0.588	0.582	0.553	0.504	0.286 (1)	0.349 (3)
T2	0.586	0.581	0.552	0.501	0.278 (4)	0.347 (4)
T3	0.597	0.584	0.544	0.502	0.285 (2)	0.339 (5)

deuxième classement. Finalement les doxels de la deuxième interrogation sont regroupés par les articles de la première interrogation (Browse) pour produire le classement final.

T3 $FB([BM25, none, K, \{article\}], [LM, none, K, T - \{article\}])$: similaire à **T2** mais utilise BM25 pour l'étape fetch.

Le tableau 1 présente les résultats obtenus. Les valeurs **ip** représentent la précision interpolée sur toutes les requêtes à une valeur fixe de rappel. La valeur **MAiP**[6] représente la précision moyenne interpolée sur 101 points de rappel (Geva *et al.*, 2009). La valeur **MAiP Doc** représente la précision moyenne **MAiP** en considérant uniquement les documents complets (de type $article$). Le rang associé à ces deux valeurs est le rang officiel de notre participation à INEX 2009. Ces résultats montrent en premier l'excellence performance du modèle de langue. Notre technique de Fetch & Browse est par contre moins concluante car elle n'est pas en première place comme nous l'espérions. Nous remarquons également que dans ce cas le modèle probabiliste BM25 est meilleur que le modèle de langue. Ce qui est également contraire à nos estimations.

5.2. ***Evaluation*** *Focused*

L'évaluation *Focused* concerne les éléments les plus précis liés à un besoin d'information, sans permettre de recouvrement entre doxels résultats. Nous avons réalisé les expériences suivantes :

F1 $RO(FB([LM, none, K, \{article\}], [LM, none, K, T - \{article\}]))$: similaire à **T2** avec retrait des recouvrements RO (Remove Overlap).

F2 $RO(FB([LM, E, K\&P, \{article\}[, [LM, none, K, T - \{article\}]))$: utilisation de syntagmes ($K\&P$) et expansion de requêtes E uniquement dans la partie Fetch.

F3 $RO(FB([BM25, none, K, \{article\}], [LM, none, K, T - \{article\}]))$: similaire à **T3** avec suppression des recouvrements.

Les résultats obtenus en se basant sur une évaluation à base de doxels, montrent que l'utilisation des expansions donne de moins bons résultats que les approches de

6. mean average interpolated precision

Tableau 2. *Résultats Focused*

Run	ip[0.00]	ip[0.01] (rang / 57)	ip[0.05]	ip[0.10]	MAiP	MAiP Doc. (rang / 62)
F1	0.586	0.585 (20)	0.543	0.506	0.270	0.351 (1)
F2	0.564	0.544 (30)	0.498	0.461	0.240	0.300 (33)
F3	0.595	0.582 (23)	0.534	0.501	0.273	0.341 (6)

Tableau 3. *Relevant In Context Task for INEX2009 Ad Hoc.*

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP (rank / 33)	MAP Doc. (rank / 42)
R1	0.303	0.260	0.206	0.155	0.176 (12)	0.357 (1)
R2	0.195	0.175	0.138	0.107	0.093 (28)	0.170 (41)
R3	0.312	0.279	0.219	0.163	0.176 (13)	0.346 (9)

base, ce qui semble induire que l'utilisation conjointe des syntagmes et de l'expansion de requête ne sont pas appropriés. Il est probable que ces résultats décevants proviennent du bruit induit par l'expansion de requête. En effet, nous avons réalisé une expansion par mots clés, alors qu'une expansion par syntagme est probablement plus précise. Ici également, le modèle de langue se comporte de manière similaire au modèle de référence à base de BM25. Les différences ne sont pas significatives.

5.3. ***Evaluation*** *Relevant In Context*

Pour l'évaluation *Relevant In Context*, nous utilisons les résultats *Focused* par défaut, mais nous ordonnons les doxels d'un même document par ordre de lecture avant de traiter les recouvrements. Nous avons réalisé 3 expériences :

R1 $RO(FB_r([LM, none, K, \{article\}], [LM, none, K, T-\{article\}]))$: similaire à **F1** mais avec l'opération FB_r de Fetch & Browse qui respecte l'ordre initial de lecture (reading) des doxels dans le document.

R2 $RO(FB_r([LM, E, K\&P, \{article\}], [LM, none, K, T - \{article\}]))$: les même modifications que **R1** mais appliqué à **F2**.

R3 $RO(FB_r([BM25, none, K, \{article\}], [LM, none, K, T - \{article\}]))$: les même modifications que **R1** mais appliqué à **F3**.

Les résultats des évaluations officielles sont présentés dans le tableau 3. Dans le cas de cette recherche en contexte, les meilleurs résultats sont obtenus par l'expérience **R1**. Ces résultats obtenus ici sont donc meilleurs en utilisant un modèle de langue que le modèle probabiliste BM25. L'expérience **R2** qui utilise l'expansion des requêtes par la ressource exogène et une indexation par syntagme par la ressource endogène produit à nouveau des résultats inférieurs, comme nous l'avons déjà remarqué précédemment.

Tableau 4. *Résultats Best In Context.*

Run	gP[5]	gP[10]	gP[25]	gP[50]	MAgP (rank / 37)	MAiP Doc. (rank / 38)
B1	0.250	0.230	0.185	0.141	0.157 (17)	0.357 (1)
B2	0.174	0.163	0.130	0.104	0.088 (31)	0.170 (38)
B3	0.278	0.256	0.197	0.147	0.157 (16)	0.346 (7)

5.4. ***Expérimentation*** *Best In Context*

Pour ces expérimentations, nous utilisons comme base les résultats de *Focused* (**F1**, **R2**, **R3**) mais nous réalisons le traitement KB pour "keep best", qui conserve le doxel ayant la plus grande valeur de pertinence comme meilleur point d'entrée. Les évaluations officielles sont présentées en table 4.

B1 $KB(FB([LM, none, K, \{article\}], [LM, none, K, T - \{article\}]))$

B2 $KB(FB_r([LM, E, K\&P, \{article\}], [LM, none, K, T - \{article\}]))$

B3 $KB(FB_r([BM25, none, K\ \{article\}], [LM, none, K, T - \{article\}]))$

Ces expériences ont fournit de bons résultats mais ne permettent pas de différencier le modèle de langue du modèle BM25. Comme dans les autres expériences, l'extension des requêtes a mal fonctionné.

5.5. ***Discussion***

Nous avons utilisé pour la première fois un modèle de langue avec un lissage par type de doxels en remplacement du modèle BM25 largement plébiscité par les participants à INEX des années précédentes. Même si nous avons devancé que très légèrement le modèle BM25, nous sommes tout de même très satisfait d'obtenir de très bon résultats avec ce modèle de langue. La formulation du lissage par type de doxels montre qu'il est plus adapté dans sa formulation, à une indexation de documents structurés. Avec une optimisation du paramètre μ, ou un lissage mixte, avec le type de doxel et le corpus entier, nous pensons qu'il existe encore une marge d'amélioration.

Indépendamment du modèle de correspondance, notre technique de Fetch & Browse fonctionne très bien puisque qu'une comparaison avec les autres participants sur des expériences basées sur le même classement avec BM25, 3 de nos expériences sur les 4 (**T3, B3, F3**) arrivent en tête par rapport aux autres participants (voir (Geva *et al.*, 2009) page 40).

Par contre, l'utilisation de la ressource exogène dans l'extension des requêtes n'a pas amélioré les résultats. Nous avons réalisé une extension au niveau des mots clés et sans restriction : cela a trop changé les requêtes et ajouté du bruit. Ceci explique très probablement la baisse des résultats. L'expansion de requête génère trop de mots supplémentaires pour être réellement efficace. Par exemple, la requête 2009_005 :

Tableau 5. *Comparaisons par requêtes pour la tâche Relevant In Context.*

Requête	gP[10]			AgP		
	R1	R2	R3	R1	R2	R3
2009_001	0.279	0.379	0.279	0.268	0.280	0.314
2009_015	0.686	0.687	0.686	0.395	0.295	0.421
2009_026	0.110	0.170	0.110	0.284	0.273	0.307
2009_029	0.736	0.926	0.636	0.369	0.439	0.287
2009_069	0.299	0.199	0.500	0.289	0.332	0.285
2009_088	0.427	0.391	0.462	0.429	0.450	0.486
2009_091	0.099	0.653	0.182	0.189	0.511	0.221
Moyenne	0.377 (+22.6%)	0.487	0.408 (+16.2%)	0.318 (+13.8%)	0.369	0.332 (+10.0%)

```
chemists physicists scientists alchemists periodic table elements
```

est étendue en une nouvelle requête très longue :

```
chemists chemist periodic_table periodic table elements chemical
periodic system mendeleev properties natural element symbol
list groups representative chart peroidic periodicity elements
organization fourth period patterns group 2a nuclear
```

Cette requête élargie contient le syntagme `periodic_table` et le mot `Mendeleev`, qui sont de très bons termes pour la requête. Mais d'un autre côté, nous constatons sur cet exemple que beaucoup trop de mots restent sans rapport avec la requête initiale, comme : `representative organization system properties fourth`.

Ainsi, le principal problème auquel nous sommes confrontés ici est que l'expansion des requêtes ajoute trop de mots parasites pour être vraiment utile. C'est à notre avis ce qui explique les mauvais résultats.

L'examen détaillé des résultats de certaines requêtes avec ou sans expansion, nous indique que nous sommes tout de même sur une voie prometteuse. En examinant les sept meilleurs résultats obtenus pour notre expansion (relevant in contexte) (soit 10 % des 69 requêtes d'INEX 2009), nous obtenons les résultats présentés dans le tableau 5. Par exemple, la requête 091, `Himalaya trekking peak`, a été transformée en `trekking_peak himalaya himalayas` et permet d'augmenter nettement les résultats par l'utilisation du syntagme nominal `trekking_peak` au lieu des deux termes `trekking` et `peak`. Sur ces quelques requêtes (tableau 5) l'amélioration est donc très nette : l'expérience **R2** qui utilise l'extension permet d'atteindre sur ces requêtes jusqu'à 22% d'amélioration de la précision des réponses.

6. Conclusion

Nous avons montré qu'un modèle de langue avec un lissage de type Dirichlet part type de doxel permet de mieux s'adapter à une indexation de documents structurés par la prise en compte explicite de la taille des doxels. Ce résultat s'est reflété dans les résultats de notre participation à la campagne INEX 2009, où certaines de nos expériences arrivent en tête (4 évaluations officielles sur les 8). Notre méthode de Fetch & Browse est également une bonne idée car elle nous place en tête des résultats utilisant le classement de référence fournit par les organisateurs d'INEX et basé sur le modèle BM25.

Par contre, l'utilisation des ressources endogènes et exogènes n'ont pas donné globalement de bons résultats, même si sur quelques requêtes, le gain est très significatif. Cela nous laisse espérer une marge d'amélioration importante, si bien sur, nous trouvons un moyen de maîtriser cette expansion en limitant le bruit produit par des termes hors du sujet de la requête initiale.

7. Bibliographie

Chevallet J.-P., « X-IOTA Une plateforme distribuée ouverte pour l'expérimentation en Recherche d'Information », *COnférence en Recherche Information et Applications CORIA'2004*, Toulouse, p. 287-304, mar, 2004.

Chevallet J.-P., Lim J. H., Le T. H. D., « Domain Knowledge Conceptual Inter-Media Indexing, Application to Multilingual Multimedia Medical Reports », *ACM Sixteenth Conference on Information and Knowledge Management (CIKM 2007), Lisboa, Portugal*, November 6–9, 2007.

Chiaramella Y., « Information Retrieval and Structured Documents », *Lecture Notes in Computer Science, Lectures on Information Retrieval, LNCS 1980/2001*, p. 286-309, 2007.

Geva S., Kamps J., Lethonen M., Schenkel R., andAndrew Trotma J. A. T., « Overview of the INEX 2009 Ad Hoc Track », *INEX 2009 Workshop Pre-proceedings*, p. 16-50, dec, 2009.

Ponte J. M., Croft W. B., « A Language Modeling Approach to Information Retrieval », *Research and Development in Information Retrieval*, 1998.

Robertson S., Walker S., Jones S., Hancock-Beaulieu M., Gatford M., « Okapi at TREC-3 », *The Third Text REtrieval Conference (TREC 1994)*, p. 109-126, nov, 1994.

Schenkel R., Suchanek F. M., Kasneci G., « YAWN : A semantically annotated Wikipedia XML corpus », *2. GI-Fachtagung fur Datenbanksysteme in Business, Technologie und Web (BTW 2007)*, p. 277-291, 2007.

Zhai C., Lafferty J., « A study of smoothing methods for language models applied to Ad Hoc information retrieval », *SIGIR '01 : Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, NY, USA, p. 334-342, 2001.

Zhai C. X., *Statistical Language Models for Information Retrieval*, Morgan & Claypool, 2009.