HAL
open science

BM25t : une extension de BM25 pour la Recherche d'Information ciblée

Mathias Géry, Christine Largeron, Thollard Franck

To cite this version:

Mathias Géry, Christine Largeron, Thollard Franck. BM25t : une extension de BM25 pour la Recherche d'Information ciblée. Document numérique - Revue des sciences et technologies de l'information. Série Document numérique, 2010, 13 (1), pp.83-110. hal-00531162

HAL Id: hal-00531162

https://hal.science/hal-00531162v1

Submitted on 2 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

BM25t : une extension de BM25 pour la Recherche d'Information ciblée

Mathias Géry⋆ — Christine Largeron⋆ — Franck Thollard†

⋆ *Université de Lyon, F-42023, Saint-Étienne, France*
CNRS, UMR 5516, Laboratoire Hubert Curien, F-42000, Saint-Étienne, France
Université de Saint-Étienne, Jean-Monnet, F-42000, Saint-Étienne, France
{mathias.gery, christine.largeron}@univ-st-etienne.fr

† *Laboratoire d'Informatique de Grenoble UJF-CNRS - 38041 Grenoble Cedex 9, France — Franck.Thollard@imag.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 a term-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 the tags distinguish relevant terms. Then, these weights are integrated in a term-weighting function. Experiments on a large collection during the INEX 2008 XML IR evaluation campaign showed improvements on focused XML retrieval.*

MOTS-CLÉS : *Modèle probabiliste de document, Recherche d'information structurée, XML, Balises, Pondération, BM25*

KEYWORDS: *Probabilistic IR model, Structured IR, XML, Tags, Weighting, BM25*

1. Introduction

Avec le développement des langages de balises tels que HTML ou XML, les documents disponibles sur Internet sont le plus souvent structurés. C'est probablement ce qui a conduit au développement de la recherche d'information ciblée (*focused information retrieval*) dont l'objectif est de fournir à l'utilisateur des extraits de documents plutôt que des documents entiers, comme c'est le cas en général, en recherche d'information. Ainsi, l'information pertinente est repérée à l'intérieur du document, ce qui est particulièrement utile lorsque ce dernier est long. Selon que la liste renvoyée à l'utilisateur est formée de passages de documents ou d'éléments XML les composant, on parle plus spécifiquement de recherche de passage ou de recherche XML (Lalmas, 2009b). Les workshops et les compétitions telles que INEX[1] (Baeza-Yates *et al.*, 2000; Baeza-Yates *et al.*, 2006; Trotman *et al.*, 2007; Fuhr *et al.*, 2008; Geva *et al.*, 2008; Kamps *et al.*, 2007) ont largement contribué au développement de la recherche XML ces dernières années. Dans ce cadre, les balises des langages ne servent pas seulement à décomposer un document en éléments, elles permettent aussi d'annoter le document de façon à décrire sa structure logique, sa mise en forme ou encore sa présentation indépendamment de son contenu. Dès lors, les travaux réalisés en recherche XML se sont attachés non seulement à identifier des unités d'information plus concises mais aussi à exploiter ces balises afin de détecter les informations répondant de façon plus pertinente à un besoin d'information.

Dans cette optique, deux types d'approches ont été développées. Le premier, orienté utilisateur, a consisté à développer d'une part des interfaces de visualisation ou de navigation dans la liste des résultats renvoyés par le système et d'autre part des langages de requêtes tels que W3QS (Konopnicki *et al.*, 1995), XIRQL (Fuhr *et al.*, 2000; Fuhr *et al.*, 2001), NEXI (Trotman *et al.*, 2004a; Trotman *et al.*, 2004b) ou Bricks (van Zwol *et al.*, 2006) permettant à l'utilisateur d'exprimer sa requête en tenant compte de la structure. Cependant, l'usage de tels langages est resté limité car, dans la pratique, peu d'utilisateurs sont capables d'exprimer leur besoin d'information sous forme de requêtes complexes[2]. Le plus souvent, celles-ci sont réduites à quelques mots-clés (O'Keefe *et al.*, 2003; Kamps *et al.*, 2005b; Kazai *et al.*, 2007).

Le second type d'approches explorées revisite les modèles classiques en proposant un schéma de pondération de la structure (Fuller *et al.*, 1993; Lalmas, 2009a). Le poids affecté alors à un mot ne dépend pas seulement de sa fréquence dans le document et éventuellement dans la collection mais aussi de sa position dans le document. Les balises sont utilisées pour définir cette position. Ainsi, le classement ne dépend pas seulement de la présence d'un mot dans le document mais de la présence du mot marqué par la balise appropriée. Différents types de balises peuvent être considérées : les balises de mise en forme (*ex.* gras, italique, centré, etc.) et les balises logiques

1. INitiative for Evaluation of XML Retrieval : `http://www.inex.otago.ac.nz`
2. Par 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".

qui définissent la structure interne du document (*ex.* titre, section, paragraphe, etc.) ou permettent de la représenter sous forme d'arbre.

Dans cette perspective, nous proposons une extension du modèle probabiliste (Maron *et al.*, 1960; Robertson *et al.*, 1976) qui évalue la pertinence d'un document pour une requête donnée à travers deux probabilités : la probabilité de trouver une information pertinente et celle de trouver une information non pertinente. Cette extension exploite les balises de mise en forme et les balises de structure logique. En effet, nous faisons l'hypothèse que ces deux types de balises peuvent être employés pour souligner certains mots. Ainsi, un mot n'aura pas la même importance s'il apparaît dans une certaine fonte de caractères (gras, italique, taille, etc.). De même, il n'aura pas le même poids selon sa localisation dans le document, par exemple dans le titre, dans la légende d'une figure ou encore dans une sous-section.

Pour une balise donnée (de structure ou de mise en forme), nous évaluons à l'aide d'un modèle basé sur un apprentissage, si elle met en évidence des termes dans des éléments pertinents ou au contraire des termes dans des éléments non pertinents.

La première contribution de cet article[3] est la proposition d'un cadre formel, présenté dans la section 3, prenant en compte explicitement la structure du document. Ce modèle est décrit après un état de l'art, donné dans la section suivante. La seconde contribution consiste en une expérimentation du modèle, présentée dans la section 4, sur une collection d'envergure (la collection INEX).

2. État de l'art

La pertinence d'un document pour une requête donnée est généralement évaluée en fonction du poids des mots de la requête. Ce poids dépend de la fréquence d'apparition du mot dans le document et, éventuellement de sa fréquence d'apparition dans la collection. Par rapport à ce modèle classique, la prise en compte de la structure dans le schéma de pondération consiste à attribuer aussi un poids aux balises en fonction de leur importance. Ce poids est ensuite combiné à celui des mots pour déterminer la pertinence d'un document. Ainsi, la pertinence d'un document ne dépend plus seulement de la fréquence avec laquelle les termes de la requête apparaissent mais aussi en fonction du poids des balises qui marquent ces termes dans le document.

Ce principe a déjà été utilisé dans le contexte de la recherche d'information classique (Lalmas, 2009a). Les balises considérées, de même que leur poids, peuvent être choisis de façon empirique. Par exemple dans (Rapela, 2001), le poids de la balise *title* est fixé à 2 et celui de la balise *abstract* à 1,5. Une alternative consiste à apprendre de façon automatique ces poids, à l'aide par exemple d'algorithmes génétiques (Kim *et al.*, 2000; Trotman, 2005) ou grâce à des techniques d'optimisation basées sur le recuit simulé (*simulated annealing*) (Boyan *et al.*, 1996).

3. Ce travail a été soutenu par le projet "Web Intelligence" de la région Rhône-Alpes.

Une fois les poids des balises déterminés, que ce soit empiriquement ou par apprentissage automatique, il reste à les combiner avec les poids des mots.
Dans le cas où seules des balises de structure logique sont prises en compte, la solution la plus simple consiste à les combiner de façon ad hoc. Dans ce cas, le document peut être divisé en autant d'éléments qu'il y a de parties définies par ces balises (résumé, introduction, etc.) et, chacune peut être traitée indépendamment des autres. Ensuite, le score attribué au document peut être calculé par combinaison linéaire des scores accordés à chaque partie en respectant leurs poids respectifs. Cependant, dans le cas du modèle BM25, (Robertson *et al.*, 2004) a démontré qu'il pouvait être plus avantageux de dupliquer chaque partie autant de fois que son poids le requiert puis de traiter de façon usuelle le document ainsi obtenu. Ainsi, par exemple, un document structuré dont le titre a un poids égal à deux, sera transformé en un document plat dont le contenu du titre aura été dupliqué, et qui sera traité de manière classique. Les évaluations expérimentales réalisées par (Robertson *et al.*, 2004) ont confirmé que cette approche permettait d'obtenir de meilleurs résultats qu'une simple combinaison linéaire des scores de chacune des parties. Par contre, dans les travaux cités précédemment, les systèmes visaient à retourner des documents complets et aucune évaluation n'en a été faite dans un contexte de recherche d'information ciblée.

Dans le cadre de la recherche d'information ciblée, les schémas de pondération ont aussi été employés pour prendre en compte la structure. Une fois les poids des balises déterminés, le principe utilisé est basé sur le produit scalaire : les poids des balises sont employés comme facteurs multiplicatifs des poids des mots qu'elles marquent. Cette approche a été employée pour améliorer le modèle probabiliste (Wolff *et al.*, 2000; Lu *et al.*, 2005) aussi bien que le modèle vectoriel (Wilkinson, 1994). Cependant, dans ces travaux, les poids des balises devaient être fixés empiriquement.

Dans le contexte de la recherche d'information ciblée, d'autres études ont cherché à exploiter la structure en considérant la représentation arborescente des documents XML (Kotsakis, 2002; Schlieder *et al.*, 2002; Trotman, 2005). Chaque élément XML, correspondant à un nœud de l'arbre, peut être caractérisé par le chemin allant de la racine de l'arbre jusqu'à lui. Un poids est ensuite fixé pour chaque chemin et, la structure est prise en compte pour chaque mot en considérant aussi le poids du chemin de l'élément qui le contient. De cette manière, (Kotsakis, 2002) attribue un poids à chaque type de chemin, de sorte que le poids d'une occurrence d'un mot situé dans l'élément (*i.e.* nœud) journal/issue/article/title sera supérieur à celui d'une occurrence du même mot placé dans l'élément journal/ issue/article/abstract. En adoptant également ce principe, (Schlieder *et al.*, 2002) a introduit une version étendue du modèle vectoriel. Ainsi, le poids final d'un mot comporte deux composantes : la première, calculée suivant la formule classique tf.idf, tandis que la seconde correspond au poids associé à la position du mot dans l'arbre XML *i.e.* au poids du chemin jusqu'à ce nœud. Les modalités de calcul des poids structurels ne sont pas détaillés dans (Kotsakis, 2002), alors que (Trotman, 2005) propose quant à lui de les apprendre automatiquement à l'aide d'un algorithme génétique. Les expérimentations rapportées par ce dernier ont montré une amélioration des résultats obtenus avec les modèles vectoriel et probabiliste mais aucune amélioration avec le modèle BM25.

Par contre, le modèle BM25E, introduit par (Lu *et al.*, 2005) a fourni des résultats encourageants dans le contexte de la recherche d'information ciblée. Il s'agit probablement du modèle le plus proche de celui que nous proposons dans la mesure où le score attribué à un élément est obtenu en effectuant une combinaison précoce des poids des mots de la requête figurant dans cet élément avec ceux des balises qui marquent ces mots. Par contre, dans (Lu *et al.*, 2005) les poids des balises sont déterminés empiriquement alors qu'ils sont appris automatiquement dans notre modèle. De plus, comme dans la majorité des travaux antérieurs, très peu de balises sont considérées (en général moins de 5) et celles ci sont le plus souvent choisies arbitrairement. C'est probablement la raison pour laquelle (Lu *et al.*, 2005) note que *"the creation of a practical algorithm to generate values for tuning parameters at the element level is a challenging task"*. C'est dans cette optique que nous avons entrepris cette recherche.

Dans le modèle que nous avons développé, la structure des documents est exploitée à deux niveaux :

1) **S**tructure logique : les balises de structure logique sont utilisées 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) **S**tructure de mise en forme : les balises de structure logique et les balises de mise en forme sont intégrées au niveau du schéma de pondération. Le poids de chacune des balises est estimé par apprentissage. Ce poids est basé sur la probabilité que la balise mette en exergue un terme pertinent ou au contraire un terme non pertinent. Ceci rejoint les principes du modèle probabiliste (Robertson *et al.*, 1976) 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).

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.

Ainsi, les balises de mise en forme sont considérées lors du calcul du score d'un élément, et les balises de structure logique sont considérées en plus lors de l'indexation.

Notre approche se caractérise donc 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 (Robertson *et al.*, 2004).

– 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 balises.

– L'extension de la fonction de pondération BM25 (Robertson *et al.*, 1976) via l'intégration d'un apprentissage automatique du poids des balises.

– 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 (Trotman, 2005).

Une présentation plus formelle de ce modèle est donnée dans la section suivante.

3. Un modèle probabiliste pour la représentation de documents structurés

3.1. *Notations et exemples*

Soit $\mathcal{D}$ un ensemble de documents structurés. Sans perte de généralité, nous considérerons des documents XML. Chaque balise XML décrivant la structure logique (*article*, *section*, *p*, *table*, etc.) définit un élément XML qui correspond à une partie du document. En conséquence, chaque élément logique (*article*, *section*, *paragraphe*, *table*, etc.) sera représenté par un ensemble de termes et sera indexé.

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 considérons la balise comme une balise de mise en forme et non comme une balise logique : elle ne définit donc pas un élément logique à indexer.

On note :

– $E = \{e_1, ..., e_j, ..., e_l\}$, l'ensemble des éléments logiques disponibles dans la collection (*article*, *section*, *p*, *table*, 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.

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.

3.2. *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 suppose que le poids de t_i dans Q est égal à 1.

On définit X_j un vecteur de variables aléatoires et $x_j = (x_{j1}, ..., x_{ji}, ..., x_{jn})$ 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 :

$$f_{term}(x_j) = \sum_{t_i \in T \cap Q} x_{ji} \times w_{ji} \qquad [1]$$

Sous ce produit scalaire général se cachent différentes fonctions de pondération, comme par exemple les fonctions `ltn`, `ltc` implantées dans le système SMART (Salton *et al.*, 1983), ou la fonction BM25 (Robertson *et al.*, 1976).

Des expérimentations antérieures (Géry *et al.*, 2008) avec `ltn` et `ltc` ayant donné des résultats médiocres relativement à ceux obtenus avec BM25, nous ne considérerons par la suite que BM25 :

$$w_{ji} = \frac{tf_{ji} \times (k_1 + 1)}{k_1 \times ((1 - b) + (b * ndl)) + tf_{ji}} \times \log \frac{N - df_i + 0,5}{df_i + 0,5} \qquad [2]$$

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 (en nombre de termes) ;

– k_1 et b : les paramètres classiques de BM25.

Le paramètre k_1 permet de régler la saturation de tf_{ji}, et le paramètre b permet de régler l'importance accordée à ndl, c'est-à-dire l'importance de la normalisation de la taille des éléments. Notons que la modification des paramètres k_1 et b permet de faire de BM25 une fonction non linéaire en la fréquence des termes. Par exemple, si $k_1 = 1,1$ un tf_{ji} égal à 10 donnera quasiment la même valeur pour la partie tf de BM25 qu'un tf_{ji} égal à 25. Cette propriété de non-linéarité de la fonction de pondération est très importante dans notre modèle : en effet, le résultat est très différent si l'on intègre le poids d'une balise sur tf_{ji} ou directement sur w_{ji}. Dans le cadre considéré par Robertson et *al.* (Robertson *et al.*, 2004) il est apparu important de ne pas violer la propriété de non linéarité de BM25. Nous avons donc comparé une stratégie d'impact précoce du poids des balises (sur tf_{ji}) avec une stratégie d'impact tardif (directement sur w_{ji}).

3.3. *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) pour une requête donnée. En suivant les principes du modèle probabiliste de RI (Robertson *et al.*, 1976), on veut donc estimer :

$P(R\|m_j)$:	la probabilité de trouver une information pertinente (R) étant donné l'élément m_j et une requête.
$P(NR\|m_j)$:	la probabilité de trouver une information non pertinente (NR) étant donné l'élément m_j et une requête.

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 :

$$P(M_j = m_j|R) = \prod_{t_{ik} \in m_j} P(T_{ik} = t_{ik}|R) \quad [3]$$
$$= \prod_{t_{ik} \in m_j} P(T_{ik} = 1|R)^{t_{ik}} P(T_{ik} = 0|R)^{1-t_{ik}}$$

$$P(M_j = m_j|NR) = \prod_{t_{ik} \in m_j} P(T_{ik} = 1|NR)^{t_{ik}} P(T_{ik} = 0|NR)^{1-t_{ik}} \quad [4]$$

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 :

$$P(m_j|R) = \prod_{t_{ik} \in m_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}$$
$$P(m_j|NR) = \prod_{t_{ik} \in m_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}$$

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 :

$$\begin{aligned} f_3(m_j) &= \log(f_2(m_j)) \\ &= \sum_{t_{ik} \in m_j} (t_{ik} \log(p_{ik}) + (1 - t_{ik}) \log(1 - p_{ik}) \\ &\qquad -t_{ik} \log(q_{ik}) - (1 - t_{ik}) \log(1 - q_{ik})) \\ &= \sum_{t_{ik} \in m_j} t_{ik} \times \left(\log\left(\frac{p_{ik}}{1 - p_{ik}}\right) - \log\left(\frac{q_{ik}}{1 - q_{ik}}\right)\right) + \sum_{t_{ik} \in m_j} \log\left(\frac{1 - p_{ik}}{1 - q_{ik}}\right) \end{aligned}$$

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 :

$$f_{tag}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} t_{ik} \times \log\left(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}\right) \qquad [5]$$

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} :

$$w'_{ik} = \log\left(\frac{p_{ik}(1-q_{ik})}{q_{ik}(1-p_{ik})}\right) \qquad [6]$$

La pertinence d'un élément XML m_j relativement aux balises est définie par $f_{tag}(m_j)$:

$$f_{tag}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} t_{ik} \times w'_{ik} \qquad [7]$$

Cette formule est similaire à celle de la fonction de pondération classique (équation 1), sauf que les poids des balises remplacent ici les poids des termes.

En 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

Avec :

– 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 $\begin{cases} p_{ik} = & P(t_{ik} = 1|R) & = \frac{r_{ik}}{R} \\ q_{ik} = & P(t_{ik} = 1|NR) & = \frac{n_{ik} - r_{ik}}{N-R} \end{cases}$

Il vient w'_{ik} :

$$\begin{aligned} w'_{ik} &= \log \frac{\frac{r_{ik}}{R}(1 - \frac{n_{ik}-r_{ik}}{N-R})}{\frac{n_{ik}-r_{ik}}{N-R}(1 - \frac{r_{ik}}{R})} \\ &= \log \frac{r_{ik} \times (N - n_{ik} - R + r_{ik})}{(n_{ik} - r_{ik}) * (R - r_{ik})} \\ &= \log \frac{r_{ik} \times (|NR| - nr_{ik})}{nr_{ik} \times (R - r_{ik})} \end{aligned} \quad [8]$$

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.

3.4. *Estimation du poids des balises*

D'un point de vue théorique, nous pouvons estimer un poids pour chaque paire (terme, balise) (cf. équation 8), c'est-à-dire la capacité pour une balise donnée de renforcer un terme donné (ou, au contraire, d'atténuer un terme). 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 au poids de chaque paire (terme-balise), mais au poids d'une balise indépendamment des termes qu'elle étiquette. Nous estimons donc finalement pour chaque balise b_k un poids global w'_k :

$$w'_k = \frac{\sum_{t_i \in T} w'_{ik}}{|T|} \quad [9]$$

3.5. *Score de pertinence global d'un élément XML*

À partir des poids des termes et des balises, nous pouvons calculer un score global des éléments. Nous proposons deux stratégies d'intégration du poids des balises dans la fonction de pondération BM25 :

– CLAW[4] : stratégie d'impact tardif sur le résultat de BM25.

4. f_{claw} : Combining Linearly Average tag-Weights.

– TTF[5] : stratégie d'impact précoce, intégrant le poids des balises dans BM25.

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}, s'écrit comme suit :

$$f_{claw}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} w_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \qquad [10]$$

avec w_{ji} le poids du terme t_i dans le document m_j, calculé à l'aide de BM25.

Dans (Géry *et al.*, 2008), l'intégration du poids des balises à l'aide de f_{claw} permet d'améliorer le rappel, mais de manière peu significative. La fonction de pondération BM25 étant non linéaire (cf. section 3.2), 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.*, 2004), 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 BM25.

$$ttf_{ji} = tf_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \qquad [11]$$

4. Expérimentations

Nos expérimentations ont été menées dans le cadre d'INEX[6], la compétition internationale de RI XML, que nous présentons dans la section suivante. Les résultats obtenus par notre modèle lors de l'édition 2009 d'INEX ont montré, dans un cadre expérimental rigoureux, l'avantage de prendre en compte les balises XML pour la RI. Ces résultats sont présentés dans la section 5.1. Nous avons ensuite approfondi l'expérimentation du comportement de notre modèle et de l'impact de certains paramètres, à la fois 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. Les résultats de ces expérimentations sont présentés dans la section 5.2.

4.1. *Collection INEX - Wikipédia*

Nous avons utilisé le corpus XML anglophone INEX - Wikipédia (Denoyer *et al.*, 2006), développé dans le cadre d'INEX. Ce corpus est composé de 659 388 articles

5. TTF : Tagged Term Frequency.
6. INEX : Initiative for the Evaluation of XML Retrieval

extraits de l'encyclopédie en ligne Wikipédia[7] en 2006, 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, p, title, list, item, etc.), des balises de mise en forme (bold, emphatic, italic, small, etc.) et des balises représentant des liens (collectionlink, etc.). Il n'existe pas de DTD définissant la liste des balises autorisées, ce qui entraîne une certaine profusion des balises : il en existe 1 257 différentes dans la collection, dont beaucoup apparaissant dans seulement quelques articles. Les articles sont fortement structurés (il y a au total 52 millions d'éléments XML), ce qui permet d'évaluer les systèmes de RI ciblée. 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 seulement 1,6 Go. L'information structurelle XML (balises + attributs) représente donc le double de l'information textuelle.

4.2. *Mesures d'évaluation INEX*

L'évaluation d'INEX 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). Cette mesure 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* (Kamps *et al.*, 2007)).

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). Cela est problématique dans le cas de la RI ciblée, car les réponses ciblées, d'une granularité infra-article, pénaliseront les résultats d'un système, alors même que l'objectif de la RI ciblée est de retourner des éléments de la granularité la plus ciblée possible. 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).

Notons que le classement principal d'INEX est basé sur la mesure $iP[0.01]$ et non la mesure globale $MAiP$, afin de prendre en compte l'importance de la précision aux taux de rappel faibles. Cela entraîne une évaluation privilégiant la précision plutôt que le rappel.

Tous les résultats présentés ici, incluant ceux des systèmes d'INEX, ont été obtenus à l'aide de l'outil d'évaluation fourni par INEX en 2008 : *inex_eval*, version 1.0.

7. Wikipédia : http ://wikipedia.org

4.3. *Protocole expérimental*

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 70 nouvelles requêtes de l'édition 2008 d'INEX. Seuls les mots-clés des requêtes ont été utilisés (champ *title*). Nous n'avons pas utilisé les champs *description*, *narrative* et *castitle* (partie structurée de la requête).

Nous avons expérimenté notre modèle (CLAW et TTF) 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. Ces expérimentations, présentées dans la section 5.1, nous ont permis de montrer l'intérêt de la prise en compte des balises en RI ciblée lors de notre participation à INEX 2008 (Géry *et al.*, 2009).

Ensuite, l'objectif de nos expérimentations a été d'étudier de manière relativement exhaustive l'impact de certains paramètres sur la RI ciblée, et de vérifier le comportement de notre modèle dans le cadre d'un système dont les paramètres ont été soigneusement réglés. Est-ce que notre modèle nous permet d'améliorer les résultats d'une fonction BM25 optimisée ? Qu'en est-il de l'indexation des articles complets par rapport à l'indexation d'éléments XML de granularité fine ? Nous avons donc mené plusieurs expérimentations : articles ; articles + CLAW ; articles + TTF : éléments + CLAW ; éléments + TTF.

Certains paramètres ont été réglés à la suite d'expérimentations préliminaires (cf. section 4.4), au cours desquelles deux paramètres essentiels ont été étudiés afin de mieux comprendre les spécificités de la RI ciblée. Il s'agit des paramètres b et k_1 de la fonction BM25.

Il existe un risque de sur-apprentissage, car l'optimisation des paramètres a été réalisée avec la collection 2008, et cette même collection a été aussi utilisée pour l'évaluation. C'est un problème classique des compétitions de RI (TREC, INEX, etc.), cf. (Robertson *et al.*, 2004; Taylor *et al.*, 2006). Nous pensons, comme (Robertson *et al.*, 2004), qu'il est tout de même pertinent d'évaluer notre modèle dans ce contexte. En effet, nous souhaitons évaluer le potentiel de notre modèle en optimisant les paramètres b et k_1, tout en gardant à l'esprit que, dans des conditions réelles, il sera nécessaire de régler ces paramètres à l'aide d'une collection d'apprentissage.

4.4. *Paramétrage du système*

Selon les expérimentations (notamment RI classique / RI ciblée), différents paramètres doivent être réglés. Certains d'entre eux ont été fixés à l'aide d'expérimentations préliminaires, d'autres, plus importants, ont été étudiés exhaustivement :

– Paramètres fixés : choix de la fonction de pondération (BM25), taille minimum des éléments retournés, taille minimum des termes, profondeur maximum dans l'arbre

XML des éléments retournés, anti-dictionnaire, mode "*andish*", termes obligatoires ou interdits (opérateurs +/-), liste des balises de mise en forme à considérer.

– Paramètres basés sur la granularité[8] : liste des balises de structure logique[9], calcul du df.

– Paramètres étudiés : impact des balises (sans impact, CLAW ou TTF), BM25 b, BM25 k_1.

Toutes les expérimentations présentées dans cet article partagent donc les mêmes réglages concernant l'anti-dictionnaire[10], et la manière de traiter les requêtes : en utilisant le même mode "*andish*" (privilégiant les documents contenant la totalité des mots-clés de la requête) et en considérant non strictement les opérateurs + (termes obligatoire) et - (termes interdits).

Certains paramètres spécifiques ont été choisis lors d'expérimentations préliminaires. Par exemple, dans le cas de la RI ciblée, nous devons définir la taille minimum des éléments retournés par le système. La conversion des articles de Wikipédia en XML ayant été entièrement réalisée de manière automatique, il existe dans la collection des éléments XML très petits (voire de taille nulle), qui, pris isolément, ne sont pas porteurs de sens et donc ne doivent pas être retournés à l'utilisateur. C'est par exemple le cas avec les éléments *languagelink*. De plus, une analyse des jugements de pertinence effectués lors des campagnes 2006 et 2007 d'INEX (non présentée ici) a montré qu'il n'est pas utile de considérer des éléments d'une taille inférieure à 10 termes, car soit ils sont non pertinents, soit leur père dans l'arbre XML est lui même 100% pertinent et dans ce cas il est préférable de le retourner directement. Notons que (Kamps *et al.*, 2005a) a montré, en analysant une collection précédente d'INEX 2002, qu'une valeur optimale de ce paramètre se trouve autour de 40.

Un autre paramétrage du système qui doit être considéré, spécifiquement dans le cas des éléments, est le calcul du df : doit-on calculer le df au niveau des éléments ou doit-on calculer un df global pour la collection (c'est-à-dire au niveau des articles) ? Et, dans le premier cas, faut-il considérer la taille des éléments ? Nous avons choisi de calculer le df aux deux niveaux, c'est-à-dire que la valeur df de discriminance d'un terme peut être différente selon que l'on indexe des articles ou des éléments.

(Taylor *et al.*, 2006) montre qu'une collection d'apprentissage basée sur 100 requêtes est suffisante pour estimer les 9 paramètres de la fonction `BM25F` qui a été expérimentée avec succès lors de TREC 2004 (cf. (Zaragoza *et al.*, 2004)). Notons que Taylor calcule un df global au cours de ces expérimentations.

---

8. Paramètres fixés à des valeurs différents selon le cadre (RI ciblé versus RI classique).
9. Ensemble des balises de structure logique : les types d'éléments que le système sera capable de retourner ; en RI classique cette liste est réduite à une seule balise : *article*.
10. Anti-dictionnaire : liste de 319 mots sélectionnés par l'équipe Glasgow Information Retrieval Group, cf. http ://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words.

4.5. *Sélection des balises*

Un autre paramètre important est la liste des balises de structure logique, c'est-à-dire les types d'éléments XML que le système considère lors des phases d'indexation et d'interrogation. Le système pourra retourner uniquement des éléments dont le type appartient à cette liste. Des expérimentations préliminaires nous ont permis de sélectionner 16 balises de structure logique (cf. table 1).

Tableau 1. *Balises de structure logique sélectionnées*

article	li	row	template
cadre	normallist	section	title
indentation1	numberlist	table	th
item	p	td	tr

Ensuite, les 61 balises ayant un nombre d'occurrence supérieur à 300 ont été sélectionnées parmi la totalité des 1 257 balises apparaissant dans les 659 388 documents (cf. table 2). Enfin, 6 balises ont été supprimées de cette liste manuellement : *article* et *body* (qui contiennent la totalité d'un article), *br*, *hr*, *s* et *value* (qui sont des balises sans contenu).

Tableau 2. *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

4.6. *Pondération des balises*

Un score est ensuite calculé pour chacune des 55 balises restantes (incluant 14 des 16 balises de structure logique), suivant l'équation 8, estimant la capacité des balises à mettre en évidence des termes pertinents. Le tableau 3 présente les six balises ayant obtenu les poids les plus élevés et les six balises ayant obtenu les poids les plus faibles. Leur nombre d'occurrence dans la collection est aussi donné.

Tableau 3. *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

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. Notons la présence des balises *section* et *ul* dans les 6 premières balises, ainsi que la présence des balises *emph4* et *big* dans les 6 dernières.

5. Résultats

5.1. *Résultats INEX*

Nous présentons maintenant les résultats obtenus par notre modèle lors de la compétition INEX 2008. Suivant la procédure d'INEX, nous avons soumis trois expérimentations (*Foc-1*, *Foc-2*, *Foc-3*) 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, c'est-à-dire d'intersection vide. 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 4 présente les résultats de nos trois expérimentations, comparés à ceux du vainqueur de la compétition, l'université de Waterloo (FOERStep). 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). Afin de prendre en compte la contrainte de non recouvrement de la tâche "focused", la liste des éléments renvoyés par notre système est filtrée en supprimant tout les éléments ayant un autre élément d'intersection non vide mieux classé. En gras : le score $iP[0.01]$ du vainqueur, et nos meilleurs scores ($iP[0.01]$, $MAiP$, $R[1500]$ et $S[1500]$).

Notre première expérimentation, *Foc-1* (RI classique), se classe $13^{\text{ème}}$ sur 61. La seconde expérimentation, *Foc-2* (RI ciblée), obtient un moins bon résultat : $37^{\text{ème}}$ sur

Tableau 4. *Évaluation de 61 expérimentations de la tâche "focused"*

	Granularité	Balises	$iP[0.01]$	Rang	$MAiP$	Rang	$R[1500]$	$S[1500]$
FOERStep	Éléments	-	**0,6897**	1	0,2071	27	0,4494	1,11
Foc-1	Articles	-	0,6412	13	**0,2791**	6	**0,7897**	5,57
Foc-2	Éléments	-	0,5688	37	0,1206	45	0,2775	**0,73**
Foc-3	Éléments	TTF	**0,6640**	7	0,2342	19	0,6110	3,34

61. L'impact précoce du poids des balises (stratégie TTF), donne de très bons résultats en RI ciblée : *Foc-3* se classe 7ème sur 61. Ce résultat est bien meilleur que celui obtenu en RI classique (*Foc-1*), et de plus *Foc-3* améliore significativement la RI ciblée à des taux de rappel faibles (de 0,5688 à 0,6640 selon le critère $iP[0.01]$).

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 sans adaptation. 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.

Ces résultats montrent l'avantage de la RI ciblée (*Foc-3*) comparée à la RI classique (*Foc-1*) ; cela montre aussi l'avantage de prendre en compte l'information structurelle (*Foc-2* vs. *Foc-3*) ; et finalement, cela donne de bien meilleurs résultats par rapport à la stratégie d'impact tardif du poids des balises (cf. (Géry *et al.*, 2008)).

Toutefois, notre expérimentation *Foc-1* donne les meilleurs résultats à des taux de rappel supérieurs à 0,05, et *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* tandis que la taille en Mo de ces 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.

Alors que la collection que nous avons utilisée présente des caractéristiques différentes (en particulier, un plus grand nombre et une plus grande diversité des balises XML considérées), nous parvenons aux mêmes conclusions que (Robertson *et al.*, 2004) : il est intéressant de prendre en compte les balises dans la fonction de pondération BM25, à condition que l'impact soit précoce (stratégie TTF, *Foc-3*), ce qui permet de conserver la non-linéarité de BM25, et non pas tardif (directement sur

les poids finaux des termes : stratégie CLAW, cf. (Géry *et al.*, 2008)). Par ailleurs, contrairement aux résultats de (Trotman, 2005), la prise en compte du poids des balises permet une amélioration significative de la fonction de pondération BM25.

5.2. *Analyse a posteriori*

Certains paramètres étant fixés (cf. section 4.4), nous avons expérimenté nos deux stratégies d'impact des balises (CLAW et TTF) en RI classique (articles) et en RI ciblée (éléments), en utilisant une grille 2D pour faire varier le paramètre b (entre 0 et 1, par pas de $0,1$) et le paramètre k_1 (entre $0,2$ et $3,8$ par pas de $0,2$), ce qui représente un total de 1 254 expérimentations.

Les résultats obtenus dans la configuration optimale des paramètres sont présentés dans la table 5 selon le critère $iP[0.01]$, et dans la table 6 selon le critère $MAiP$.

Tableau 5. *Évaluation de 1 254 expérimentations selon le critère* $iP[0.01]$

	Granularité	Balises	b	k_1	$MAiP$	#doc	#art	$R[1500]$	$S[1500]$
R1	Articles	-	0,4	1,6	0,6587	1 457	1 457	**0,8422**	8,22
R2	Articles	CLAW	1,0	3,8	0,6278	1 457	1 457	0,7424	4,26
R3	Articles	TTF	0,6	1,6	0,6654	1 457	1 457	0,8214	7,69
R4	Éléments	-	0,5	0,8	0,6738	1 463	1 257	0,4134	**1,65**
R5	Éléments	CLAW	0,2	3,0	0,6061	1 461	1 280	0,5730	2,83
R6	Éléments	TTF	0,3	0,8	**0,6837**	1 461	1 294	0,5180	2,98

Tableau 6. *Évaluation de 1 254 expérimentations selon le critère* $MAiP$

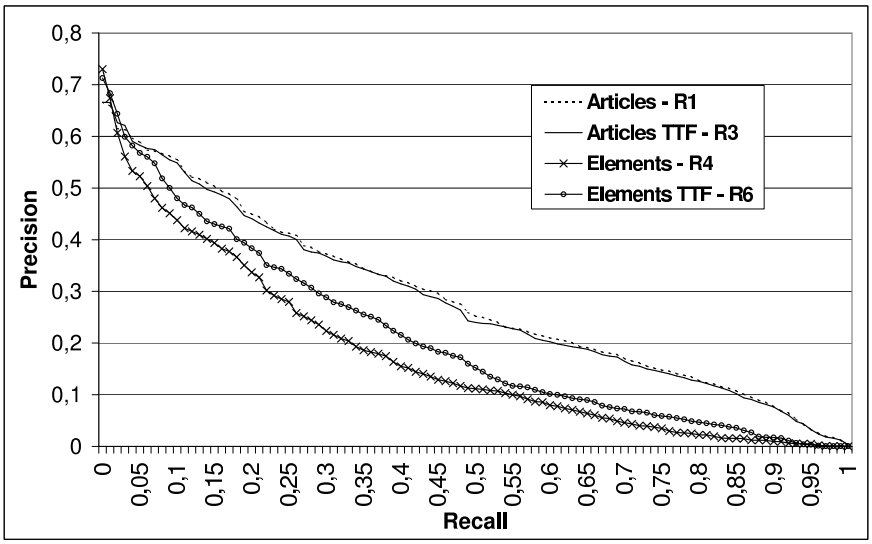
	Granularité	Balises	b	k_1	$MAiP$	#doc	#art	$R[1500]$	$S[1500]$
R7	Articles	-	0,6	2,2	**0,2910**	1 457	1 457	0,8216	6,15
R8	Articles	CLAW	0,8	2,4	0,2522	1 457	1 457	0,8004	6,24
R9	Articles	TTF	0,6	2,6	0,2860	1 457	1 457	**0,8299**	7,09
R10	Éléments	-	0,1	2,2	0,2664	1 459	1 408	0,7476	5,24
R11	Éléments	CLAW	0,1	3,8	0,2137	1 459	1 356	0,6985	**5,00**
R12	Éléments	TTF	0,1	2,8	0,2576	1 459	1 389	0,7285	5,37

Afin de situer ces résultats par rapport à l'évaluation d'INEX, nous pouvons ajouter que l'expérimentation R7 aurait été classée $4^{\text{ème}}$ à INEX 2008 en terme de

$MAiP$ (le vainqueur ayant obtenu $0,3065$), et l'expérimentation R6 aurait été classée également $4^{\text{ème}}$ en terme de $iP[0.01]$ (le vainqueur ayant obtenu 0,6897).

La figure 1 présente les courbes de rappel / précision des 4 expérimentations ayant obtenu les meilleurs résultats selon le critère $iP[0.01]$ (expérimentations $R1, R3, R4, R6$), en excluant les expérimentations CLAW ($R2, R5$) qui donnent de moins bons résultats que TTF ($R3, R6$).

Figure 1. *Courbes de rappel / précision des expérimentations ayant obtenu les meilleurs résultats selon le critère* $iP[0.01]$



5.2.1. *Articles (RI classique : expérimentations 1 et 7) vs éléments (RI ciblée : expérimentations 4 et 10)*

La compétition INEX évalue les systèmes en fonction de leur capacité à retourner des *éléments* pertinents. Néanmoins, il est important d'étudier comment un système classique, basé sur l'indexation des articles seuls, se comporte dans cette tâche. Un tel modèle peut aussi servir de modèle de référence pour étudier les résultats.

La RI classique obtient de meilleurs résultats en terme de rappel : $MAiP$ = 0,2910 pour R7, contre 0,2664 pour R10 (RI ciblée). D'un autre côté, la RI ciblée obtient de meilleurs résultats que la RI classique en terme de précision (à des taux de rappel faibles) : $iP[0.01]$ = 0,6738 pour R4 contre 0,6587 pour R1 (RI classique). Cela confirme les résultats obtenus lors d'INEX 2008 (cf. section 5.1).

Le protocole d'INEX limite le nombre de documents retournés à 1 500. En conséquence, la RI ciblée retournera de plus petites quantités d'information (articles fragmentés) que la RI classique (articles complets), quand la précision est recherchée : R4 retourne 1,65 Mo par requête contre 8,22 Mo pour R1. D'un autre côté, quand l'objectif est d'optimiser le rappel (critère $MAiP$), les différences sont moins importantes : R10 retourne 5,24 Mo par requête contre 6,15 Mo pour R1.

Nous pouvons aussi noter que si le nombre de documents retournés (articles ou éléments) est à peu près identique pour les 4 expérimentations (entre 1 457 et 1 463 sur un maximum théorique de 1 500), cela correspond à un plus petit nombre d'articles dans le cas de la RI ciblée, en raison de la fragmentation des articles en éléments : par exemple 1 257 articles (R4) contre 1 457 (R1). En retournant seulement 1,65 Mo par requête correspondant à 1 463 éléments provenant de 1 257 articles, la fragmentation réalisée par R4 permet d'améliorer la précision (granularité plus fine) mais au détriment du rappel (une plus petite quantité d'information et un plus petit nombre d'articles).

5.2.2. *BM25 (RI textuelle) vs BM25t (TTF : expérimentations 3, 6, 9, 12)*

La table 6 confirme nos résultats à INEX 2008 (cf section 5.1) : la stratégie TTF n'améliore les résultats selon le critère $MAiP$ ni en RI classique (R7 vs R9) ni en RI ciblée (R10 vs R12). D'un autre côté, la table 5 confirme également nos résultats à INEX 2008 : la stratégie TTF améliore les résultats selon le critère $iP[0.01]$ à la fois en RI classique (R1 vs R3) et en RI ciblée (R4 vs R6). L'expérimentation R6 obtient les meilleurs résultats de toutes nos expérimentations en terme de $iP[0.01]$ (0,6837).

5.2.3. *BM25t-TTF (impact précoce) vs BM25t-CLAW (impact tardif : expérimentations 2, 5, 8, 11)*

Finalement, les tables 5 et 6 confirment un autre de nos résultats à INEX 2008 : la stratégie TTF donne de meilleurs résultats que la stratégie CLAW en RI classique et en RI ciblée, aussi bien selon le critère $MAiP$ ($R8$ vs $R9$ et $R11$ vs $R12$) que selon le critère $iP[0.01]$ ($R2$ vs $R3$ et $R5$ vs $R6$). Nous en concluons que l'impact précoce des poids des balises dans la fonction de pondération BM25 (TTF) est une meilleure stratégie que la combinaison de ces poids directement sur les poids finaux des termes calculés avec BM25.

5.3. *Étude des paramètres b et k_1*

Étudions maintenant l'impact des paramètres b et k_1 dans la formule BM25 :

Paramètre b : Le rôle de b est de contrôler la normalisation en fonction de la taille des documents (cf. équation 2). C'est particulièrement important en RI ciblée, car la taille des éléments varie beaucoup plus que celle des articles complets, étant donné que chaque article est fragmenté en éléments (nous avons fixé la

taille minimum des éléments à 10 termes, alors que l'article le plus volumineux compte 35 000 termes).

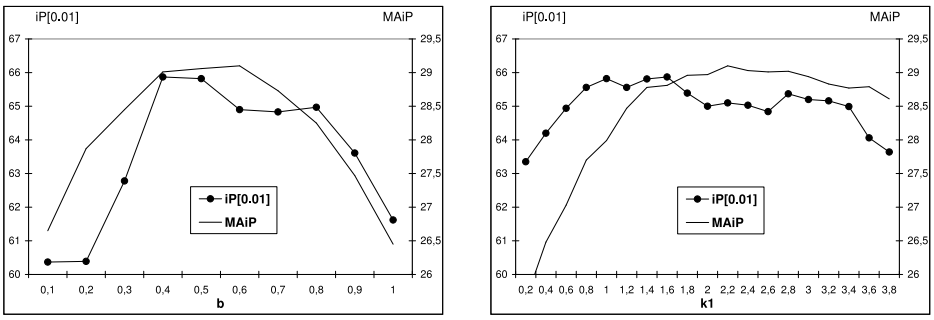
Paramètre k_1 : Le rôle de k_1 est de contrôler le taux de saturation de la fréquence des termes, ce qui est très important pour la stratégie TTF, car TTF modifie directement le tf.

Notons que la stratégie CLAW ne donne pas de bons résultats (ni dans le cadre de la compétition INEX, ni dans l'analyse plus approfondie des résultats), en conséquence de quoi cette stratégie n'est pas présentée dans cette section.

5.3.1. *RI classique*

La figure 2 présente le comportement de la RI classique (articles), avec l'évolution du $MAiP$ et du $iP[0.01]$ en fonction des valeurs de b (resp. k_1). Pour une valeur donnée de b (resp. k_1), les mesures $iP[0.01]$ et $MAiP$ présentées sont celles obtenues avec la valeur optimale de k_1 (resp. b).

Figure 2. *Résultat de la RI classique en fonction de b et k_1*

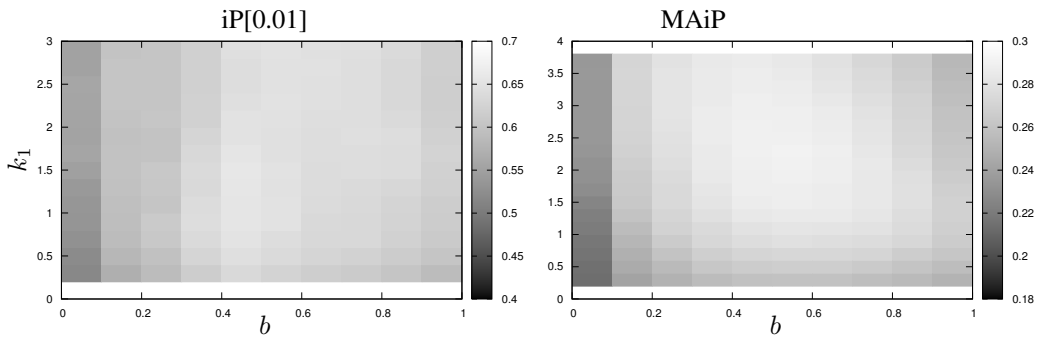


Les meilleures valeurs des paramètres $(b; k_1)$ sont légèrement supérieures pour $MAiP$ ($(b; k_1) = (0,6; 2,2)$) par rapport à $iP[0.01]$ ($(b; k_1) = (0,4; 1,6)$). Ces valeurs sont relativement proches des valeurs souvent rencontrées dans la littérature (*ex.* (0,7 ; 1,2) : pour ces valeurs, notre système obtient un $iP[0.01]$ de 0,6352).

La figure 3 montre une vue 3D des résultats pour $iP[0.01]$ (resp. $MAiP$). L'axe Z (niveaux de gris) montre la mesure $iP[0.01]$ (resp. $MAiP$). Comme on peut le constater, ces résultats sont plutôt réguliers[11], ce qui explique que les quelques expérimentations réalisées pour optimiser les paramètres lors de notre participation à INEX 2008, ont été suffisantes pour obtenir des résultats relativement proches des résultats optimaux ($iP[0.01]$ = 0,6412 *vs* 0,6587).

11. Notons que, comme la mesure $MAiP$ est obtenue par une moyenne des valeurs iP, les résultats en terme de $MAiP$ sont plus réguliers que les résultats en terme de $iP[0.01]$.

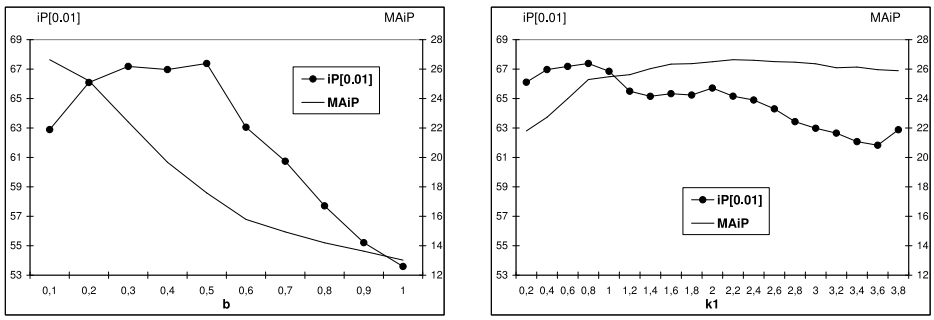
Figure 3. *Résultat de la RI classique en utilisant $iP[0.01]$ et $MAiP$ comme mesures d'évaluation*



5.3.2. *RI ciblée*

La figure 4 présente le comportement du modèle BM25 en RI ciblée.

Figure 4. *RI ciblée en fonction de b et k_1*



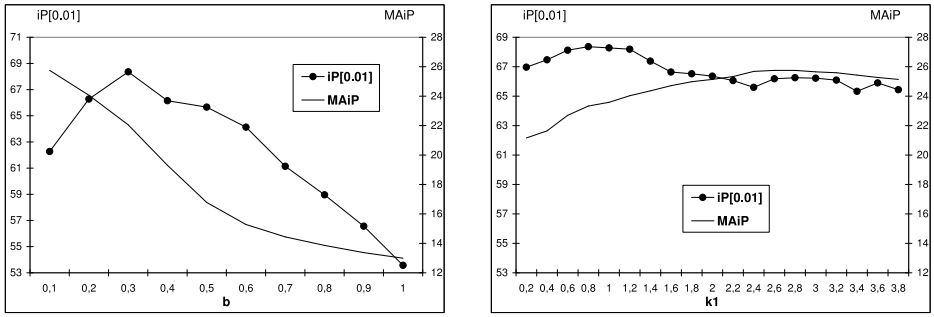
Les meilleures valeurs de $(b; k_1)$ sont différentes pour $MAiP$ ($(b; k_1) = (0,1; 2,2)$) et pour $iP[0.01]$ ($(b; k_1) = (0,5; 0,8)$). Le meilleur résultat en terme de $MAiP$ est obtenu avec une valeur minimum de $b = 0,1$. La normalisation de la taille du document par BM25 semble être contre-productive lorsque l'objectif est d'optimiser le rappel en RI ciblée (MAiP). D'un autre côté, cette normalisation reste efficace lorsque l'objectif est d'optimiser la précision (meilleure valeur pour $iP[0.01]$: $b = 0,5$).

Le paramètre k_1 (saturation du tf) semble être moins important pour la RI ciblée : les mesures de $iP[0.01]$ et de $MAiP$ varient peu en fonction de k_1.

5.3.3. *RI ciblée et BM25t (stratégie TTF)*

La figure 5 présente le comportement du modèle BM25t (stratégie TTF) en RI ciblée.

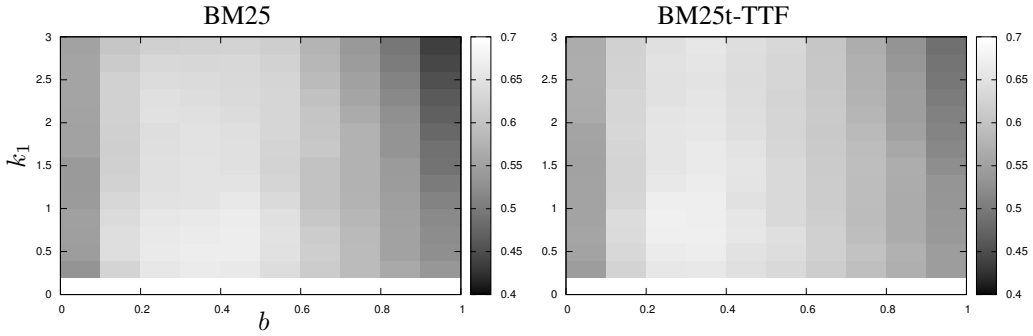
Figure 5. *RI ciblée + TTF en fonction de b et k_1*



Les meilleurs valeurs de $(b; k_1)$ sont, là encore, différentes pour $MAiP$ ($(b; k_1) = (0,1; 2,8)$) et pour $iP[0.01]$ ($(b; k_1) = (0,3; 0,8)$). Comme dans le cas de la RI ciblée sans stratégie TTF, le meilleur $MAiP$ est obtenu avec la valeur minimum de $b = 0,1$. Le comportement de la RI ciblée est similaire avec ou sans stratégie TTF.

La figure 6 montre une vue 3D des résultats selon le critère $iP[0.01]$. Comme auparavant, on constate que le système est assez stable. Nous pensons donc qu'avec notre modèle il est possible d'obtenir, en quelques expérimentations, un réglage des paramètres relativement proche du réglage optimal. Par ailleurs, nous pensons que ce modèle est robuste du point de vue de la généralisation.

Figure 6. *RI ciblée selon le critère $iP[0.01]$*



6. 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 forme 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 ce travail consiste en une modélisation de la capacité des balises à mettre en évidence 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 (stratégie CLAW, cf. (Géry *et al.*, 2008)), nous avons proposé dans cet article une intégration précoce (stratégie TTF)), qui permet de conserver la non-linéarité de la fonction BM25 et donne de bien meilleurs résultats.

La seconde contribution de ce travail est une expérimentation des modèles BM25 dans le contexte de la RI XML. Nous avons tout d'abord 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 (stratégie CLAW, cf. (Géry *et al.*, 2008)).

Alors que la collection que nous avons utilisée présente des caractéristiques différentes (en particulier, un plus grand nombre et une plus grande diversité des balises XML considérées), nous parvenons aux mêmes conclusions que (Robertson *et al.*, 2004) : il est intéressant de prendre en compte les balises dans la fonction de pondération BM25, à condition que l'impact soit précoce (stratégie TTF, *Foc-3*) et non pas tardif (directement sur les poids finaux des termes : stratégie CLAW, cf. (Géry *et al.*, 2008)) ; ce qui permet de conserver la non-linéarité de BM25. Par ailleurs, contrairement aux résultats de (Trotman, 2005), la prise en compte du poids

des balises permet une amélioration significative de la fonction de pondération BM25.

Dans le contexte de la RI XML, le nombre de balises est beaucoup trop important pour pouvoir optimiser les paramètres b et k_1 pour chaque balise, comme expérimenté par (Robertson *et al.*, 2004) avec BM25f pour un très petit nombre de champs. Toutefois, nous pensons que le poids des balises utilisés par notre stratégie TTF peuvent remplacer le réglage fin de BM25f (et très lourd à mettre en œuvre) de b et k_1 pour chaque balise. En effet, les poids des balises impactent le tf_{ji} comme le fait le paramètre k_1.

La dernière contribution de ce travail est une étude relativement exhaustive de l'impact des paramètres b et k_1 de BM25, aussi bien selon la mesure $iP[0.01]$ que selon la mesure $MAiP$. Le premier résultat de cette étude est la relative stabilité de la qualité du système quand les paramètres sont modifiés. C'est important car cela facilite le réglage des paramètres en un nombre raisonnable d'expérimentations préliminaires. De plus, nous pouvons espérer un bon comportement du modèle du point de vue de la généralisation. Cela explique les bons résultats de notre modèle lors d'INEX 2008 : le paramétrage de notre système à l'aide d'une autre collection, a permis d'atteindre un réglage tout à fait correct des paramètres pour l'évaluation sur un nouveau jeu de requêtes, et la hiérarchie de nos stratégies observée durant la compétition est la même que celle observée dans notre étude plus exhaustive. Quand l'objectif est d'optimiser la mesure $MAiP$, le meilleur modèle est le classique modèle BM25. Cela peut probablement s'expliquer par le fait que les systèmes qui retournent des documents de grande taille (granularité des articles) sont mécaniquement favorisés quand les mesures basées sur le rappel sont utilisées. Par contre, à des taux de rappel faibles, quand l'objectif est d'optimiser la précision, les meilleurs résultats sont obtenus par le modèle BM25t qui prend en compte les balises et retourne des éléments XML.

Des perspectives s'offrent à nous à plusieurs niveaux.

Tout d'abord, la stratégie TTF met en œuvre 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 fonctions 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.

D'un point de vue expérimental, nous avons ciblé notre étude sur les paramètres b et k_1 de BM25. D'autres paramètres méritent d'être également étudiés. En particulier, la manière dont le df est calculé peut avoir un impact important. Nous devons aussi travailler à une meilleure prise en compte la grande variation de la taille des documents quand une RI ciblée (éléments) est effectuée. Cela pourrait se faire à l'aide de normalisation des pondérations (cf. (Kamps *et al.*, 2005a)), ou par un calcul plus sophistiqué du df.

7. Bibliographie

Baeza-Yates R. A., Fuhr N., Maarek Y. S., « Introduction to the special issue on XML retrieval », *ACM Transactions on Informations Systems*, vol. 24, n° 4, p. 405-406, 2006.

Baeza-Yates R., Fuhr N., Sacks-Davis R., Wilkinson R. (eds), *Proceedings of the SIGIR 2000 Workshop on XML and Information Retrieval*, Athens, Greece, july, 2000.

Boyan J., Freitag D., Joachims T., « A Machine Learning Architecture for Optimizing Web Search Engines », *AAAI Workshop on Internet-Based Info. Systems*, 1996.

Denoyer L., Gallinari P., « The Wikipedia XML corpus », *SIGIR forum*, vol. 40, p. 64-69, 2006.

Fuhr N., Großjohann K., « XIRQL : An extension of XQL for information retrieval », *In ACM SIGIR, Workshop On XML and Information Retrieval*, Athens, Greece, july, 2000.

Fuhr N., Großjohann K., « XIRQL : A Query Language for Information Retrieval in XML Documents », *24th Conference on Research and development in Information Retrieval, SIGIR'01*, p. 172-180, 2001.

Fuhr N., Kamps J., Lalmas M., Trotman A. (eds), *Focused Access to XML Documents, 6th Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007*, vol. 4862 of *LNCS*, Springer, 2008.

Fuller M., Mackie E., Sacks-Davis R., Wilkinson R., « Coherent Answers for a Large Structured Document Collection », *SIGIR*, p. 204-213, 1993.

Geva S., Kamps J., Trotman A. (eds), *INEX 2008 Workshop Preproceedings, Dagstuhl Castle, Germany, December 15-18*, 2008.

Géry M., Largeron C., Thollard F., « Integrating structure in the probabilistic model for Information Retrieval », *Web Intelligence*, p. 763-769, 2008.

Géry M., Largeron C., Thollard F., « UJM at INEX 2008 : pre-impacting of tags weights », *Proc. of INitiative for the Evaluation of XML Retrieval (INEX), Dagstuhl*, 2009.

Kamps J., de Rijke M., Sigurbjörnsson B., « The Importance of Length Normalization for XML Retrieval », *Inf. Retr.*, vol. 8, n° 4, p. 631-654, 2005a.

Kamps J., Marx M., de Rijke M., Sigurbjörnsson B., « Structured queries in XML retrieval », *CIKM*, p. 4-11, 2005b.

Kamps J., Pehcevski J., Kazai G., Lalmas M., Robertson S., « INEX 2007 Evaluation Measures », *Focused access to XML documents, INEX Workshop*, 2007.

Kazai G., Trotman A., « Users' perspectives on the Usefulness of Structure for XML Information Retrieval », *1st Conference on the Theory of Information Retrieval*, p. 247-260, 2007.

Kim Y.-H., Kim S., Eom J.-H., Zhang B.-T., « SCAI Experiments on TREC-9 », *Text Retrieval Conference (TREC-9)*, p. 392-399, 2000.

Konopnicki D., Schmueli O., « W3QS : A Query System for the World-Wide Web », *21th Conference on Very Large Data Bases (VLDB'95)*, p. 54-65, Sept., 1995.

Kotsakis E., « Structured Information Retrieval in XML documents », *Symposium on Applied Computing*, p. 663-667, 2002.

Lalmas M., « Structure Weight », *Encyclopedia of Database Systems, O.M. Tamer and L. Ling (Eds), Springer*, 2009a.

Lalmas M., « XML Information Retrieval », *Encyclopedia of Library and Information Sciences, J. Bates and M.N. Maack (Eds)*, 2009b.

Lu W., Robertson S. E., MacFarlane A., « Field-Weighted XML Retrieval Based on BM25 », *INEX*, p. 161-171, 2005.

Maron M., Kuhns J., « On Relevance, Probabilistic Indexing and Information Retrieval », *Journal of the ACM*, vol. 7, n° 3, p. 216-244, 1960.

O'Keefe R. A., Trotman A., « The Simplest Query Language That Could Possibly Work », *2nd INEX Workshop, Dagstuhl, Germany, December 15-17*, p. 167-174, 2003.

Rapela J., « Automatically combining ranking heuristics for HTML documents », *Workshop on Web Information and Data Management (WIDM), CIKM*, p. 61-67, 2001.

Robertson S., Jones K. S., « Relevance weighting of search terms », *JASIST*, vol. 27, n° 3, p. 129-146, 1976.

Robertson S., Zaragoza H., Taylor M., « Simple BM25 extension to multiple weighted fields », *CIKM*, New York, USA, p. 42-49, 2004.

Salton G., McGill M., *Introduction to modern Information Retrieval*, McGraw-Hill, 1983.

Schlieder T., Meuss H., « Querying and ranking XML documents », *JASIST*, vol. 53, n° 6, p. 489-503, 2002.

Taylor M., Zaragoza H., Craswell N., Robertson S., Burges C., « Optimisation methods for ranking functions with multiple parameters », *15th Conference on Information and knowledge management, CIKM'06*, New York, USA, p. 585-593, 2006.

Trotman A., « Choosing document structure weights », *Information Processing and Management*, vol. 41, n° 2, p. 243-264, 2005.

Trotman A., Geva S., Kamps J., « Report on the SIGIR 2007 workshop on focused retrieval », *SIGIR Forum*, vol. 41, n° 2, p. 97-103, 2007.

Trotman A., Sigurbjörnsson B., « Narrowed Extended XPath I (NEXI) », *INEX*, p. 16-40, 2004a.

Trotman A., Sigurbjörnsson B., « NEXI, Now and Next », *INEX*, p. 41-53, 2004b.

van Zwol R., Baas J., van Oostendorp H., Wiering F., « Bricks : The Building Blocks to Tackle Query Formulation in Structured Document Retrieval », *ECIR*, p. 314-325, 2006.

Wilkinson R., « Effective Retrieval of Structured Documents », *SIGIR*, p. 311-317, July, 1994.

Wolff J. E., Florke H., Cremers A. B., « Searching and Browsing Collections of Structural Information », *Advances in Digital Libraries*, p. 141-150, 2000.

Zaragoza H., Craswell N., Taylor M., Saria S., Robertson S., « Microsoft Cambridge at TREC 2004 : Web and Hard track », *13th Text REtrieval Conference*, 2004.