Impact précoce du poids des balises pour la recherche d'information ciblée

Mathias Géry, Christine Largeron, Franck Thollard

Université de Lyon, F-69003, Lyon, France
Université de Saint-Étienne, F-42000, Saint-Étienne, France
CNRS UMR5516, Laboratoire Hubert Curien
{mathias.gery, christine.largeron, franck.thollard}@univ-st-etienne.fr

RÉSUMÉ. *Cet article traite de l'intégration des balises XML dans la fonction de pondération des termes, pour la recherche d'information (RI) XML ciblée. Notre modèle permet de considérer un certain type d'information structurelle : les balises qui représentent la structure logique des documents (titre, section, paragraphe, etc.), ainsi que les balises liées à la mise en forme (gras, italique, centré, etc.). Nous prenons en compte l'influence des balises sous forme d'un poids en estimant la probabilité pour une balise de mettre en évidence les termes pertinents. Ensuite, ces poids sont intégrés à la fonction de pondération des termes. Des expérimentations sur une collection de grande taille dans le cadre de la compétition de RI XML, INEX 2008, ont montré une amélioration de la qualité des résultats en RI ciblée.*

ABSTRACT. *This paper addresses the integration of XML tags in terms weighting function for focused XML Information Retrieval (IR). Our model allows to consider a certain kind of structural information: tags that represent logical structure (title, section, paragraph, etc.) as well as tags related to formatting (bold, italic, center, etc.). We take into account the tags influence by estimating the probability that tags distinguishe relevant terms. Then, these weights are integrated in terms weighting function. Experiments on a large collection during INEX 2008 XML IR evaluation campaign showed improvements on focused retrieval.*

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

KEYWORDS: *Probabilistc IR model, Structured IR, XML, Tags, Weighting*

1. Introduction

La plupart des documents disponibles dans des bases textuelles ou sur Internet sont fortement structurés. C'est le cas par exemple pour les articles scientifiques ou pour les documents écrits à l'aide de langages de balises (HTML, XML). L'information fournie par la structure peut être utilisée pour mettre en exergue certains mots : un mot ne revêt pas la même importance s'il apparaît dans une fonte particulière (gras, italique, etc.). De la même manière, un mot est plus important s'il apparaît dans certaines parties de document (un titre, la légende d'une figure, etc.). Cependant, les modèles de Recherche d'Information (RI) classiques (modèles booléen, vectoriel, probabiliste), dans leur version de base, ne prennent pas en compte cette structure.

Un état de l'art est présenté dans la section suivante. La première contribution de ce papier[1] est la proposition d'un cadre formel, présenté dans les sections 3 et 4, prenant en compte explicitement la structure du document. La seconde contribution consiste en une expérimentation du modèle, présentée dans la section 5, sur une collection d'envergure (la collection INEX[2]).

2. État de l'art

La prise en compte de la structure peut être faite soit à l'étape d'indexation soit à l'étape d'interrogation. Nous distinguons 3 types d'approches :

Modèle de requête et structure : l'intégration de la structure à l'étape de l'interrogation peut se faire en adaptant le langage SQL de façon à autoriser des requêtes portant sur la structure du document (cf. [NAV 95], XIRQ [FUH 01]). En pratique, peu d'utilisateurs sont en fait capables de formuler leurs besoins par des requêtes complexes[3] : les besoins sont en général exprimés par quelques mots-clés.

Modèle de document, structure et poids des termes : la seconde approche explorée revisite les modèles classiques en proposant un schéma de pondération de la structure [FUL 93]. Le poids alors affecté à un mot ne dépend pas seulement de sa fréquence (dans le document et la collection) mais aussi de sa position dans le document, définie par rapport aux balises de structure logique et de structure de mise en forme. Le classement final ne dépend pas uniquement de la présence d'un terme dans un document mais de la présence d'un terme étiqueté de manière appropriée.

L'intégration des balises peut être traitée de manière ad-hoc lorsque seules des balises de structure logique sont utilisées. Dans ce cas, le document peut être scindé en autant d'éléments qu'il y a de parties dans le document initial. Chaque élément est alors traité comme un document. Différentes stratégies sont alors possibles pour combiner les scores des éléments pour former le score de celui qui les contient [WIL 94].

---

1. Ce travail a été soutenu par le projet "Web Intelligence" de la région Rhône-Alpes.
2. INitiative for Evaluation of XML Retrieval. See `http://www.inex.otago.ac.nz`
3. Exemple : "Je cherche un paragraphe qui traite de course à pied, contenu dans un article qui parle du marathon de New-York et qui contient une photo d'un marathonien"

Les balises utilisées pour déterminer la structure peuvent être sélectionnées empiriquement [RAP 01]. Afin d'intégrer la structure logique dans un modèle classique, [ROB 04] duplique les éléments autant de fois que les poids le suggèrent, ce qui permet de conserver la non linéarité de la fonction de pondération `BM25`. Les poids peuvent également être appris automatiquement [BOY 96, KIM 00, TRO 05].

Dans toutes ces approches cependant, les systèmes retournent au final des documents complets. Leurs capacités à mener une RI ciblée en retrouvant des parties de documents n'ont pas été évaluées.

Modèle de document, structure et poids des chemins

Une troisième approche consiste à s'appuyer sur une représentation arborescente de la structure des documents [SCH 02, TRO 05]. Chaque élément XML, correspondant à un nœud de l'arbre, est caractérisé par un chemin allant de la racine de l'arbre jusqu'à ce nœud. La structure est prise en compte au niveau d'un mot en considérant le chemin de l'élément qui le contient. Cette approche a été utilisée en considérant un nombre limité de balises [KOT 02, WIL 94, WOL 00]. Par exemple, dans [KOT 02] les termes qui apparaissent dans le titre `journal/article/title` ont un poids plus important que ceux situés dans le résumé `journal/article/abstract`. Le poids du terme est donc une combinaison de son poids calculé classiquement et du poids associé à sa position dans le document.

Dans [TRO 05], Trotman propose de prendre en compte la structure en assignant un poids estimé par algorithme génétique à chaque nœud XML (c'est-à-dire à la position du mot dans le document). Ce poids est combiné au tf dans différents schémas de pondération. Cette technique ne permet cependant pas d'améliorer les résultats de la fonction de pondération la plus performante (`BM25`).

Dans la plupart de ces approches, très peu de balises sont prises en compte (en général moins de 5), et leur sélection nécessite souvent une intervention manuelle.

Notre approche se caractérise par :

– La prise en compte de balises de structure logique et de mise en forme, comme il en existe dans les documents XML, en levant la limitation liée au nombre de balises prises en compte comme dans [ROB 04].

– Une étape d'apprentissage automatique pour estimer le poids de chaque balise, permettant d'évaluer son impact de manière générale et non relativement aux termes qu'elle étiquette. Les poids pouvant avoir un impact négatif, cette étape peut également être considérée comme une étape de sélection de balise.

– La RI ciblée : notre modèle vise à retourner à l'utilisateur des éléments XML de la granularité la plus adaptée possible, au contraire des approches qui visent à améliorer la recherche de documents complets [TRO 05].

– L'extension de la fonction de pondération `BM25` [ROB 76] via l'intégration du poids des balises.

3. Modélisation de la structure des documents

La structure des documents est intégrée dans notre modèle à deux niveaux. Notons que les balises de structure logique contribuent à ces deux niveaux :

1) La structure logique est utilisée pour déterminer la granularité de l'indexation et donc la granularité des éléments que le système sera susceptible de renvoyer. La pertinence n'est plus estimée au niveau du document complet, mais au niveau de parties de documents, par exemple des éléments XML.

2) La structure logique et la structure de mise en forme sont intégrées au niveau de la fonction de pondération des termes, par une étape d'apprentissage au cours de laquelle un poids est associé à chaque balise. Ce poids est basé sur la probabilité que la balise mette en exergue un terme pertinent ou au contraire un terme non pertinent[4].

À l'étape d'interrogation, la probabilité pour un élément d'être pertinent est estimée en combinant les poids des termes qu'il contient avec les poids des balises qui les étiquettent.

4. Un modèle probabiliste pour la représentation de documents structurés

Soit $\mathcal{D}$ un ensemble de documents structurés. Sans perte de généralité, nous considérerons des documents XML. Chaque élément (article, section, paragraphe, etc.) sera représenté par un ensemble de termes. Dans l'exemple suivant nous disposons de trois documents D_0, D_1 et D_2 :

D_0	D_1	D_2
<article>	<article>	<article>
$t_1 t_2 t_3$	<section>	<section>
<section>	$t_2 t_4$	t_5
$t_1 t_4$	$t_2 t_5$	$t_3 t_4$
$t_2 t_5$	</section>	$t_3 t_5$
</section>	$t_2 t_1$	</section>
</article>	</article>	</article>

Le document D_2 est indexé par cinq éléments : un *article* (balise <article>), une *section* (balise <section>) et trois *paragraphes* (balise).

Nous notons :

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

– $T = \{t_1, ..., t_i, ...t_n\}$, un index de termes construit à partir de E ;

– $B = \{b_1, ..., b_k, ..., b_m\}$, l'ensemble des balises.

4. Ceci rejoint les principes du modèle probabiliste [ROB 76] qui, à partir d'une collection de test dans laquelle la pertinence des documents est disponible, estime la probabilité qu'un terme donné apparaisse dans un document pertinent (resp. non pertinent).

Dans la suite, la représentation d'un élément e_j est notée x_j lorsque seuls les termes sont considérés et m_j lorsque à la fois les termes et les balises sont considérés.

4.1. *Score de pertinence d'un élément XML basé sur les termes*

La pertinence d'un élément relativement à une requête Q est fonction du poids des termes qui apparaissent dans l'élément et dans la requête. On note w_{ji} le poids du terme t_i dans l'élément x_j. On définit X_j un vecteur de variables aléatoires et $x_j = (x_{j0}, ..., x_{ji}, ..., x_{jn})$ une réalisation de ce vecteur X_j, avec $x_{ji} = 1$ (resp. 0) si le terme t_i apparaît (resp. n'apparaît pas) dans l'élément e_j.

Étant données ces notations, f_{term}, la pertinence de x_j basée sur les poids des termes, est donnée par le score :

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

Sous ce produit scalaire général se cachent différentes fonctions comme par exemple `ltn`, `ltc` [SAL 83] ou `BM25` [ROB 76]. Des expérimentations antérieures [GéR 08] avec `ltn` et `ltc` ayant donné des résultats médiocres relativement à ceux obtenus avec `BM25`, nous ne considérerons par la suite que `BM25` :

$$w_{ji} = \frac{tf_{ji} \times (k_1 + 1)}{k_1 \times ((1 - b) + (b * ndl)) + tf_{ji}} \times \log \frac{N - df_i + 0.5}{df_i + 0.5} \tag{2}$$

avec :

- tf_{ji} : la fréquence de t_i dans e_j ;
- N : le nombre d'éléments dans la collection ;
- df_i : le nombre d'éléments qui contiennent le terme t_i ;
- ndl : le ratio entre la taille de e_j et la taille moyenne des éléments ;
- k_1 et b : les paramètres classiques de `BM25`.

Notons que la modification des paramètres k_1 et b permet de faire de `BM25` une fonction non linéaire en la fréquence des termes (voir l'analyse approfondie de Robertson et al. [ROB 04] pour plus de détails).

4.2. *Score de pertinence d'un élément XML basé sur les balises*

De la même manière que dans la section précédente, nous définissons M_j comme un vecteur de variables aléatoires T_{ik} à valeur dans $\{0, 1\}$. Les variables aléatoires M_j et leurs réalisations m_j représentent les éléments structurés :

$M_j = (T_{10}, ..., T_{1k}, ..., T_{1m}, ..., T_{n0}, ..., T_{nk}, .., T_{nm})$

avec :
$T_{ik} = 1$ si le terme t_i apparaît dans cet élément étiqueté par b_k
$T_{ik} = 0$ si le terme t_i n'est pas étiqueté par b_k
$T_{i0} = 1$ si le terme t_i apparaît sans étiquette dans B
$T_{i0} = 0$ si terme t_i n'apparaît pas sans être étiqueté

Nous notons $m_j = (t_{10}, ..., t_{1k}, ..., t_{1m}, ..., t_{n0}, ..., t_{nk}, .., t_{nm})$ une réalisation de la variable aléatoire M_j. Dans notre exemple, nous avons b_1 = *article*, b_2 = *section*, b_3 = *p*, b_4 = *b* et T = $\{t_1,..,t_5\}$. L'élément : e_j = t_1 t_2 t_3 de D_0 peut être représenté par le vecteur :

$$m_j = \{t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{20}, t_{21}, ..., t_{53}, t_{54}\} = \{0, 1, 0, 1, 0, 0, 1, ..., 0, 0\}$$

car le terme t_1 est étiqueté par *article* ($t_{11} = 1$), et *p* ($t_{13} = 1$) mais ni par *section* ($t_{12} = 0$) ni par *b* ($t_{14} = 0$). De plus, $t_{10} = 0$ car le terme n'apparaît pas sans étiquette.

Afin d'intégrer la structure des documents, nous ne considérons pas uniquement les poids des termes w_{ji}, mais aussi le poids des balises. Nous voulons estimer la pertinence d'un élément XML e_j (représenté par le vecteur m_j). On veut donc estimer :

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

Soit $f_1(m_j) = \frac{P(R|m_j)}{P(NR|m_j)}$ une fonction de classement. Plus grande est la valeur de $f_1(m_j)$, plus pertinent est l'élément m_j. Utilisant la formule de Bayes, nous avons :

$$f_1(m_j) = \frac{P(m_j|R) \times P(R)}{P(m_j|NR) \times P(NR)}$$

Le terme $\frac{P(R)}{P(NR)}$ étant constant au regard de la collection pour une requête, il ne modifie pas la fonction de classement. Nous pouvons donc définir la fonction f_2 (proportionnelle à f_1) : $f_2(m_j) = \frac{P(m_j|R)}{P(m_j|NR)}$.

Admettant l'hypothèse d'indépendance nous avons :

$$\begin{aligned} P(M_j = m_j|R) &= \prod_{t_{ik} \in m_j} P(T_{ik} = t_{ik}|R) \\ &= \prod_{t_{ik} \in m_j} P(T_{ik} = 1|R)^{t_{ik}} P(T_{ik} = 0|R)^{1-t_{ik}} \end{aligned} \tag{3}$$

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

Pour simplifier les notations, on note, pour un élément XML donné :

$p_{i0} = P(T_{i0} = 0\|R))$:	la probabilité que t_i n'apparaisse pas sans étiquette étant donné un élément pertinent ;
$p_{ik} = P(T_{ik} = 1\|R)$:	la probabilité que t_i apparaisse étiqueté par la balise k, étant donné un élément pertinent ;
$q_{i0} = P(T_{i0} = 0\|NR)$:	la probabilité que t_i n'apparaisse pas sans étiquette étant donné un élément non pertinent ;
$q_{ik} = P(T_{ik} = 1\|NR)$:	la probabilité que t_i apparaisse étiqueté par la balise k, étant donné un élément non pertinent.

Avec ces notations les équations 3 et 4 deviennent :

$$\begin{aligned} P(m_j|R) &= \prod_{t_{ik} \in m_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}, \\ P(m_j|NR) &= \prod_{t_{ik} \in m_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}. \end{aligned}$$

La fonction de classement $f_2(m_j)$ peut alors s'écrire :

$$f_2(m_j) = \frac{\prod_{t_{ik} \in m_j} (p_{ik})^{t_{ik}} \times (1 - p_{ik})^{1-t_{ik}}}{\prod_{t_{ik} \in m_j} (q_{ik})^{t_{ik}} \times (1 - q_{ik})^{1-t_{ik}}}$$

La fonction *log* étant monotone croissante, prendre le logarithme ne changera pas les classements. On a donc la fonction f_3 :

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

Comme précédemment, le terme $\sum_{t_{ik} \in m_j} \log(\frac{1-p_{ik}}{1-q_{ik}})$ est constant relativement à la collection (indépendant de t_{ik}). En ne le considérant pas, on obtient :

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

La fonction de classement obtenue prend en compte les poids des termes (t_i) et des balises (b_k). Le poids d'un terme t_i étiqueté par la balise b_k sera noté w'_{ik} :

$$w'_{ik} = \log(\frac{p_{ik}(1 - q_{ik})}{q_{ik}(1 - p_{ik})}) \tag{6}$$

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

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

Du point de vue pratique, nous devons estimer les probabilités p_{ik} et q_{ik}, $i \in \{1, .., n\}$, $k \in \{0, .., m\}$ pour pouvoir évaluer la pertinence des éléments. À ces fins, nous utilisons un ensemble d'apprentissage LS composé d'éléments pour lesquels la pertinence est connue. Étant donné l'ensemble R (resp. NR) qui contient les éléments pertinents (resp. non pertinents), une table de contingence peut être construite pour chaque terme t_i étiqueté par la balise b_k :

	R	NR	$LS = R \cup NR$
$t_{ik} \in m_j$	r_{ik}	$nr_{ik} = n_{ik} - r_{ik}$	n_{ik}
$t_{ik} \notin m_j$	$R - r_{ik}$	$N - n_{ik} - R + r_{ik}$	$N - n_{ik}$
Total	R	$\|NR\| = N - R$	N

– r_{ik} : le nombre de fois où le terme t_i étiqueté par b_k est pertinent dans LS ;
– $\sum_i r_{ik}$: le nombre de termes pertinents étiquetés par b_k dans LS ;
– n_{ik} : le nombre de fois où le terme t_i est étiqueté par b_k dans LS ;
– nr_{ik} : le nombre de fois où le terme t_i étiqueté par b_k est non pertinent dans LS ;
– $R = \sum_{ik} r_{ik}$: le nombre de termes pertinents dans LS ;
– $|NR| = N - R$: le nombre de termes non pertinents dans LS.

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

Il vient w'_{ik} :

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

Cette fonction de pondération évalue la probabilité, pour une balise donnée, de distinguer les termes pertinents des termes non pertinents : elle augmente avec la capacité de la balise à distinguer un terme pertinent. Notons que l'estimation des probabilités pourrait comporter un lissage dans le cas de collection d'apprentissage de taille limitée ; cela n'a pas été utile dans le cadre de nos expérimentations.

4.3. *Estimation du poids des balises*

D'un point de vue théorique (cf. équation 8), nous pouvons estimer un poids pour chaque paire (terme, balise), c'est-à-dire la capacité pour une balise donnée de renforcer un terme donné. Ce niveau de granularité est à notre avis trop fin. En effet, on cherche à modéliser l'impact d'une balise, non pas relativement à un terme particulier, mais de manière globale. Nous pensons que la capacité d'une balise à mettre en évidence les termes pertinents (ou au contraire à diminuer leur visibilité) est une propriété intrinsèque de la balise et ne dépend donc pas des termes. L'objectif est d'évaluer si un mot apparaissant dans un titre a plus d'importance qu'un mot apparaissant dans une section, et ce indépendamment du mot en question.

Nous nous intéressons donc non plus à un poids par paire (terme-balise), mais à un poids d'une balise indépendamment des termes qu'elle étiquette. Nous obtenons finalement pour chaque balise b_k le poids $w'_k = \frac{\sum_{t_i \in T} w'_{ik}}{|T|}$.

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

À partir des poids des termes et des balises, nous devons calculer un score global des éléments. Afin de prendre en compte toutes les balises qui englobent un terme, nous proposons de combiner la moyenne des poids de ces balises avec le poids du terme lui-même.

Ainsi, notre première fonction de combinaison, f_{claw} (Combining Linearly Average tag-Weights), s'écrit comme suit :

$$f_{claw}(m_j) = \sum_{t_{ik} \in m_j / t_i \in Q} w_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{9}$$

avec w_{ji} le poids du terme t_i dans le document m_j, calculé à l'aide d'une fonction de pondération classique (`ltn`, `ltc`, ou `BM25`).

Dans [GéR 08], l'intégration du poids des balises permet d'améliorer le rappel, mais de manière peu significative. Or, la fonction de pondération `BM25` est non linéaire (cf. section 4.1). En conséquence, impacter le poids d'une balise sur le poids global w_{ji} est très différent de l'impacter sur le nombre d'occurrences du terme tf_{ji}. En accord avec Robertson et *al.* [ROB 04], nous proposons une prise en compte précoce du poids des balises, en intervenant directement sur tf_{ji}. Ainsi, la non-linéarité de la fonction `BM25` est exploitée. Le poids modifié (tf_{ji} multiplié par la moyenne du poids des balises qui englobent t_i), noté ttf (Tagged Term Frequency), remplace le tf dans la fonction de pondération `ltn`, `ltc`, ou `BM25`.

$$ttf_{ji} = tf_{ji} \times \frac{\sum_{k/t_{ik}=1} w'_k}{|\{k/t_{ik} = 1\}|} \tag{10}$$

5. Expérimentations

L'objectif de nos expérimentations est de comparer notre modèle avec un système de référence basé sur un modèle de RI classique, et avec les meilleurs systèmes participant à la compétition internationale de RI XML, INEX 2008. Nous avons expérimenté ces modèles sur une tâche de RI classique où la granularité des réponses est l'article complet, ainsi que sur une tâche de RI ciblée où la granularité des réponses est l'élément XML.

5.1. *Collection INEX - Wikipédia*

Nous avons utilisé le corpus XML anglophone INEX - Wikipédia [DEN 06], développé dans le cadre d'INEX. Ce corpus est composé de 659'388 articles extraits de l'encyclopédie en ligne Wikipédia[5], et d'un ensemble de requêtes et de jugements de pertinence associés. La syntaxe Wiki originelle a été convertie en XML, en utilisant des balises représentant la structure logique des articles (article, section, paragraph, title, list, item, etc.), des balises de mise en forme (bold, emphatic, italic, small, etc.) et des balises représentant des liens (collectionlink, etc.). Les articles sont fortement structurés : il y a au total 52 millions d'éléments XML. Chaque article peut être représenté comme un arbre contenant en moyenne 79 éléments, et ayant une hauteur moyenne de 6,72. Les articles complets (contenu textuel + structure XML) représentent 4,5 Go alors que le contenu textuel seul représente 1,6 Go. L'information structurelle XML (balises + attributs) représente donc le double de l'information textuelle.

5.2. *Protocole experimental*

Dans la phase d'apprentissage, les articles, les 114 requêtes et les jugements de pertinence de la collection 2006 ont été utilisés pour estimer le poids des balises w'_k. Ensuite, l'expérimentation a consisté à traiter sur la même collection de 659'388 documents les nouvelles requêtes de l'édition 2008 d'INEX.

L'évaluation est basée sur les critères de *précision* et de *rappel*. $iP[x]$ est la précision au point de rappel x. La mesure AiP combine *rappel* et *précision* en une seule mesure en calculant la moyenne de $iP[x]$ à 101 points de rappel (x = 0,00 ; 0,01 ; 0,02 ; ... 0,99 ; 1,00). Elle fournit une évaluation du système pour chaque requête. Enfin, le calcul de la moyenne des AiP sur l'ensemble des requêtes donne la mesure globale de performance $MAiP$ (*interpolated mean average precision* [KAM 07]). Le classement principal d'INEX est basé sur $iP[0.01]$ et non $MAiP$, afin de prendre en compte l'importance de la précision aux taux de rappel faibles.

5. Wikipédia : http ://wikipedia.org

Étant donné que chaque expérimentation est soumise à INEX sous la forme d'une liste ordonnée d'au plus 1'500 éléments XML pour chaque requête, ces mesures favorisent, en terme de rappel, les expérimentations retournant des articles complets (et donc une plus grande quantité d'information). Pour en tenir compte, nous avons aussi calculé $R[1500]$, le taux de rappel à 1'500 documents, et $S[1500]$, la taille des 1'500 éléments retournés (en Mo).

5.3. *Sélection des balises*

Pour décomposer les articles XML en éléments à indexer, 14 balises ont été sélectionnées comme représentant la structure logique des documents XML. Il s'agit des balises : *title*, *table*, *caption*, *article*, *body*, *section*, *numberlist*, *definitionitem*, *normallist*, *th*, *td*, *tr*, *p*, *row*. En conséquence, tous les éléments retournés par notre système correspondront à l'une de ces 14 balises.

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

Tableau 1. *Nombre d'occurrences des balises (top 20)*

Balise	#occs	Balise	#occs
collectionlink	16'645'121	normallist	1'087'545
item	5'490'943	row	954'609
unknownlink	3'847'064	outsidelink	84'1443
cell	3'814'626	languagelink	739'391
p	2'689'838	name	659'405
emph2	2'573'195	body	659'396
template	2'396'318	article	659'389
section	1'575'519	conversionwarning	659'388
title	1'558'235	br	378'990
emph3	1'484'568	td	359'908

5.4. *Pondération des balises*

Les scores des 55 balises restantes ont été calculés suivant l'équation 8. Le tableau 2 présente les balises ayant obtenu les poids les plus élevés et les poids les plus faibles. Certaines balises ayant un score élevé sont inattendues (ex. : *sub*). Malgré le score très élevé de la balise *h4*, son impact sera minime sur les estimations de pertinence des éléments XML, car elle n'apparaît que 307 fois dans la collection.

Tableau 2. *Balises ayant les poids w'_k les plus faibles et les plus forts*

Poids les plus élevés (top 6)			Poids les plus faibles (top 6)		
Balise	Poids	#occs	Balise	Poids	#occs
h4	12,32	307	emph4	0,06	940
ul	2,70	3'050	font	0,07	27'117
sub	2,38	54'922	big	0,08	3'213
indentation1	2,04	135'420	em	0,11	608
section	2,01	1'610'183	b	0,13	11'297
blockquote	1,98	4'830	tt	0,14	6'841

6. Résultats

Nous présentons maintenant les résultats obtenus par notre modèle lors de la compétition INEX 2008. Suivant la procédure d'INEX, nous avons soumis 3 expérimentations à la tâche "focused" de la piste Ad-hoc. Cette tâche impose aux systèmes de retourner à l'utilisateur une liste d'éléments XML (ou de passages de texte) non recouvrants.

Notre objectif était tout d'abord d'obtenir une expérimentation de référence performante, puis d'évaluer notre modèle en RI classique et en RI ciblée, et enfin d'analyser l'impact de la prise en compte du poids des balises dans la fonction `BM25`. La table 3 présente les 3 expérimentations. La structure n'est prise en compte ni dans *Foc-1*, où les articles complets sont retournés (granularité : articles), ni dans *Foc-2*, où ce sont les éléments qui sont renvoyés (granularité : éléments), alors que dans *Foc-3*, le poids des balises est intégré dans `BM25` dans une recherche d'information ciblée (granularité : éléments, TTF).

Tableau 3. *Expérimentations soumises à la tâche "focused"*

			Pondération	
Expérimentations	Tâche	Granularité	des termes	des balises
Foc-1	Focused	articles	BM25	-
Foc-2	Focused	éléments	BM25	-
Foc-3	Focused	éléments	BM25	TTF

6.1. *Paramétrage du système*

Les paramètres de la fonction de pondération `BM25` ont été optimisés afin d'améliorer la RI classique (granularité : articles) et la RI ciblée (granularité : élément XML). Parmi les paramètres étudiés, nous pouvons mentionner l'utilisation d'un anti-dictionnaire, l'optimisation des principaux paramètres de `BM25` (k_1 = 1,1 et b = 0,75), etc. Considérant les requêtes, nous avons implémenté un mode "andish" (privilégiant

les documents contenant la totalité des mots-clés de la requête) et nous avons considéré les mots-clés *or* et *and* dans les requêtes. Certains paramètres spécifiques ont aussi été optimisés pour la RI ciblée (par exemple la taille minimum des éléments retournés). Nos expérimentations sont entièrement automatiques. Seuls les mots-clés de la requête ont été utilisés (champ *title* des requêtes INEX). Nous n'avons pas utilisé les champs *description*, *narrative* ou *castitle* (partie structurée de la requête).

6.2. *Classement INEX :* $iP[0.01]$

Notre système donne des résultats très intéressants comparés aux meilleurs participants à INEX (cf. tableau 4, et critères définis en section 5.2). Nos expérimentations sont comparées sur la figure 1 à *FOERStep* (Université de Waterloo), l'équipe qui a remporté la tâche "focused". *FOERStep* donne de meilleurs résultats à des taux de rappel faible. Par contre, notre expérimentation *Foc-1* donne les meilleurs résultats à des taux de rappel supérieurs à 0,05, et également en considérant le critère $MAiP$.

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

Expérimentation	$iP[0.01]$	Rang	$MAiP$	Rang	R[1500]	S[1500]
FOERStep	**0.6873**	1	0.2071	27	0.4494	78
Foc-1	0.6412	13	**0.2791**	6	**0.7897**	390
Foc-2	0.5688	37	0.1206	45	0.2775	**51**
Foc-3	**0.6640**	7	0.2342	19	0.6110	234

6.3. *Articles versus éléments*

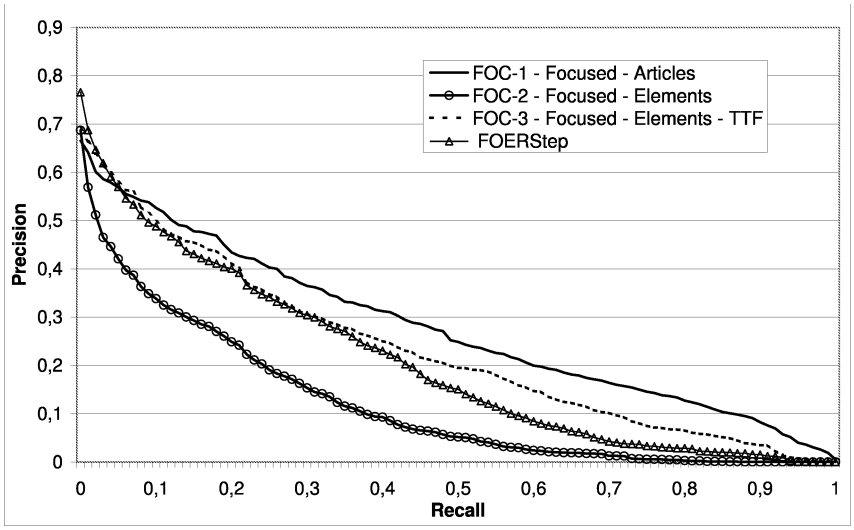
Afin de comparer la RI classique et la RI ciblée, nous avons indexé les articles complets d'une part (*Foc-1*) et les éléments XML de l'autre (*Foc-2*), et nous avons optimisé les paramètres du système dans les deux cas.

La RI ciblée (*Foc-2*), portant sur des éléments XML de taille et de granularité très variables, donne de moins bons résultats que la RI classique (*Foc-1*), bien que le paramètre `ndl` de `BM25` soit justement conçu pour prendre en compte des tailles de documents différentes, et donc des granularités de documents différentes. Les méthodes classiques de RI semblent peu adaptées à la RI ciblée lorsqu'elles sont appliquées telles quelles. D'ailleurs, 3 des 10 meilleures expérimentations sont basées sur des articles complets uniquement. La RI ciblée ne parvient donc pas encore à améliorer significativement les résultats de la RI classique.

6.4. *Impact des poids des balises sur les poids des termes*

Les paramètres utilisés pour *Foc-3* sont inchangés par rapport à ceux de *Foc-2*. La figure 1 montre que notre stratégie TTF (*Foc-3*) améliore significativement la RI

Figure 1. *Rappel / Précision de nos 3 expérimentations et des vainqueurs de la tâche "focused"*



ciblée à des taux de rappel faibles (de 0.5688 à 0.6640 selon le critère $iP[0.01]$). TTF donne également de meilleurs résultats que la RI classique (*Foc-1*). Enfin, le tableau 4 montre que *Foc-1* et *Foc-3* donnent de très bons résultats en terme de rappel : $MAiP$ de 0,2791 (resp. 0,2341) et $R[1500]$ de 0,7897 (resp. 0,6110). Le rappel à 1'500 documents décroît de 16% entre *Foc-1* et *Foc-3* alors que la taille en Mo des 1'500 documents décroît dans le même temps de 40%. Cela montre que le "tamis" de la RI ciblée élimine plus d'éléments non pertinents que d'éléments pertinents.

Ces résultats confirment aussi qu'il est important de conserver la non linéarité de la fonction de pondération `BM25`, par un impact précoce de l'information structurelle sur la fréquence des termes (stratégie TTF) plutôt que par un impact tardif de cette information directement sur les poids finaux des termes (stratégie CLAW, cf. [GéR 08]).

7. Conclusion et perspectives

Nous avons présenté dans cet article une nouvelle approche de prise en compte de la structure XML pour la RI ciblée, basée sur les principes du modèle probabiliste de RI. Nous considérons à la fois la structure logique et la structure de mise en forme. La structure logique est utilisée lors de la phase d'indexation, afin de définir les types d'éléments XML indexés (et potentiellement retournés) par le système. La structure logique et la structure de mise en page sont intégrées dans le modèle de document : lors d'une phase d'apprentissage, un poids est calculé pour chaque balise, basé sur la probabilité que la balise distingue les termes pertinents des termes non pertinents.

Lors de la phase d'interrogation, le calcul de la pertinence d'un élément XML pour une requête est une combinaison des poids des termes contenus et des poids des balises qui les étiquettent.

La contribution principale de notre modèle consiste en une modélisation de la capacité des balises à mettre en évidences les termes, suivant les principes du modèle probabiliste de RI. De cette manière, le réglage du poids des balises s'effectue de manière entièrement automatique. L'intégration tardive du poids des balises dans la fonction de pondération des termes ayant montré une amélioration peu significative des résultats [GéR 08], nous avons proposé dans cet article une intégration précoce, qui permet de conserver la non-linéarité de la fonction `BM25` et donne de bien meilleurs résultats.

Nous avons évalué notre modèle lors de la compétition internationale de RI XML, INEX 2008. Notre première expérimentation *Foc-1*, en RI classique (granularité des réponses : articles complets), se classe 13ème sur 61. Notre seconde expérimentation *Foc-2*, en RI ciblée (granularité des réponses : éléments XML), obtient un moins bon classement : 37ème sur 61. L'intégration précoce du poids des balises *Foc-3*, en RI ciblée, donne de très bons résultats en obtenant une 7ème place sur 61, montrant ainsi l'intérêt de la RI ciblée (*Foc-3*) comparée à la RI classique (*Foc-1*), montrant également l'intérêt de la prise en compte de l'information structurelle (*Foc-2* vs *Foc-3*) et montrant enfin de bien meilleurs résultats que l'intégration a posteriori du poids des balises [GéR 08].

Nous arrivons aux mêmes conclusions que Robertson et *al.* [ROB 04], bien que les collections utilisées soient très différentes (nombre et diversité des balises considérées) : il est intéressant de prendre en compte les balises dans la fonction de pondération `BM25`, dans la mesure où elles sont prises en compte de manière précoce. Par ailleurs, au contraire de [TRO 05], la prise en compte du poids des balises permet une amélioration significative de la fonction de pondération `BM25`.

Des perspectives s'offrent à nous à plusieurs niveaux. Tout d'abord, la stratégie TTF met en oeuvre une simple moyenne du poids des balises qui étiquettent un terme. De précédentes expérimentations ont montré que cette méthode donnait de meilleurs résultats que d'autres fonction de combinaison (multiplication des poids, prise en compte de la plus proche balise uniquement, etc.). Une analyse tant théorique qu'expérimentale est nécessaire sur ce point. La moyenne arithmétique utilisée met au même plan toutes les balises englobant un terme donné. Une pondération non uniforme des poids des balises, en fonction par exemple de la distance entre le terme et la balise, pourrait se révéler plus performante. Par ailleurs, des résultats positifs en RI ciblée ouvre des perspectives intéressantes en terme de présentation des résultats à l'utilisateur.

8. Bibliographie

[BOY 96] BOYAN J., FREITAG D., JOACHIMS T., « A Machine Learning Architecture for Optimizing Web Search Engines », *AAAI Workshop on Internet-Based Info. Systems*, 1996.

[DEN 06] DENOYER L., GALLINARI P., « The Wikipedia XML corpus », *SIGIR forum*, vol. 40, 2006, p. 64-69.

[FUH 01] FUHR N., GROSSJOHANN K., « XIRQL : A Query Language for Information Retrieval in XML Documents », *SIGIR*, 2001, p. 172-180.

[FUL 93] FULLER M., MACKIE E., SACKS-DAVIS R., WILKINSON R., « Coherent Answers for a Large Structured Document Collection », *SIGIR*, 1993, p. 204-213.

[GéR 08] GÉRY M., LARGERON C., THOLLARD F., « Integrating structure in the probabilistic model for Information Retrieval », *Web Intelligence*, 2008, p. 763-769.

[KAM 07] KAMPS J., PEHCEVSKI J., KAZAI G., LALMAS M., ROBERTSON S., « INEX 2007 Evaluation Measures », *Focused access to XML documents, INEX Workshop*, 2007.

[KIM 00] KIM Y.-H., KIM S., EOM J.-H., ZHANG B.-T., « SCAI Experiments on TREC-9 », *Text Retrieval Conference (TREC-9)*, 2000, p. 392-399.

[KOT 02] KOTSAKIS E., « Structured Information Retrieval in XML documents », *Symposium on Applied Computing*, 2002, p. 663-667.

[NAV 95] NAVARRO G., BAEZA-YATES R. A., « A Language for Queries on Structure and Contents of Textual », *SIGIR*, 1995, p. 93-101.

[RAP 01] RAPELA J., « Automatically combining ranking heuristics for HTML documents », *Workshop on Web Information and Data Management (WIDM), CIKM*, 2001, p. 61-67.

[ROB 76] ROBERTSON S., JONES K. S., « Relevance weighting of search terms », *JASIST*, vol. 27, n° 3, 1976, p. 129–146.

[ROB 04] ROBERTSON S., ZARAGOZA H., TAYLOR M., « Simple BM25 extension to multiple weighted fields », *CIKM*, New York USA, 2004, p. 42–49.

[SAL 83] SALTON G., MCGILL M., *Introduction to modern Information Retrieval*, McGraw-Hill, 1983.

[SCH 02] SCHLIEDER T., MEUSS H., « Querying and ranking XML documents », *JASIST*, vol. 53, n° 6, 2002, p. 489-503.

[TRO 05] TROTMAN A., « Choosing document structure weights », *Information Processing and Management*, vol. 41, n° 2, 2005, p. 243-264.

[WIL 94] WILKINSON R., « Effective Retrieval of Structured Documents », *SIGIR*, July 1994, p. 311–317.

[WOL 00] WOLFF J. E., FLORKE H., CREMERS A. B., « Searching and Browsing Collections of Structural Information », *Advances in Digital Libraries*, 2000, p. 141-150.