

# Propositions pour la pondération des termes et l'évaluation de la pertinence des éléments en recherche d'information structurée

Karen Pinel-Sauvagnat, Mohand Boughanem

IRIT-SIG

118 route de Narbonne,

F- 31062 Toulouse Cedex 4

sauvagna@irit.fr et boughane@irit.fr

## Résumé

*La recherche d'information dans des corpus de documents structurés doit faire face à de nombreuses problématiques. L'une d'elles concerne l'évaluation de la pertinence des éléments : le but est de renvoyer à l'utilisateur une liste triée de résultats. Cette évaluation repose sur la pondération des termes d'indexation utilisée ainsi que sur le modèle suivi pour la mise en correspondance de la requête et des éléments. Dans cet article, nous nous proposons d'explorer diverses pistes pour répondre à ce problème, parmi lesquelles on peut citer l'introduction du contexte des éléments à divers niveaux de granularité. Nos expérimentations utilisent le système XFIRM sur la campagne d'évaluation INEX 2005, et permettent de montrer l'importance de la spécificité des éléments pour les mesures de pertinence utilisées. La pertinence contextuelle semble quant à elle n'avoir que peu d'impact sur la pertinence des éléments, ce qui contredit de précédents résultats. Ceci met en lumière les contradictions obtenues par les différentes mesures de pertinence utilisées dans le cadre de la campagne d'évaluation INEX.*

**Mots-clés :** RI structurée, XML, pondération des termes, pertinence contextuelle

## Abstract

*Structured Information Retrieval copes with a number of open issues. One can cite the evaluation of elements relevance : the aim is to return to the user a ranked list of results. This evaluation is based on the term weighting scheme and on the model used for the matching of queries and elements. In this paper, we propose to explore some clues to answer to this problem. For example, we propose to introduce the*

*element context at different granularity levels. Our experiments use the XFIRM system on the INEX 2005 test suite. Results show the importance of elements specificity for the considered metrics. Contextual relevance seems however to have no impact on elements relevance, which contradicts prior results. It highlights contradictions on the relevance metrics used in the INEX evaluation campaign.*

**Key-words:** *Structured IR, XML, terms weighting schemes, contextual relevance*

## 1 INTRODUCTION

Ces dernières décennies, la Recherche d'Information (RI) s'est principalement intéressée à la recherche de documents pertinents à une requête utilisateur. L'émergence récente du format XML (eXtensible Markup Language) comme format standard pour la représentation des documents, soulève de nouvelles problématiques. Le balisage des documents XML permet de structurer les documents sous forme d'éléments imbriqués les uns dans les autres. Le but de la recherche d'information dans de tels documents structurés (on parle de RI structurée) est alors d'utiliser cette structure afin de renvoyer à l'utilisateur des éléments se focalisant sur son besoin, c'est-à-dire des éléments de granularité appropriée.

La recherche en RI structurée est grandement facilitée depuis 4 ans par la campagne d'évaluation INEX (INitiative for the Evaluation of XML retrieval). Depuis 2002, des tâches de recherche variées et des workshops annuels ont fourni une plateforme de discussion à de nombreuses équipes intéressées par le problème. Les méthodes proposées pour répondre aux problématiques de la RI structurée sont nombreuses et variées. Certaines approches voient le problème sous un angle orienté **base de données**. Les documents XML sont alors considérés comme une suite de données, typées et relativement homogènes. La recherche consiste à représenter de façon complète la structure des documents et à évaluer de manière exacte des expressions du type *attribut=valeur*. D'autres approches cherchent à adapter les techniques utilisées traditionnellement en **RI**, et évaluent la pertinence du contenu (textuel ou structurel) des documents vis-à-vis de la requête. La plupart des équipes proposent une **approche hybride**, cherchant ainsi à tirer parti des deux domaines de compétence.

Les participants à INEX s'entendent cependant sur les principales problématiques liées à la RI structurée. Parmi elles, on peut citer les problématiques liées la pondération des termes d'indexation et au calcul de la pertinence des éléments. C'est sur ces deux points que nous focalisons notre attention dans cet article. Nous nous proposons d'étudier différentes formules de pondération et d'algorithmes de tri, afin de faire ressortir quels sont les éléments fondamentaux que la RI structurée doit prendre en compte. Nos expérimentations sont basées sur le système XFIRM [21], et utilisent le jeu de test des requêtes portant sur le contenu seul (CO : Content-Only)

de la campagne d'évaluation INEX 2005. Nous avons déjà mené un certain nombre d'expérimentations sur ces formules de pondération et de calcul de pertinence dans les campagnes d'évaluation précédentes. Nous nous proposons ici d'étendre notre réflexion avec de nouvelles formules (qui seront comparées avec les précédentes), et de confirmer nos résultats avec les nouvelles mesures de pertinence introduites en 2005.

Le reste de l'article est organisé comme suit. Nous présentons dans la section 2 les différentes approches proposées dans la littérature pour indexer l'information textuelle et évaluer la pertinence des éléments. La section 3 présente le système XFIRM, basé sur une méthode de propagation de la pertinence. Nos propositions pour la pondération et l'évaluation de la pertinence des éléments sont décrites dans la section 4. Les sections 5 et 6 présentent la campagne d'évaluation INEX 2005 et les différents résultats obtenus.

## 2 ETAT DE L'ART

### 2.1 Indexation de l'information textuelle : portée et pondération des termes d'indexation

Le processus d'indexation de la recherche d'information traditionnelle consiste à extraire les termes importants des documents. Cette problématique reste bien entendue d'actualité dans le cadre des documents structurés.

Pour les approches orientées BD, l'unité textuelle d'indexation est le texte complet des nœuds feuilles. Pour les approches orientées RI, il s'agit au contraire du terme, qui sera de plus pondéré afin de refléter son importance. Dans ce qui suit, nous nous intéressons exclusivement aux approches orientées RI.

Avant d'aborder le problème de la pondération des termes d'indexation, quelques remarques s'imposent concernant un problème spécifique aux documents structurés et auquel est liée la problématique de pondération, à savoir la portée des termes d'indexation.

Le problème de la portée des termes d'indexation est le suivant : Comment rattacher les termes à l'information structurelle ? Doit-on chercher à agréger le contenu des nœuds ou au contraire à indexer tous les contenus des nœuds séparément ? Ces deux solutions correspondent aux approches d'indexation dites *des sous-arbres imbriqués* et *des unités disjointes* [1].

Les approches du premier groupe considèrent que le texte complet de chaque nœud de l'index est un document atomique [1, 12] et propagent donc les termes des nœuds feuilles dans l'arbre des documents. En d'autres termes, ces approches *indexent tous les sous-arbres* (jugés potentiellement pertinents) des documents. Comme les documents XML possèdent une structure hiérarchique, les nœuds de l'index sont imbriqués les uns dans les autres et l'index contient de nombreuses informations redondantes.

On trouvera une illustration de l'indexation de sous-arbres imbriqués sur la

figure 1.

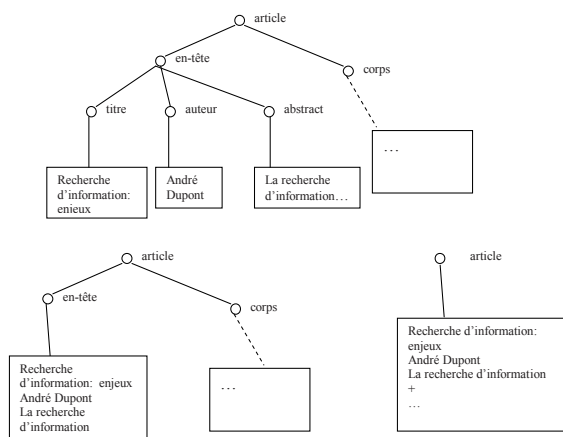


FIG. 1 – Indexation de sous-arbres imbriqués

Les termes "andré dupont" sont par exemple reliés aux nœuds */article/en-tête/auteur*, */article/en-tête*, et */article*.

Dans les approches du second groupe, le document XML est décomposé en unités disjointes, de telle façon que le texte de chaque nœud de l'index est l'union d'une ou plus de ces parties disjointes [18, 7, 20, 2]. Les termes des nœuds feuilles sont uniquement reliés au nœud parent qui les contient.

Si on reprend en exemple l'arbre de la figure 1, les termes "recherche d'information enjeux" seront uniquement reliés au nœud */article/en-tête/titre*, les termes "andré dupont" au nœud */article/en-tête/auteur* et les termes "la recherche d'information" au nœud */article/en-tête/abstract*. Le nœud */article/en-tête* n'est quant à lui relié à aucun terme.

L'approche utilisée pour indexer le contenu des documents semi-structurés implique l'utilisation de méthodes différentes pour la recherche dans les documents. Nous reviendrons sur ces différentes méthodes dans la section 2.2. Considérons maintenant la problématique de la pondération des termes d'indexation. Les approches orientées RI extraient les termes d'indexation selon des processus similaires à ceux utilisés en RI traditionnelle. La pondération de ces termes doit cependant être vue sous un nouvel angle. Alors qu'en RI traditionnelle, le poids d'un terme cherche à rendre compte de son importance de manière locale au sein du document et de manière globale au sein de la collection, s'ajoute en RI structurée l'importance du terme au niveau de l'élément qui le contient.

Les occurrences des termes ne suivent plus forcément une loi de Zipf [30]. Le nombre de répétitions des termes peut être (très) réduit dans les documents XML et l'utilisation d'*idf* (Inverse Document Frequency) n'est pas

forcément appropriée.

L'utilisation d'*ief* (Inverse Element Frequency) a été proposée par de nombreux auteurs [28, 9]. On trouvera des exemples d'adaptation des formules de pondération traditionnellement utilisées en RI à la RI structurée dans [26]. En outre, l'utilisation de la pertinence des nœuds descendants pour calculer le score d'un nœud est souvent proposée, notamment dans les techniques de propagation de la pertinence [7, 10, 20, 2]. La pertinence d'un nœud peut aussi être calculée à part, puis combinée avec la pertinence des nœuds descendants. Dans [6] par exemple, le score de chaque nœud est combiné avec les scores des nœuds fils divisés par le nombre de nœuds fils. Dans [29], le calcul du poids des termes est influencé par le contexte (l'unité d'indexation) dans lequel ils apparaissent. Ce calcul de poids s'inspire de la méthode *tf-idf* qu'on applique aux balises. Ainsi, les auteurs définissent le *tf-idf* (*Term Frequency - Inverse Tag and Document Frequency*), qui permet de calculer la force discriminatoire d'un terme  $t$  pour une balise  $b$  relative à un document  $d$ .

Dans [13], l'importance d'un terme dans un élément est l'agrégation (effectuée à l'aide d'opérateurs OWA) de l'importance du terme dans le contenu du nœud même, dans le contenu de ses descendants, dans le contenu de ses voisins directs et dans le contenu des nœuds auquel il est relié. Le calcul du poids des termes est effectué au moment de l'indexation.

D'autres paramètres permettant d'évaluer l'importance des termes peuvent être pris en compte : la fréquence du terme au sein de l'élément bien sûr, mais aussi la fréquence du terme au sein du document, ou encore la longueur de l'élément et la longueur moyenne des éléments de la collection.

## 2.2 Evaluation de la pertinence des éléments

Dans les approches présentées dans la littérature, les modèles de RI classiques ont été adaptés pour tenir compte de l'information structurelle contenue dans les documents XML et des tailles variées des éléments (c'est-à-dire des granularités variées de l'information).

Dans les approches issues du **modèle vectoriel**, une mesure de similarité de *chaque* élément à la requête est calculée, et ce à l'aide de mesures de distance dans un espace vectoriel. Les éléments sont représentés par des vecteurs de termes pondérés. Pour ce faire, la plupart des approches indexent des sous-arbres imbriqués, c'est-à-dire propagent les termes des nœuds feuilles dans l'arbre du document. Les éléments sont renvoyés à l'utilisateur par ordre décroissant de pertinence.

On trouvera dans [8] et [24] les premières adaptations du modèle. Dans [9], Grabs et Scheck proposent d'évaluer l'importance d'un terme dans un élément donné en fonction de l'importance du terme dans les éléments du même type. Soit  $SE(e)$  l'ensemble des descendants de  $e$  incluant  $e$ .  $\forall se \in SE(e), l \in path(e, se)$  est une étiquette appartenant au chemin reliant  $e$  à  $se$ , c'est-à-dire un type d'élément. Soit enfin  $aw_l \in [0, 1]$  un facteur modélisant

l'importance de l'étiquette  $l$ . La similarité d'un élément  $e$  à une requête  $q$  composée de simples mots-clés est définie de la façon suivante :

$$RSV(e, q) = \sum_{se \in SE(e)} \sum_{t \in terms(q)} tf(t, se) \left( \prod_{l \in path(e, se)} aw_l \right) \cdot ief_{cat(se)}^2(t) \cdot tf(t, q) \quad (1)$$

où  $ief_{cat(se)} = \log \frac{N_{cat(se)}}{ef_{cat(se)}(t)}$ , avec  $N_{cat(se)}$  le nombre d'éléments du type  $cat(se)$ , c'est à dire du même type que  $se$  et  $ef_{cat(se)}(t)$  la fréquence du terme  $t$  dans les éléments de type  $cat(se)$ . Cette approche a été évaluée dans la campagne d'évaluation INEX 2002 et les résultats ont cependant été peu convaincants.

Le modèle JuruXML [17] propose d'indexer les éléments selon leur type (un index par type d'élément) et d'appliquer ensuite le modèle vectoriel pour la pondération des éléments. Les requêtes orientées contenu sont évaluées sur chacun des index et les résultats, qui ont été normalisés, sont ensuite fusionnés afin de fournir à l'utilisateur une liste unique de résultats. Cette dernière approche, évaluée dans le cadre de la campagne INEX 2004 permet d'obtenir de bons résultats par rapport à l'ensemble des participants.

Le moteur de recherche XXL [25] est lui aussi basé sur le modèle vectoriel et utilise une fonction de tri basée sur  $tf$  et  $idf$ . On trouvera d'autres exemples d'adaptation du modèle vectoriel dans [2, 5, 4, 27, 11].

D'autres approches se basent sur le modèle **probabiliste**.

Pour étendre le modèle probabiliste inférentiel aux documents XML, les probabilités doivent tenir compte de l'information structurelle. Une approche est d'utiliser des probabilités conditionnelles de jointure, avec par exemple  $P(d|t)$  devenant  $P(d|p \text{ contains } t)$ , où  $d$  représente un document ou une partie de document,  $t$  est un terme et  $p$  est un chemin dans l'arbre structurel de  $d$ .

Une méthode de *propagation de la pertinence* est proposée par Fuhr et al. dans [7, 10]. Cette méthode est basée sur le langage de requêtes XIRQL, et a été implémentée au sein du moteur de recherche HyRex. Dans cette approche, les nœuds sont considérés comme des unités disjointes. Tous les nœuds feuilles ne sont cependant pas indexés (car d'une granularité trop fine). Dans ce cas-là les termes sont propagés jusqu'au nœud indexable le plus proche. Afin de préserver des unités disjointes, on ne peut associer à un nœud que des termes non reliés à ses nœuds descendants. Le poids de pertinence des nœuds dans le cas de requêtes orientées contenu est calculé grâce à la *propagation* des poids des termes les plus spécifiques dans l'arbre du document. Les poids sont cependant diminués par multiplication par un facteur, nommé facteur "*d'augmentation*". Par exemple, considérons la structure de document suivante, contenant un certain nombre de termes pondérés, et la requête "XML" :

`<section> 0.5 XML`

`<paragraphe> 0.8 définition </paragraphe>`

<paragraphe>0.8 XML 0.9 recherche </paragraphe>  
</section>

Le poids de pertinence de l'élément *section* est calculé comme suit, en utilisant un facteur d'augmentation égal à 0.7 :  $0.5 + 0.7 \times 0.8 - 0.5 \times 0.7 \times 0.8 = 0.78$ .

Le nœud *section* sera donc moins bien classé que le nœud *paragraphe*.

Dans [12], les auteurs proposent une approche basée sur les *modèles de langage* pour traiter les requêtes orientées contenu. Les auteurs considèrent que comme n'importe quel élément XML peut potentiellement être renvoyé à l'utilisateur, chaque élément doit être traité comme une unité d'indexation à part entière. Par conséquent, pour chaque élément, le texte qu'il contient ainsi que le texte contenu dans ses descendants est indexé. Un modèle de langage est ensuite estimé pour chaque élément de la collection. Pour une requête donnée, les éléments sont triés par rapport à la probabilité que le modèle de langage de l'élément génère la requête. On trouvera d'autres approches basées sur les modèles de langages dans [16, 1, 18, 12].

Enfin, dans [19], on trouve un exemple d'utilisation des *réseaux bayésiens* à la recherche d'information structurée.

### 3 LE MODÈLE XFIRM

Le modèle XFIRM [21] est basé sur un modèle de données générique permettant l'implémentation de nombreux modèles de RI et le traitement de collections hétérogènes (c'est-à-dire contenant des documents ne suivant pas la même DTD).

Nous considérons qu'un document structuré  $ds_i$  est un arbre, composé de nœuds simples  $n_{ij}$ , de nœuds feuilles  $n_{ij}^f$  et d'attributs  $a_{ij}$ . Les nœuds feuilles sont porteurs de contenu, alors que les autres nœuds donnent simplement des indications de structure. On trouvera un exemple d'arbre XML sur la figure 2. Le traitement des requêtes portant sur le contenu seul des éléments (requêtes à base de mots-clés, encore appelées requêtes CO (Content-Only)) est effectué comme présenté ci-dessous : une première étape consiste à évaluer la similarité des nœuds feuilles de l'index à la requête (on parle alors de calcul des poids des nœuds feuilles) et une seconde étape consiste à rechercher les sous-arbres pertinents. La pertinence des sous-arbres est évaluée en propageant le poids des feuilles dans l'arbre du document.

#### 3.1 Evaluation du poids des nœuds de l'index

Soit  $q = t_1, \dots, t_n$  une requête CO. Les poids des nœuds feuilles identifiés dans l'arbre du document sont calculés grâce à la fonction de similarité  $RSV_m(q, n^f)$  (Retrieval Status Value), où  $m$  est le modèle de RI considéré.

$$RSV_m(q, n^f) = \sum_{i=1}^n w_i^q \cdot w_i^{n^f} \quad (2)$$

où  $w_i^q$  et  $w_i^{nf}$  sont respectivement le poids du terme  $i$  dans la requête  $q$  et le nœud feuille  $nf$ , le calcul de ces poids dépendant du modèle  $m$  considéré. Nous avons testé plusieurs fonctions, présentées dans la section 4.

### 3.2 Propagation de la pertinence

Une valeur de pertinence est ensuite calculée pour chaque nœud de l'arbre de document, en utilisant les poids des nœuds feuilles qu'il contient. Les termes apparaissant près de la racine d'un sous-arbre paraissent plus porteurs d'information pour le nœud associé que ceux situés plus bas dans le sous-arbre. Il semble ainsi intuitif que plus grande est la distance entre un nœud et son ancêtre, moins il contribue à sa pertinence. Par exemple, sur l'arbre de la figure 2 et par rapport à la requête "hypertexte", le nœud feuille  $nf_3$  doit plus participer au calcul de la pertinence du nœud *section*  $n_5$  que du nœud *article*  $n_1$ . Nous modélisons cette intuition par l'utilisation dans la fonction de propagation du paramètre  $dist(n, nf_k)$ , qui représente la distance entre le nœud  $n$  et un de ses nœuds feuille  $nf_k$  dans l'arbre du document, c'est-à-dire le nombre d'arcs séparant les 2 nœuds. Il paraît aussi intuitif que plus un nœud possède de nœuds feuilles pertinents, plus il est pertinent. Nous introduisons alors dans la formule de propagation le paramètre  $|F_n^p|$ , qui est le nombre de nœuds feuilles descendants de  $n$  ayant un score non nul. La valeur de pertinence  $p_n$  d'un nœud est alors calculée selon la formule 3 :

$$p_n = |F_n^p| \cdot \sum_{nf_k \in F_n} \alpha^{dist(n, nf_k)-1} \cdot (RSV_m(q, nf_k)) \quad (3)$$

où  $F_n$  est l'ensemble des nœuds feuilles  $nf_k$  descendants de  $n$ , et  $\alpha \in ]0..1]$  est un paramètre permettant de quantifier l'importance de la distance séparant les nœuds dans la formule de propagation.

Les nœuds sont ensuite renvoyés à l'utilisateur par ordre décroissant de pertinence à la requête.

On trouvera sur la figure 2 une illustration de notre méthode de propagation de la pertinence avec la requête "Moteur de recherche". Les nœuds feuilles  $nf_1, nf_5, nf_8$  et  $nf_9$  ont un poids  $> 0$  pour la requête et propagent ce poids dans l'arbre du document. Les nœuds  $n_1, n_2, n_3, n_8, n_9, n_{10}, n_{13}$  et  $n_{14}$  seront renvoyés à l'utilisateur par ordre décroissant de pertinence.

## 4 PROPOSITIONS

### 4.1 Pondération des termes d'indexation

Comme nous l'avons vu dans l'état de l'art, le calcul du poids des termes au sein des nœuds feuilles n'est pas un problème trivial. Ce poids doit modéliser l'importance du terme dans le nœud feuille, mais aussi au sein du document et de la collection. Le calcul de  $w_i^j$  dépend du modèle de pondération

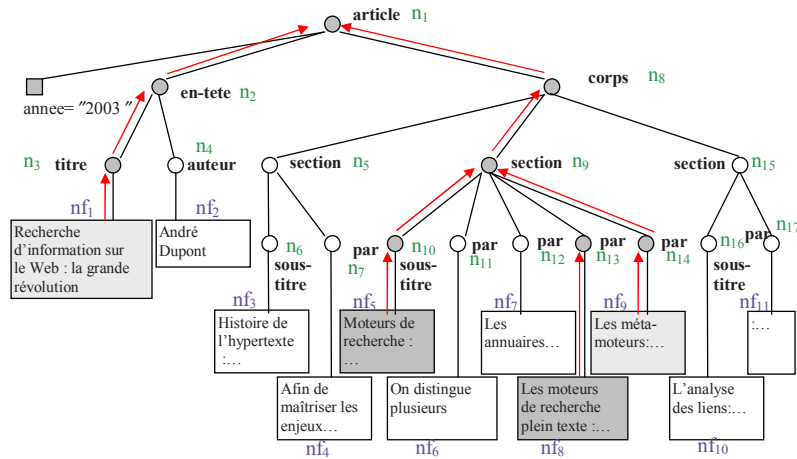


FIG. 2 – Exemple de propagation de la pertinence dans un arbre XML

considéré.

De nombreux paramètres peuvent entrer en compte pour la pondération des termes d'indexation. Des expérimentations précédentes [22] nous ont montré que l'introduction de la taille du nœud feuille (c'est-à-dire le nombre de termes du nœud feuille) et de la taille moyenne des nœuds feuilles de la collection au niveau des formules de pondération ne permettait pas d'améliorer les performances de notre modèle. Dans cet article, nous nous proposons de comparer les facteurs suivants :

- $tf_i^{nf}$  la fréquence du terme  $t_i$  dans le nœud feuille  $nf$
- $idf_i$  la fréquence inverse de document pour le terme  $t_i$ , définie par :

$$idf_i = \log\left(\frac{|D|}{|d_i|}\right) \quad (4)$$

où  $|D|$  est le nombre total de document de la collection et  $|d_i|$  est le nombre de documents contenant le terme  $t_i$

- $ief_i$  la fréquence inverse d'élément pour le terme  $t_i$ , qui est une adaptation de la formule  $idf_i$  à la granularité de l'information que nous traitons (on évalue le poids d'un terme dans un nœud feuille et non plus dans un document).  $ief_i$  est défini de la façon suivante :

$$ief_i = \log\left(\frac{|F_c|}{|nf_i|}\right) \quad (5)$$

où  $|F_c|$  est le nombre total de nœuds feuilles de la collection et  $|nf_i|$  est le nombre de nœuds feuilles de la collection contenant le terme  $t_i$

- $ief_i^d$  la fréquence inverse d'élément dans le document pour le terme  $t_i$ .

$ief_i^d$  est défini comme suit :

$$ief_i^d = \log\left(\frac{|F_d|}{|nf_i^d|}\right) \quad (6)$$

où  $|F_d|$  est le nombre total de nœuds feuilles dans le document  $d$  et  $|nf_i^d|$  est le nombre de nœuds feuilles du document  $d$  contenant le terme  $t_i$ .

$tf_i$  permet de rendre compte de l'importance *locale* du terme  $t_i$  dans un élément,  $idf_i$  et  $ief_i$  permettent de rendre compte de l'importance *globale* du terme respectivement dans la collection de documents et la collection d'éléments, et  $ief_i^d$  permet de rendre compte de l'importance *semi-globale* du terme dans la collection d'éléments formée par un document.

Ces différents facteurs ont été combinés de manières diverses, et les résultats obtenus sont présentés dans la section 6.

## 4.2 Evaluation des requêtes

### 4.2.1 Calcul du score des nœuds feuilles

Nous nous proposons d'évaluer ici les formules de pondération des termes utilisées pour le calcul du score des nœuds feuilles (équation 2).

Nous testons tout d'abord une première formule simple, uniquement basée sur la fréquence d'apparition des termes :

$$w_i^q = tf_i^q \quad w_i^{nf} = tf_i^{nf} \quad (7)$$

où  $tf_i^q$  et  $tf_i^{nf}$  sont respectivement la fréquence du terme  $i$  dans la requête  $q$  et le nœud feuille  $nf$ .

Afin de vérifier la nécessité de s'adapter à une nouvelle granularité de l'information, nous testons la fonction  $tf^*idf$ , couramment utilisée en RI. On a alors :

$$w_i^q = tf_i^q \cdot idf_i \quad w_i^{nf} = tf_i^{nf} \cdot idf_i \quad (8)$$

Ces formules sont ensuite adaptées pour tenir compte de la nouvelle granularité de l'information que nous traitons (on ne parle plus de documents mais de nœuds feuilles). Nous utilisons le paramètre  $ief$ , et les formules de pondération des termes sont alors les suivantes :

$$w_i^q = tf_i^q \cdot ief_i \quad w_i^{nf} = tf_i^{nf} \cdot efi \quad (9)$$

Afin d'évaluer l'importance d'un terme au sein d'un document et non plus au sein d'une collection, nous utilisons le paramètre  $ief^d$  :

$$w_i^q = tf_i^q \cdot ief_i^d \quad w_i^{nf} = tf_i^{nf} \cdot ief_i^d \quad (10)$$

Enfin, les paramètres précédents sont combinés pour tenir compte à la fois de l'importance des termes au sein de la collection et des documents :

$$w_i^q = tf_i \quad w_i^{nf} = tf_i^{nf} \cdot idf_i \cdot ief_i^d \quad (11)$$

$$w_i^q = tf_i \quad w_i^{nf} = tf_i^{nf} \cdot ief_i \cdot ief_i^d \quad (12)$$

#### 4.2.2 Pertinence contextuelle

Nous nous proposons ensuite d'évaluer l'impact de la pertinence du document dans son ensemble sur la pertinence des éléments qu'il contient (on parle de pertinence contextuelle). De manière intuitive, cette idée est facilement explicable : le concepteur d'un document suit une certaine unité dans ses idées, même si le contenu du document est hétérogène. La pertinence des unités d'informations du document est alors liée à la pertinence de cette unité de pensée à la requête. Nous introduisons la pertinence contextuelle de deux façons différentes : (i) par *retro-propagation* de la pertinence du nœud racine (c'est-à-dire du document) vers les nœuds internes, et (ii) en triant les éléments en fonction de la pertinence du documents qui les contient.

##### (i) *Retro-propagation*

Nous nous proposons de modifier le calcul de la pertinence d'un nœud  $n$  comme présenté dans l'équation 13, inspirée des travaux présentés dans [17] :

$$\begin{aligned} p'_n &= \rho \cdot |F_n^p| \cdot \sum_{nf_k \in F_n} \alpha^{dist(n, nf_k)-1} \cdot RSV_m(q, nf_k) + (1 - \rho) \cdot p_{racine} \\ &= \rho \cdot p_n + (1 - \rho) \cdot p_{racine} \end{aligned} \quad (13)$$

avec  $p_{racine}$  la pertinence du nœud *racine* du document, calculée d'après l'équation 3.  $\rho \in [0..1]$  est un paramètre servant de pivot et permettant d'ajuster l'importance de la pertinence du nœud racine lors de la rétro-propagation.

##### (ii) *Tri sur la pertinence du document*

Dans les expérimentations que nous avons présentées jusqu'ici, les unités d'informations étaient triées indépendamment les unes des autres en fonction de leur score de pertinence. Nous nous proposons d'étendre l'étude de l'impact du contexte de la manière suivante : (i) nous calculons un score de pertinence pour tous les documents de la collection, grâce au moteur de recherche Mercure [3], (ii) nous calculons un score de pertinence pour tous les éléments de la collection, (iii) nous trions les documents par ordre décroissant de pertinence, et (iv) pour chaque document, nous trions par ordre décroissant de pertinence les éléments qu'il contient.

De cette façon, les éléments sont d'abord triés en fonction de la pertinence du document auquel ils appartiennent puis en fonction de leur propre pertinence.

## 5 LA TÂCHE CO DE LA CAMPAGNE D'ÉVALUATION INEX 2005

### 5.1 Collection et requêtes

Afin d'évaluer la performance des divers Systèmes de Recherche d'Information pour la RI structurée, la campagne d'évaluation INEX met à dispo-

sition des participants une collection de test, des tâches de recherche composées de requêtes de type divers ainsi que les jugements de pertinence associés. La collection de test 2005 complète celle des années précédentes et est composée de plus de 17000 documents provenant de 21 revues IEEE Computer Society parues de 1995 à 2004.

Les expérimentations présentées dans cet article concernent la tâche Content-Only (CO), qui a pour but de retrouver des parties de documents pertinentes sans que l'utilisateur ne donne d'information sur la granularité de l'information à renvoyer (requêtes composées de simples mots-clés). La tâche CO 2005 est composée de 29 requêtes et des jugements de pertinence associés. Les jugements de pertinence pour chaque requête sont effectués par les différents participants. Deux dimensions sont utilisées pour définir la pertinence : l'*exhaustivité* ( $e$ ) et la *spécificité* ( $s$ ). L'exhaustivité est mesurée selon une échelle à 4 niveaux :  $e=2$  exhaustivité élevée,  $e=1$  exhaustivité moyenne,  $e=0$  pas d'exhaustivité et  $e=?$  élément trop petit. La spécificité est mesurée dans un intervalle continu  $[0,1]$  où  $s=1$  représente un élément totalement spécifique.

## 5.2 Mesures de pertinence

Les mesures d'évaluation utilisées durant la campagne 2005 diffèrent des mesures des années précédentes et sont basées sur les mesures  $xCG$  et  $ep/gr$  [14]. Pour obtenir des résultats de performance avec ces mesures, les 2 dimensions de pertinence (exhaustivité et spécificité) sont agrégées en une seule valeur. Deux types de fonction d'agrégation sont utilisées :

- une agrégation "stricte" pour évaluer si un SRI est capable de retrouver des éléments très spécifiques et très exhaustifs

$$f_{stricte}(e, s) = \begin{cases} 1 & \text{si } e = 2 \text{ et } s = 1 \\ 0 & \text{sinon} \end{cases} \quad (14)$$

- une agrégation "généralisée" pour évaluer les éléments selon leur degré de pertinence

$$f_{generalisee}(e, s) = e \cdot s \quad (15)$$

La mesure  $xCG$  cumule les scores de pertinences des éléments de la liste des résultats. Etant donnée une liste triée d'éléments  $xG$  (encore appelée vecteur de gain) dans laquelle les identifiants des éléments sont remplacés par leur score de pertinence, le gain cumulé au rang  $i$ , noté  $xCG[i]$ , est calculé comme la somme des pertinences jusqu'à ce rang :

$$xCG[i] = \sum_{j=1}^i xG[j] \quad (16)$$

Par exemple, soit  $xG_6 = \langle 2, 1, 0, 1, 0, 0 \rangle$  un vecteur de gain jusqu'au rang 6. Le vecteur de gain cumulé sera  $\langle 2, 3, 3, 4, 4, 4 \rangle$ .

Pour chaque requête, on calcule un vecteur de gain idéal  $xCI$  à partir de la base de rappel, en cumulant les scores de pertinences des éléments triés par ordre décroissant. Le  $xCG$  peut alors être comparé au gain idéal. Le  $xCG$  normalisé ( $nxCG$ ) est obtenu par :

$$nxCG[i] = \frac{xCG[i]}{xCI[i]} \quad (17)$$

Pour un rang donné  $i$ , le gain cumulé  $nxCG[i]$  reflète le gain relatif de l'utilisateur accumulé jusqu'à ce rang, comparé à ce qu'il aurait dû atteindre si le système avait produit une liste triée optimale.

Par analogie au gain cumulé, on définit l'effort-précision ( $ep(r)$ ) :

$$ep(r) = \frac{e_{ideal}}{e_{run}} \quad (18)$$

où  $e_{ideal}$  est le rang pour lequel le gain cumulé est atteint par la courbe idéale et  $e_{run}$  est le rang pour lequel le gain cumulé est atteint par le système. La valeur 1 correspond à une performance idéale, pour laquelle l'utilisateur effectue un minimum d'effort pour atteindre un niveau de gain donné.

L'effort-précision est calculé à des points de gain-rappel arbitraires, où le gain-rappel  $gr$  est la valeur du gain cumulé divisé par la valeur totale atteignable du gain cumulé :

$$gr[i] = \frac{xCG[i]}{xCI[n]} \quad (19)$$

avec  $n$  le nombre total de document pertinents.

L'effort-précision à une valeur donnée de gain-rappel mesure l'effort d'un utilisateur pour atteindre un gain relatif au gain total qu'il peut obtenir. La moyenne non interpolée MAep (Mean Average Effort Precision) d'effort-précision est utilisée pour moyenniser les valeurs d'effort-précision pour chaque rang auquel un élément pertinent est renvoyé.

## 6 EXPÉRIMENTATIONS ET RÉSULTATS

### 6.1 Formules de pondération

Les résultats présentés dans le tableau 1 ont été obtenus en utilisant  $\alpha = 1$  dans la formule de propagation (équation [3]). Le but est en effet d'évaluer l'impact de la formule utilisée pour le calcul du poids des termes d'indexation, et non d'évaluer la fonction de propagation. Pour obtenir le score des nœuds internes, les scores des nœuds feuilles sont donc simplement sommés.

On observe une perte très significative de performance lorsque le nouveau facteur  $ief^d$  est utilisé. Ceci montre que la modélisation de l'importance "semi-globale" des éléments au sein des documents n'a pas d'impact sur leur pertinence.

De plus, de manière surprenante, la simple utilisation du facteur  $tf$  permet

		nxCG[10]	nxCG[25]	nxCG[50]	ep/gr - MAP
Généralisée	$tf$ (eq. [7])	0.1555	0.1409	0.1307	0.043
	$tf \cdot idf$ (eq. [8])	0.1277	0.1267	0.1396	0.0413
	$tf \cdot ief$ (eq. [9])	0.1278	0.1281	0.137	0.0438
	$tf \cdot ief^d$ (eq. [10])	0.0704	0.0758	0.0807	0.0213
	$tf \cdot idf \cdot ief^d$ (eq. [11])	0.078	0.0946	0.1036	0.0304
	$tf \cdot ief \cdot ief^d$ (eq. [12])	0.0749	0.0913	0.0981	0.0270
Stricte	$tf$ (eq. [7])	0.0115	0.0233	0.039	0.0009
	$tf \cdot idf$ (eq. [8])	0	0.014	0.035	0.0006
	$tf \cdot ief$ (eq. [9])	0	0.0155	0.0313	0.0006
	$tf \cdot ief^d$ (eq. [10])	0	0.0024	0.0112	0.0002
	$tf \cdot idf \cdot ief^d$ (eq. [11])	0	0.0116	0.0212	0.0002
	$tf \cdot ief \cdot ief^d$ (eq. [12])	0	0.0116	0.0212	0.0002

TAB. 1 – Comparaison des formules de pondération des nœuds feuilles

d’obtenir des résultats aussi bons, voire souvent meilleurs, que ceux obtenus en tenant compte également de l’importance du terme dans la collection de documents (facteur  $idf$ ) ou la collection d’éléments (facteur  $ief$ ).

Dans les expérimentations suivantes, nous conservons les équations [7], [8] et [9] pour évaluer l’introduction du contexte dans l’évaluation de la pertinence des nœuds internes.

## 6.2 Introduction de la pertinence contextuelle

Pour les mêmes raisons que précédemment, on utilise  $\alpha = 1$  dans l’équation [3].

### Retro-propagation

On trouvera sur la figure 3 l’évolution de la mesure nxCG[10] en fonction de  $\rho$  pour la fonction d’agrégation généralisée. Cette évolution, non présentée ici par manque de place, est la même pour d’autres valeurs de nxCG (notamment nxCG[25] et nxCG[50]) et pour la fonction d’agrégation stricte. L’introduction de la pertinence des documents semble être d’une importance capitale, puisque plus on donne d’importance à la pertinence du document pour calculer la pertinence de l’élément (faibles valeurs de  $\rho$ ), plus les performances augmentent. Les performances pour la mesure ep/gr stagnent quant à elles et ne semblent pas dépendre de l’introduction de contexte.

### Tri sur la pertinence du document

Le tableau 2 présente les résultats obtenus en triant les éléments en fonction de la pertinence des documents. On constate que de manière générale, ce tri permet d’améliorer les performances sur les 2 mesures, et pour les 3 fonc-

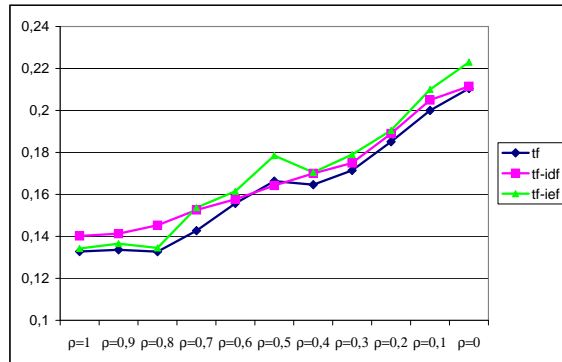


FIG. 3 – Evolution de la mesure nxCG[10] en fonction de  $\rho$ , fonction d'agrégation généralisée

tions de pondération. Ceci tend donc à prouver que le contexte des éléments (ici le document qui les contient) joue un rôle fondamental dans le calcul de leur pertinence.

		nxCG[10]	nxCG[25]	nxCG[50]	ep/gr - MAP
Généralisée	<i>tf</i> (eq. [7])	0.1518	0.1868	0.1935	0.048
	Amélioration	-3.4%	+32,6%	+41.2%	+11.6%
	<i>tf · idf</i> (eq. [8])	0.1481	0.1747	0.1656	0.0472
	Amélioration	+16%	+37,9%	+18.6%	+14.3%
	<i>tf · ief</i> (eq. [9])	0.1419	0.1595	0.168	0.0473
Amélioration	+11%	+24.5%	+22.6%	+8%	
Stricte	<i>tf</i> (eq. [7])	0.0064	0.0207	0.0385	0.00185
	Amélioration	-45%	-12.2%	-1.3%	+100%
	<i>tf · idf</i> (eq. [8])	0.0154	0.0462	0.052	0.0013
	Amélioration	+∞	+138%	+71.2%	+116.6%
	<i>tf · ief</i> (eq. [9])	0.0154	0.0369	0.0536	0.0013
mélioration	+∞	+37,9%	+18.6%	+116.6%	

TAB. 2 – Résultats obtenus par tri des éléments sur la pertinence des documents

### 6.3 Discussion

Les expérimentations que nous venons de présenter montrent qu'il est nécessaire lors de la pondération des termes de modéliser l'importance du terme dans les éléments qui le contiennent (facteur *tf*). La modélisation de

l'importance globale du terme au sein de la collection de documents (facteur  $idf$ ) ou de la collection de nœuds feuilles (facteur  $ief$ ) ne semble pas avoir d'impact. Nous avons également proposé un facteur modélisant l'importance semi-globale d'un terme (facteur  $ief^d$ ) au sein d'un document, mais ce facteur entraîne une baisse des performances pour toutes les mesures utilisées. Nous avons également montré que l'introduction du contexte des éléments (c'est-à-dire de la pertinence du document qui les contient) permet de mieux trier les éléments.

Les résultats présentés ci-dessus ont utilisé la valeur  $\alpha = 1$  dans l'équation 3 pour calculer la pertinence des éléments. Ceci avait pour but d'évaluer l'impact des différentes formules utilisées sans que les résultats soient biaisés par les paramètres de la fonction de propagation. Nous avons reconduit les expérimentations présentées plus haut en faisant cette fois-ci varier  $\alpha$ . Ces résultats pour les fonctions  $nxCG[10]$  et MAP sont présentés sur les figures 4 et 5.

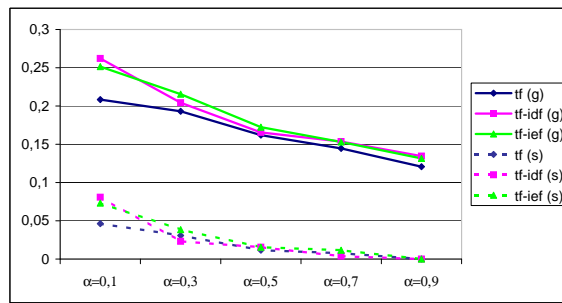


FIG. 4 – Evolution de la mesure  $nxCG[10]$  en fonction d'  $\alpha$ , fonctions d'agrégation généralisée (g) et stricte (s)

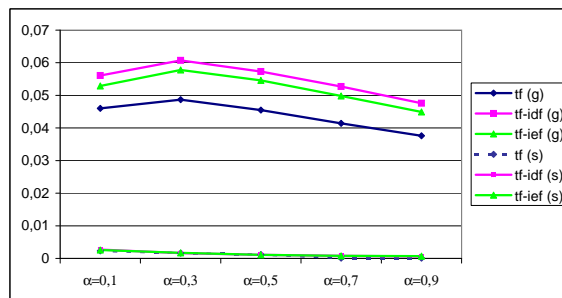


FIG. 5 – Evolution de la mesure  $ep/gr$ -Map en fonction d'  $\alpha$ , fonctions d'agrégation généralisée (g) et stricte (s)

Les meilleurs résultats sont obtenus pour de petites valeurs d' $\alpha$ , c'est-à-dire lorsque les éléments de plus petite taille sont privilégiés. Ceci contredit nos précédentes expérimentations sur les collections de 2003 et 2004 [21], pour lesquelles la valeur optimale de  $\alpha$  était aux alentours de 0.7. Une explication peut venir du nouveau processus utilisé pour le jugement des requêtes en 2005. Alors que les années précédentes, les participants jugeaient en même temps l'exhaustivité et la spécificité des éléments retournés par les différents systèmes, seul le jugement de l'exhaustivité était requis cette année. Les juges devaient sélectionner les parties exhaustives des documents (indépendamment des résultats renvoyés par les systèmes), et la spécificité des éléments était ensuite déduite par le système collectant les jugements [15]. De cette façon, les éléments spécifiques sont préférés par les participants, ce qui explique nos résultats. Si l'on examine maintenant la formule de pondération utilisée, on constate que l'introduction des facteurs *ief* et *idf* permet tout de même d'améliorer les performances par rapport à la seule utilisation de *tf*.

La figure 6 montre les résultats obtenus sur la mesure ep/gr-MAP en combinant la meilleure valeur de  $\alpha$  (0.1) avec le tri des éléments sur la pertinence des documents. On constate une baisse sensible des performances, ce qui contredit les résultats présentés en 6.2. Cette baisse des performances peut être constatée pour toutes les formules de pondération, toutes les mesures et toutes les fonctions d'agrégation. On observe parallèlement une stagnation

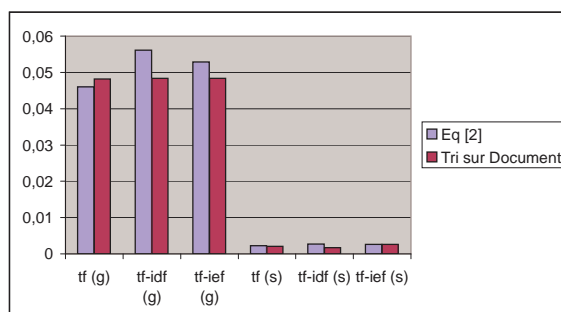


FIG. 6 – Evolution de la mesure ep/gr-MAP en triant ou non les éléments sur la pertinence du document

des performances lorsque la pertinence contextuelle est introduite par retro-propagation (résultats présentés sur la figure 7). Concernant les formules de pondération, l'intérêt des facteurs *idf* et *ief* est de nouveau souligné.

Ces résultats contredisent eux aussi ceux obtenus dans [21], pour lesquels la pertinence contextuelle était introduite par retro-propagation ou en triant les éléments en fonction de la pertinence des documents calculée par propagation (nous calculons ici la pertinence des documents grâce au moteur de

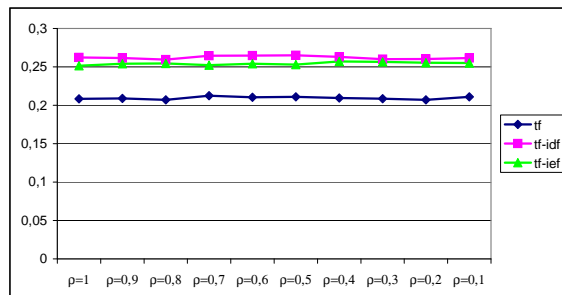


FIG. 7 – Evolution de la mesure ep/gr-MAP en triant ou non les éléments sur la pertinence du document

recherche Mercure). Alors que l'introduction du contexte des éléments était d'une importance fondamentale sur les jeux de test 2003 et 2004, ce dernier semble ne pas avoir d'impact sur le jeu de test 2005, ou plutôt sur les mesures de performances utilisées en 2005. Ceci montre la nécessité d'aboutir à des modèles utilisateurs et mesures stables pour évaluer correctement les systèmes.

En résumé, si l'on privilégie la dimension de spécificité pour l'évaluation de la pertinence (comme lors de la campagne 2005), des nœuds de petite taille doivent être renvoyés (petites valeurs d' $\alpha$ ) et un calcul "simple" de la pertinence (c'est-à-dire d'après l'équation 3) permet d'obtenir de meilleurs résultats. Si les dimensions de spécificité et d'exhaustivité sont considérées à part égale (comme lors des campagnes d'évaluation 2003 et 2004), les nœuds d'assez grande taille doivent être privilégiés (valeurs assez élevées de  $\alpha$ ) et la pertinence contextuelle joue un rôle important dans l'évaluation de la pertinence des éléments[21]. Dans les deux cas, l'importance globale d'un terme dans la collection (facteurs *idf* ou *ief*) doit être introduite dans la fonction de pondération (équation 2) pour évaluer au mieux la pertinence des nœuds. Enfin, notons tout de même les relatives bonnes performances de notre système comparé aux soumissions officielles d'INEX. Pour le jeu de test 2005, nous aurions été classé dans le top 5 ou 10 pour presque toutes les mesures en privilégiant les nœuds très spécifiques ( $\alpha = 0.1$ ).

## 7 CONCLUSION

Nous avons présenté dans cet article quelques pistes pour l'évaluation de la pertinence des éléments en RI structurée.

Concernant la pondération des éléments, nous avons montré que les formules utilisées devaient principalement prendre en compte l'importance locale (facteur *tf*). L'introduction de l'importance globale du terme dans la

collection (facteurs *idf* ou *ief*) permet d'améliorer de manière significative les résultats (pour des paramètres optimaux de la fonction de propagation). Enfin, l'importance semi-globale du terme dans le document ne permet pas un meilleur calcul de la pertinence des éléments.

En ce qui concerne l'évaluation de la pertinence des éléments, nous avons montré que sur le jeu de test d'INEX 2005, l'introduction de la pertinence contextuelle des éléments pour le calcul de leur pertinence ne permet pas d'améliorer significativement les résultats pour des valeurs optimales des paramètres de la fonction de propagation. Ceci contredit certains résultats présentés dans [21, 23] sur les jeux de test d'INEX 2003 et 2004. Ces contradictions peuvent être expliquées par le changement de processus pour les jugements de pertinence ainsi que par les nouvelles mesures de performances utilisées en 2005. Cependant, il devient nécessaire d'aboutir à des mesures stables pour que la recherche en RI structurée puisse progresser sur des bases solides.

Nos travaux futurs vont évaluer les différents paramètres présentés dans cet article pour la tâche de recherche consistant à ne renvoyer que les éléments répondant de manière la plus spécifique possible aux attentes de l'utilisateur (pas d'imbrication possible des résultats). Cette évaluation était impossible les années précédentes, les mesures de performances utilisées n'étant pas adaptées.

## RÉFÉRENCES

- [1] M. Abolhassani et N. Fuhr. Applying the divergence from randomness approach for content-only search in XML documents. In *Proceedings of ECIR 2004, Sunderland*, pages 409–419, 2004.
- [2] Vo Ngoc Anh et Alistair Moffat. Compression and an ir approach to XML retrieval. In *Proceedings of INEX 2002 Workshop, Dagstuhl, Germany*, 2002.
- [3] M. Boughanem, T. Dkaki, J. Mothe et C. Soule-Dupuy. Mercure at TREC-7. In *Proceedings of TREC-7*, 1998.
- [4] D. Carmel, Y. Maarek, M. Mandelbrot et A. Soffer. Searching xml documents via xml fragments. In *Proceedings of SIGIR 2003*, pages 151–158, 2003.
- [5] C.J. Crouch, S. Apte et H. Bapat. An IR approach to XML retrieval based on the extended vector model. In *Proceedings of INEX 2002 Workshop, Dagstuhl, Germany*, pages 98–99, 2002.
- [6] M.E. Frisse. Searching for information in a hypertext medical handbook. In *Proceedings of ACM Hypertext Conference, Chapel Hill, NC*, pages 57–66, 1987.

- [7] N. Fuhr et K. Grossjohann. XIRQL : a query language for information retrieval in XML documents. In *In Proceedings of SIGIR 2001, Toronto, Canada, 2003*.
- [8] M. Fuller, E. Mackie, R. Sacks-Davis et R. Wilkinson. Structural answers for a large structured document collection. In *Proceedings of ACM SIGIR 1993, Pittsburgh*, pages 204–213, 1993.
- [9] T. Grabs et H.-J. Scheck. Flexible information retrieval from XML with PowerDB XML. In *Proceedings in the First INEX Workshop*, pages 26–32, 2002.
- [10] Norbert Gövert, Mohamed Abolhassani, Norbert Fuhr et Kai Grossjohann. Content-oriented XML retrieval with HyReX. In *Proceedings of the first INEX Workshop, Germany, 2002*.
- [11] V. Kakade et P. Raghavan. Encoding XML in vector spaces. In *Proceedings of ECIR 2005, Saint Jacques de Compostelle, Spain, 2005*.
- [12] Jaap Kamps, Maarten de Rijke et Borkur Sigurbjornsson. Length normalization in XML retrieval. In *Proceedings of SIGIR 2004, Sheffield, England*, pages 80–87, 2004.
- [13] G. Kazai, M. Lalmas et T. Roelleke. Focused document retrieval,. In *9th International Symposium on String Processing and Information Retrieval, Lisbon, Portugal, September 2002*.
- [14] Gabriella Kazai et Mounia Lalmas. INEX 2005 evaluation metrics. In *Pre-proceedings of INEX 2005, Dagstuhl, Allemagne, November 2005*.
- [15] Mounia Lalmas et Benjamin Piwowarski. INEX 2005 relevance assessment guide. In *INEX 2005 Pre-proceedings, Dagstuhl, Allemagne, pages 391–401, november 2005*.
- [16] J. A. List, V. Mihajlovic, A.P.de Vries, G. Ramirez et D. Hiemstra. The TIJAH XML-IR system at Inex 2003. In *Proceedings of INEX 2003, Dagstuhl, Germany, 2003*.
- [17] Yosi Mass et Matan Mandelbrod. Component ranking and automatic query refinement for XML retrieval. In *Proceedings of INEX 2004, pages 134–140, 2004*.
- [18] P. Ogilvie et J. Callan. Using language models for flat text queries in XML retrieval. In *Proceedings of INEX 2003 Workshop, Dagstuhl, Germany, pages 12–18, 2003*.
- [19] B. Piwowarski, G-E. Faure et P. Gallinari. Bayesian networks and INEX. In *Proceedings in the First INEX Workshop, December 2002*.
- [20] T. Roelleke, M. Lalmas, G. Kazai, J. Ruthven et S. Quicker. The accessibility dimension for structured document retrieval. In *Proceedings of ECIR 2002, 2002*.
- [21] Karen Sauvagnat. *Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés*. PhD thesis, Toulouse : Université Paul Sabatier, 2005.

- [22] Karen Sauvagnat et Mohand Boughanem. A la recherche de noeuds informatifs dans des corpus de documents XML - ou pourquoi on a toujours besoin de plus petit que soi... In *Actes de CORIA 05, Grenoble, France, 2005*.
- [23] Karen Sauvagnat, Lobna Hlaoua et Mohand Boughanem. XML retrieval : what about using contextual relevance ? In *ACM Symposium on Applied Computing (SAC) - IAR (Information Access and Retrieval)*, Dijon, 23-27 avril 2006.
- [24] T. Schileder et H. Meuss. Querying and ranking XML documents. *Journal of the American Society for Information Science and Technology*, 53(6) :pages 489–503, 2002.
- [25] A. Theobald et G. Weikum. The index-based XXL search engine for querying XML data with relevance ranking. In *EDBT 2002, Prague, Czech Republic*, pages 477–495.
- [26] A. Trotman. Choosing document structure weights. *Information Processing and Management*, 41(2) :pages 243–264, March 2005.
- [27] Felix Weigel, Klaus U. Schulz et Holger Meuss. Ranked retrieval of structured documents with the STerm vector space model. In *Proceedings of INEX 2004, Dagstuhl, Allemagne*, pages 126–133, 2004.
- [28] J.E. Wolff, H. Flörke et A.B. Cremers. Searching and browsing collections of structural information. In *Proceedings of IEEE advances in digital libraries, Washington, 2000*, pages 141–150, 2000.
- [29] Haifa Zargayouna. Contexte et sémantique pour une indexation de documents semi-structurés. In *Actes de CORIA 04, Toulouse, France*, pages 161–178, 2004.
- [30] G. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.