Recherche d'information XML utilisant un principe de vote

Gilles Hubert *, — Josiane Mothe *,** — Sandra Poulain ***

** Institut de Recherche en Informatique de Toulouse*
Equipe SIG/EVI
118 route de Narbonne, F-31062 Toulouse cedex
{hubert, mothe, poulain}@irit.fr

*** ERT34*
Institut Universitaire de Formation des Maîtres
56 avenue de l'URSS, F-31400 Toulouse

RÉSUMÉ. *Cet article décrit une approche pour la recherche d'information dans des collections de documents XML. Cette approche utilise une méthode de vote pour déterminer les éléments XML répondant à une requête. Une requête peut combiner des informations sur le contenu recherché, sur la granularité des éléments recherchés et sur les éléments structurels associés aux concepts recherchés. La méthode proposée a été expérimentée et évaluée dans le cadre de la campagne INEX 2004.*

ABSTRACT. *This paper describes an approach for information retrieval from collections of XML documents. This approach uses a voting method to identify the XML elements to retrieve according to a query that can combine elements on the content and on the structure of the elements to be retrieved. The approach presented was experimented and evaluated within the INEX'2004 framework.*

MOTS-CLÉS : *Recherche d'information, document XML, méthode de vote, évaluation, INEX.*

KEYWORDS: *Information Retrieval, XML document, voting method, evaluation, INEX.*

1. Introduction

L'utilisation du langage XML (eXtensible Markup Language) [BRA 04] est de plus en plus répandue notamment pour les publications à caractère scientifique. Ainsi, certains systèmes de Recherche d'Information (RI) évoluent pour intégrer l'exploitation de la structure des documents et combiner recherche textuelle et recherche structurelle. Les systèmes de RI gèrent généralement des documents entiers, c'est-à-dire que l'unité d'indexation et de recherche sont les documents entiers. Des travaux se sont également intéressés à retrouver des passages (paragraphes, phrases) dans le cadre de documents non structurés. Dans le cadre de documents structurés, de nouvelles possibilités de recherche sont offertes à l'utilisateur qui peut non seulement choisir la granularité qu'il souhaite pour la réponse et indiquer des contraintes structurelles sur les éléments à restituer. INEX [FUH 04b] est un programme qui s'intéresse à l'évaluation de SRI permettant ce type d'interrogation et d'accès aux documents XML.

Dans cet article, nous présentons une approche originale de RI structurée basée sur un principe de vote. Cette approche a été évaluée dans le cadre de INEX.

Cet article est organisé comme suit. La section 2 présente un aperçu des différentes approches proposées dans le domaine de la RI XML. Dans la section 3, nous introduisons les aspects définissant le besoin d'information auxquels nous avons voulu répondre et nous détaillons comment notre méthode les prend en compte. La section 4 décrit l'initiative d'évaluation INEX'2004 ainsi que les résultats des expérimentations que nous avons réalisées dans ce cadre. Nous concluons l'article dans la section 5 en analysant les résultats et en présentant également les perspectives de travaux qui en découlent.

2. Travaux du domaine

Plusieurs propositions relatives à la RI XML sont issues de travaux sur les langages de requêtes de bases de données. Il s'agit de proposer des langages de requête XML qui introduisent une correspondance approximative pour les prédicats relatifs au contenu ou à la structure. Par exemple, XIRQL [FUH 04a] introduit un principe de classement de documents résultats suivant un calcul probabiliste à partir d'une pondération des termes d'indexation des requêtes et des documents. XIRQL introduit également des types de données (comme les noms de personnes ou les dates) avec prédicats vagues et des imprécisions structurelles (assimilation éléments et attributs, similarités entre noms d'éléments). D'autres travaux proposent un principe de relaxation de structure de requête comme par exemple le système FlexPath [AME 04] pour des requêtes XPath [CLA 99]. Le principe est de générer un ensemble de requêtes où les expressions structurelles sont élargies et de fusionner et réordonner les résultats obtenus en réponse à ces requêtes.

Dans le domaine de la RI, le modèle vectoriel [SAL 75] et les mesures de similarité comme la mesure cosinus ont inspiré des approches s'intéressant aux documents XML. Par exemple, [CAR 03] propose une décomposition de requêtes en fragments XML. Des paires (terme, contexte) sont utilisées comme unité d'indexation. Un contexte est un chemin de localisation d'un élément XML de type Xpath. Les travaux étendent la formule de similarité Cosinus pour prendre en compte la similarité entre contextes et la similarité entre termes. [CRO 04] propose une approche utilisant le modèle vectoriel étendu. Un vecteur représentant un document regroupe un ensemble de sous-vecteurs correspondant à différentes classes d'informations (par exemple, nom d'auteur, résumé). La similarité entre vecteurs étendus est définie comme la combinaison linéaire des similarités entre sous-vecteurs correspondants. [GRA 02] propose de générer automatiquement à la volée l'espace vectoriel approprié à une requête à partir de l'indexation des documents XML uniquement pour les types d'éléments XML de base. Une mesure statistique tf.ief dérivée de tf.idf est utilisée pour travailler au niveau des éléments XML plutôt qu'au niveau des documents.

Les modèles de langage [PON 98] ont également été utilisés dans le contexte de documents XML. Par exemple, [KAM 04] propose un modèle de langage multinomial avec lissage utilisant des index des documents à différents niveaux (article, élément). Une normalisation de la longueur des éléments XML est proposée pour pallier l'introduction de biais dans les résultats par un lissage trop important. Un modèle de langage hiérarchique est également proposé dans [OGI 04] où les documents XML sont représentés sous forme d'arbres (un nœud correspond à une balise dans le document). Pour chaque composant XML un modèle est estimé en utilisant une interpolation linéaire du contenu du composant, des modèles de ses enfants et du modèle de son parent. L'approche proposée dans [PIW 04] utilise quant à elle les réseaux bayésiens. La structure d'un réseau reflète la hiérarchie de documents. Les scores des éléments sont calculés récursivement au travers du réseau du plus gros élément racine du réseau (corpus) vers les feuilles (composants XML les plus fins), l'état d'un élément dépendant de l'état de son parent.

Notre approche se rapproche plutôt des modèles vectoriels bien que le principe que nous proposons permette une meilleure prise en compte des contributions individuelles de chaque élément. Le score calculé pour un élément repose directement sur le cumul des occurrences des termes d'une requête dans un élément XML. Ce principe rejoint la notion de vote (cumul de voix) si l'on considère un élément XML comme un candidat et chaque apparition d'un terme de la requête comme une voix en sa faveur.

3. Une méthode de vote pour la recherche d'information

3.1. *Objectif*

La plupart des systèmes de RI qui manipulent des documents non structurés permettent d'effectuer des recherches suivant le contenu des documents et restituent les documents en adéquation avec le besoin d'information exprimé. Lorsque l'on manipule des documents XML, la recherche suivant le contenu est toujours possible mais il est également possible de s'appuyer sur la structure XML pour affiner le besoin d'information. Notre objectif est de proposer une méthode qui permettent de traiter un besoin d'information « classique », comme par exemple rechercher les éléments qui traitent d'un certain nombre de concepts, jusqu'à une requête « plus affinée », comme par exemple rechercher les sections qui traitent de certains concepts et qui appartiennent à des articles qui traitent d'autres concepts.

Les éléments qui constituent un besoin d'information auxquels notre méthode veut répondre sont :

– le contenu recherché et éventuellement celui non recherché c'est-à-dire la possibilité d'indiquer les concepts souhaités et les concepts non souhaités avec éventuellement une échelle de valeur (par exemple, souhaité et fortement souhaité),

– les structures des éléments dans lesquels le contenu est recherché c'est-à-dire la possibilité d'indiquer la localisation des concepts recherchés dans la structure hiérarchique d'un document,

– la structure des éléments à restituer c'est-à-dire la possibilité d'indiquer la granularité des éléments recherchés voire leur localisation dans la hiérarchie d'un document.

Comme dans tout principe de RI, l'objectif est de définir une méthode qui permette de retrouver les éléments de la collection qui répondent au mieux au besoin d'information exprimé par l'utilisateur sans exiger la vérification stricte des indications définissant ce besoin. De plus, l'objectif est de proposer une méthode qui prend en compte les éléments constituant le besoin d'information de manière incrémentale. La méthode appliquée doit être la même, qu'il y ait simplement des indications de contenu recherché ou des indications de contenu combinées avec des indications de structure sur ce contenu et de granularité des éléments à restituer.

3.2. *Principe*

L'originalité de notre approche réside dans l'application d'un principe assimilable à un vote plutôt qu'à un calcul de similarité « classique ». L'approche est basée sur la méthode Vector Voting [PAU 00]. Cette méthode s'apparente à la méthode HVV (Hyperlink Vector Voting) utilisée dans le contexte du Web pour calculer la pertinence d'une page en fonction des sites web qui la référencent

[LI 98]. Notre approche a précédemment été utilisée pour la classification automatique de documents textuels suivant des hiérarchies de concepts [HUB 03].

La stratégie choisie considère que plus les termes de la requête apparaissent dans un texte plus le score obtenu par le texte est important. Chaque terme de la requête qui apparaît dans le texte apporte des « voies » en fonction de sa présence dans le texte et en fonction de son importance dans la requête. Les textes qui obtiennent les meilleurs scores sont élus pour être restitués à l'utilisateur.

3.3. *Extraction d'information*

Du point de vue de la collection, les documents XML sont considérés comme des ensembles de composants ayant une localisation dans la structure hiérarchique d'un document et un contenu textuel. Les concepts sont extraits automatiquement des éléments textuels avec leur nombre d'occurrences dans le composant. Ils sont sauvegardés avec les chemins de localisation dans les documents XML en utilisant une notation de type Xpath [CLA 99]. L'extraction de concepts met en œuvre notamment la suppression des mots vides. D'autres traitements optionnels peuvent être appliqués comme la radicalisation selon l'algorithme de Porter [POR 80]. Du point de vue des requêtes, les termes sont extraits automatiquement du contenu textuel selon le même principe. Les indications structurelles relatives à la localisation des concepts recherchés sont automatiquement identifiées et reliées aux termes correspondants. De même, les indications sur la structure des éléments à restituer sont identifiées et liées aux requêtes concernées de manière automatique.

3.4. *Fonction de vote*

La fonction de vote tient compte de l'importance de chaque terme de la requête dans l'élément XML. Elle prend également en compte l'importance de chaque terme dans la requête.

$$Vote(E,T) = \sum_{\forall t \in T} F(t,E) \cdot \frac{F(t,T)}{S(T)}$$

où T est la requête et E est un élément XML

$F(t,E)$	Ce facteur mesure l'importance du terme t dans l'élément XML E. F(t,E) correspond au nombre d'occurrences du terme t dans l'élément E.
$\frac{F(t,T)}{S(T)}$	Ce facteur mesure l'importance du terme t dans la représentation de la requête T. F(t,T) correspond au nombre d'occurrences du terme t dans la requête T et S(T) correspond à la taille (nombre de termes) de T.

La fonction de vote combine deux facteurs : la présence d'un terme dans l'élément XML et l'importance du terme dans la requête.

3.5. *Fonction de score*

Le score final obtenu par un élément XML vis-à-vis d'une requête donnée est fonction du vote qu'il obtient mais également de l'importance de la représentation de la requête au sein de l'élément XML et de la couverture entre ces deux éléments. L'objectif de la couverture est d'assurer que seuls seront sélectionnés les éléments dans lesquels la requête est suffisamment représentée. La couverture est un seuil relatif au pourcentage de termes de la requête qui apparaissent dans le texte d'un élément. Par exemple, une couverture de 50% implique qu'au moins la moitié des termes décrivant la requête doivent apparaître dans un élément pour le sélectionner.

La fonction de score est la suivante :

Si $\frac{NT(T,E)}{S(T)} \geq CT$ Alors $Score(E,T) = Vote(E,T) \cdot \varphi^{(\frac{NT(T,E)}{S(T)})}$

Sinon $Score(E,T) = 0$

où

φ est un entier naturel supérieur à 1 et CT est le seuil de couverture

$\frac{NT(T,E)}{S(T)}$ Ce facteur mesure le taux de présence des termes de la requête dans le texte (représentation de la requête dans le texte). S(T) correspond au nombre de termes de la requête et NT(T,E) correspond au nombre de termes de la requête T qui apparaissent dans l'élément XML E.

Nous avons choisi de donner de l'importance à ce facteur (i.e. taux de présence des termes de la requête dans le texte) en le considérant comme puissance d'une constante. Le choix de la constante permet de faire varier l'influence de ce facteur dans la fonction de score.

3.6. *Propagation de scores*

La propagation de scores permet de prendre en compte la structure hiérarchique des documents XML. L'hypothèse sur laquelle s'appuie notre approche considère qu'un élément XML qui contient un composant sélectionné comme pertinent est aussi pertinent et qu'il est d'autant plus pertinent qu'il contient plusieurs composants pertinents. Le score d'un composant sélectionné est répercuté sur les éléments qu'il compose mais à un degré moindre. Ce principe favorise la sélection de composants pertinents. Néanmoins, lorsqu'un élément est composé de plusieurs

composants sélectionnés pertinents, les scores des composants sont cumulés et répercutés sur l'élément composé. Ce principe privilégie donc les éléments composés de plusieurs éléments pertinents. Le score d'un composant sélectionné est propagé vers les éléments qu'il compose après application d'un facteur réducteur fonction de la distance entre le composant et l'élément composé, comme suit :

$$\forall E_a \text{ ancestor of } E \quad \text{and} \quad d(E_a,E)\cdot\alpha<1$$
$$Score'(E_a,T) = Score(E_a,T) + (1-d(E_a,E)\cdot\alpha)\cdot Score(E,T)$$

où

α est une constante réelle,

E est un élément XML,

E_a est un élément XML de la hiérarchie de composition de l'élément E

$d(E_a,E)$ est la distance entre E_a et E (par exemple,. dans le xpath /article/bdy/sec/p[2], la distance entre p[2] et bdy est égale à 2 i.e. d(bdy,p[2])=2).

$Score(E_a,T)$ est le score directement lié au contenu de E_a

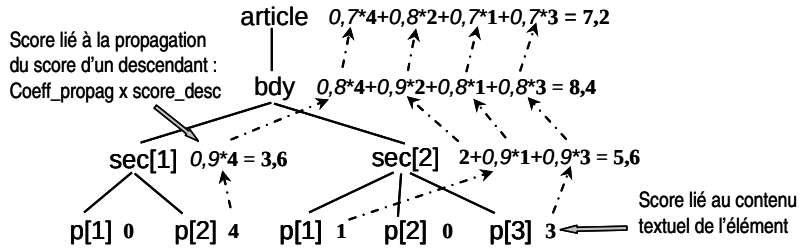

Figure 1. *Exemple de propagation de scores avec α = 0,1*

De plus, notre méthode permet de préciser des préférences sur les termes à l'image de la proposition XQUERY Full-Text [BUX 03] qui préconise les niveaux MUST, SHOULD, MAY ou l'utilisation de préfixes '+' et '–' dans INEX (cf 4.1.2).

Pour intégrer les préférences sur des termes, un coefficient est associé à chaque catégorie de préférence. Ce coefficient est appliqué dans la fonction de vote comme suit :

$$Vote(E,T) = \sum_{\forall t\in T} sc(t,T)\cdot F(t,E)\cdot\frac{F(t,T)}{S(T)}$$

avec sc(t,T) coefficient réel associé à la catégorie de préférence du terme t

3.7. *Prise en compte de contraintes structurelles*

Nous distinguons dans notre approche deux types de contraintes structurelles exprimables dans les requêtes : les contraintes structurelles associées au contenu recherché et les contraintes structurelles associées au type de résultat recherché.

Les contraintes structurelles associées au contenu peuvent être des indications sur la localisation des termes recherchés (par exemple, une recherche portera sur des éléments qui possèdent un paragraphe relatif à la *RI* ainsi qu'une section relative à *une évaluation*). Chaque contrainte de ce type est associée à chacun des termes sur lesquels elle est définie. Ces contraintes sont prises en compte comme suit :

$$Vote(E,T) = \sum_{\forall t \in T} (1+\beta) \cdot F(t,E) \cdot \frac{F(t,T)}{S(T)}$$

avec β réel tel que si E vérifie la contrainte sur le terme t alors β>0,sinon β=0.

Plus la valeur du coefficient β est grande, plus les éléments vérifiant les contraintes sur le contenu sont privilégiés.

D'autre part, les contraintes structurelles associées au type de résultat recherché précisent la nature des éléments XML constituant le résultat notamment du point de vue de la granularité de l'élément (par exemple, une requête portera sur la recherche de sections, une autre sur des articles complets). Ce type de contrainte est associé à la requête et est pris en compte dans une étape supplémentaire qui modifie le score obtenu par un élément XML selon qu'il vérifie ou non la contrainte associée au résultat. La contrainte peut être prise en compte de manière stricte (les éléments qui ne respectent pas la contrainte ne sont pas sélectionnés) ou de manière relative (les éléments qui vérifient la contrainte sont privilégiés). Le score d'un élément XML qui vérifie la contrainte est augmenté comme suit :

$$Score(E,T) = \gamma \cdot Vote(E,T) \cdot f\left(\frac{NT(T,E)}{S(T)}\right)$$

Avec γ réel tel que *si* E vérifie contrainte de résultat *alors* γ>1,0 *sinon* γ=1,0

4. Expérimentations et évaluation

Notre proposition a été évaluée dans le cadre de l'initiative INEX'2004 (INitiative for the Evaluation of XML retrieval) [HUB 05]. INEX offre un cadre pour l'évaluation de la RI dans des collections XML [FUH 04b].

4.1. *L'initiative INEX*

INEX fournit des jeux de tests (collection+requêtes+résultats) et des méthodes d'évaluation. Elle permet aux participants d'évaluer et de comparer leurs résultats.

Différentes tâches ont été proposées pour la session 2004. Nous nous intéressons ici à la tâche de recherche ad-hoc destinée à évaluer des systèmes de RI répondant à des recherches soit concernant uniquement le contenu, soit mêlant contenu et indications de structure.

4.1.1. Collection

La collection de documents correspond à approximativement 12000 articles publiés par la IEEE Computer Society de 1995 et 2002. Les articles sont tous structurés suivant le langage XML et respectent la même DTD. La collection rassemble plus de 8 millions d'éléments XML. Les éléments XML sont de longueurs et de granularités variables (par exemple, titre, paragraphe ou article).

4.1.2. Requêtes

INEX introduit deux types de requêtes :

– les requêtes de type CO (Content Only) relatives au contenu. Elles décrivent uniquement le contenu souhaité des éléments XML recherchés,

– les requêtes de type CAS (Content And Structure) qui combinent contenu et références explicites à la structure XML.

Les deux types de requêtes CO et CAS sont constituées de quatre parties : titre, description, narration et mots-clés. Lors de la campagne 2004, le titre était considéré comme la requête de l'utilisateur ; les autres parties étaient destinées aux évaluateurs (jugements de pertinence).

Le titre est constitué de mots ou groupes de mots recherchés. Ces termes peuvent être préfixés pour indiquer des concepts à favoriser (préfixe +) ou des concepts non souhaités (préfixe -). Les requêtes de type CAS incluent des indications structurelles sous formes de Xpath [CLA 99] précisant la localisation souhaitée pour les concepts et la nature des éléments XML souhaités en résultats.

Exemples de requêtes :

– requête de type CO c'est-à-dire uniquement sur le contenu :

+"query expansion" +"relevance feedback" +web

– requête de type CAS mêlant contenu et structure :

//article[about(.//p,object database)]//p[about(.,version management)]

La tâche ad-hoc regroupe deux types d'évaluation : CO pour le traitement des requêtes uniquement sur le contenu et VCAS (Vague Content And Structure) pour le traitement des requêtes mêlant contenu et structure. L'évaluation VCAS se base sur les requêtes CAS en considérant les contraintes structurelles comme de vagues conditions. Les contraintes ne doivent pas être strictement vérifiées mais correspondent à des préférences pour les résultats recherchés.

4.1.3. Jugements de pertinence

Les jugements de pertinence sont définis suivant deux dimensions : l'exhaustivité et la spécificité. L'exhaustivité tient compte de la présence ou de l'absence de l'information recherchée dans un élément XML. La spécificité traduit le degré avec lequel un élément traite de toute l'information recherchée. Une échelle de quatre valeurs est utilisée pour chacune des dimensions : non -, faiblement -, moyennement -, totalement -. Un élément peut être jugé suivant 10 valeurs possibles sachant que les valeurs "non exhaustif" et "non spécifique" ne peuvent qu'être combinées ensemble. Différentes règles sont introduites lors de la définition des jugements de pertinence afin d'assurer la consistance des jugements.

4.1.4. Mesures

Les évaluations réalisées dans le cadre d'INEX sont basées sur les notions de rappel et de précision en prenant en compte également le degré de pertinence des composants retrouvés. Les évaluations 'officielles' effectuées dans la campagne INEX'2004 [VRI 04] s'appuient sur le calcul de la probabilité P(rel|retr) qu'un document vu par un utilisateur soit pertinent :

$$P(rel|retr)(x) = \frac{x \cdot n}{x \cdot n + esl_{x \cdot n}}$$ où $esl_{x \cdot n}$ correspond au nombre estimé d'éléments non pertinents retrouvés jusqu'à un point de rappel x fixé et n est le nombre total de composants pertinents pour une requête donnée.

Une fonction de quantification est appliquée pour transformer le degré de pertinence suivant les deux dimensions exhaustivité et spécificité vers un degré de pertinence suivant une échelle unique.

$f_{quant}(e,s)$: ES $\rightarrow$ [0,1] où ES est l'ensemble des 10 paires de jugements (exhaustivité,spécificité) possibles.

Différentes fonctions de quantifications sont utilisées dans le but de décrire différents modèles d'utilisateurs. Par exemple, la quantification f_{strict} est utilisée pour évaluer les méthodes de recherche suivant leur capacité à retrouver les composants totalement exhaustifs et totalement spécifiques. D'autres quantifications, par exemple orientées exhaustivité comme f_{e3s321}, ont pour but d'évaluer les méthodes suivant leur capacité à retrouver les éléments les plus exhaustifs pour lesquels la spécificité est variable.

Exemples de fonctions de quantification :

$$f_{strict}(e,s) = \begin{cases} 1 & si\ e = 3\ et\ s = 3 \\ 0 & sinon \end{cases} \qquad f_{e3s321}(e,s) = \begin{cases} 1 & si\ e = 3\ et\ s \in \{1,2,3\} \\ 0 & sinon \end{cases}$$

Un score agrégé (macro-moyenne) des différents scores obtenus pour les fonctions de quantifications donne une mesure d'évaluation globale.

4.2. *Paramétrage des expérimentations*

Trois expérimentations utilisant la méthode de vote ont été soumises à la campagne d'évaluation INEX'2004. Deux expérimentations ont été menées sur les requêtes de type CO et une sur les requêtes de type CAS.

La différence entre les 2 expérimentations sur les requêtes de type CO a résidé dans l'importance donnée, dans la fonction de score, au facteur mesurant le taux de présence des termes de la requête dans le texte. Les 2 expérimentations ont obtenus des résultats très proches avec un léger avantage pour l'expérimentation nommée VTCOTC35xp400sC-515 utilisant la fonction de score suivante :

$$Score(E,T) = Vote(E,T) \cdot \varphi^{(\frac{NT(T,E)}{S(T)})}$$ avec φ=400.

L'expérimentation sur les requêtes de type CAS nommée VTCASC35xp200sC-515PP1 a utilisé la même fonction de score mais avec φ=200.

Une étape préliminaire a consisté à paramétrer la fonction de score. Nous avons mis en place une phase d'apprentissage sur la collection INEX, le jeu de requêtes de type CO de la campagne INEX'2003 et les mesures de précision avec les quantifications 'strict' et 'generalised'. Cet apprentissage avait pour objectif de trouver les paramètres permettant d'obtenir les meilleurs résultats pour les mesures de INEX'2003. La configuration conduisant aux meilleurs résultats est la suivante :

– la couverture fixée à 35% (c'est-à-dire que plus d'un tiers des termes de la requête doit apparaître dans le composant XML pour qu'il soit sélectionné),

– les coefficients pour prendre en compte les préfixes + et – indiquant les concepts à favoriser ou à éviter ont été fixés à +5,0 pour le préfixe +, -5,0 pour le préfixe -, le coefficient pour les concepts non préfixés étant fixé à 1,0,

– le coefficient lié à la propagation du score d'un composant dans la structure d'un document XML a été fixé à 0,1.

Cette configuration a été appliquée pour toutes les expérimentations soumises à INEX'2004. Les paramètres spécifiques au traitement des requêtes de type CAS ont été établis de manière arbitraire. Le coefficient β (cf 3.7) relatif aux contraintes structurelles sur les concepts recherchés a été fixé à 1.0 (c'est-à-dire les voix apportées par un concept pour un élément XML sont doublées lorsque l'élément satisfait la contrainte). Le coefficient γ (cf 3.7) lié aux contraintes structurelles sur le résultat a été fixé à 2.0 (c'est-à-dire le score d'un élément XML qui satisfait la contrainte est doublé). Ces valeurs permettent de privilégier les éléments XML qui vérifient les contraintes sans pour autant éliminer les autres éléments XML. Ceci correspond aux hypothèses choisies pour l'évaluation VCAS de INEX'2004.

4.3. *Résultats*

Les meilleurs résultats que nous avons obtenus sont synthétisés dans les tableaux suivants :

Expérimentation	Score agrégé	Rang	Type de requêtes
VTCOTC35xp400sC-515	0,0783	13/70	CO
VTCASTC35xp200sC-515PP1	0,0784	5/51	CAS

Tableau 1. *Résultats synthétiques des expérimentations*

Les meilleurs résultats obtenus pour les expérimentations sur les requêtes de type CO sont détaillés dans le tableau suivant pour les différentes quantifications :

VTCOTC35xp400sC-515							
Q	strict	generalised	so	s3e321	s3e32	e3s321	e3s32
PM	0,0778	0,0683	0,0559	0,0395	0,0508	0,1456	0,1106
R	18/70	14/70	16/70	22/70	17/70	10/70	11/70

Q : fonction de quantification, PM : Précision moyenne, R : Rang

Tableau 2. *Résultats détaillés de l'expérimentation sur les requêtes de type CO*

Pour les requêtes de type CO, les meilleures mesures ont été obtenues pour les quantifications e3s321 et e3s32 qui se focalisent sur les éléments jugés totalement exhaustifs. Pour la quantification e3s321, la précision moyenne est de 0,1456, plaçant l'expérimentation au $10^{ième}$ rang par rapport à l'ensemble des expérimentations soumises par les différents participants. Pour la quantification e3s32, la précision moyenne est de 0,1106, plaçant l'expérimentation au $11^{ième}$ rang. Globalement, l'expérimentation se classe $13^{ième}$ par rapport aux 70 expérimentations relatives aux requêtes de type CO soumises par l'ensemble des participants.

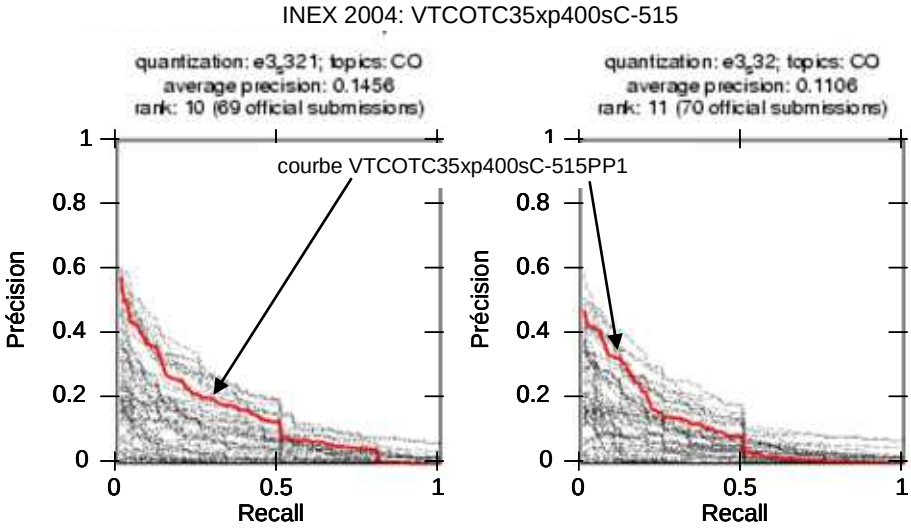

Figure 2. *Courbes précision/rappel pour les requêtes de type CO pour les quantifications 'e3s321' et 'e3s32'*

Les courbes de la figure 2 ci-dessus montrent un positionnement de nos résultats constant dans les 10 premiers pour des taux de rappels jusqu'à 0,5 plus particulièrement pour la quantification e3s321.

Pour les requêtes de type CAS, l'expérimentation a été globalement classée en 5^{ième} place. Les résultats obtenus pour les différentes quantifications sont détaillés dans le tableau suivant :

VTCASTC35xp200sC-515PP1							
Q	strict	generalised	so	s3e321	s3e32	e3s321	e3s32
PM	0,1053	0,0720	0,0554	0,0462	0,0644	0,1162	0,0892
R	5/51	6/51	9/51	12/51	10/51	5/51	5/51

Q : fonction de quantification, PM : Précision moyenne, R : Rang

Tableau 3. *Résultats détaillés de l'expérimentation sur les requêtes de type CAS*

Les meilleures mesures ont été obtenues pour les quantifications ‘strict’, ‘e3s321’ et ‘e3s32’ pour lesquelles l’expérimentation est classée 5ième.

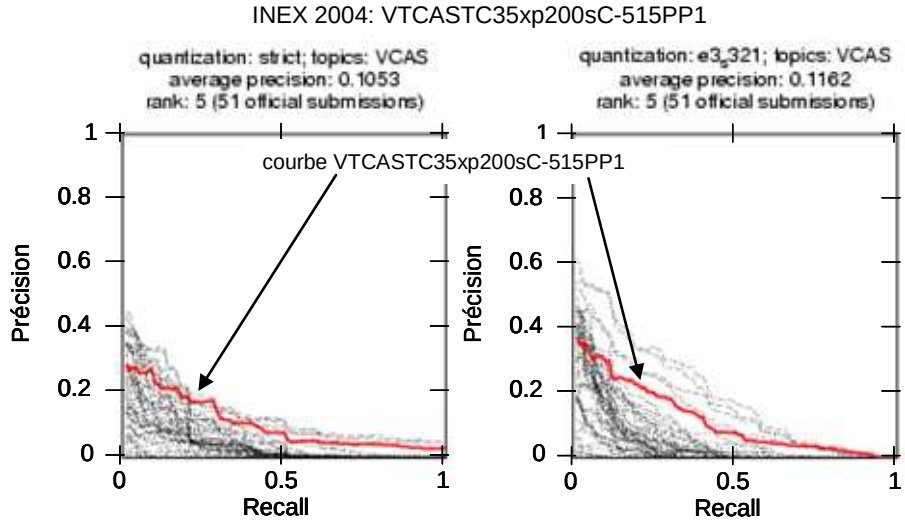

Figure 3. *Courbes précision/rappel pour les requêtes de type CAS pour les quantifications ‘strict’ et ‘e3s321’*

Les courbes précision/rappel de la figure 3 ci-dessus montrent un bon positionnement de nos résultats constamment dans les 5 premiers notamment pour des taux de rappels entre 0,2 et 0,4 avec un passage au rang 1 aux alentours de 0,3 pour la quantification ‘strict’.

5. Discussion et perspectives

Nous avons présenté dans cet article une approche de recherche d’information dans des collections de documents XML. Cette approche est basée sur un principe de vote. L’approche est à la fois simple et performante puisqu’elle se classe 5ième dans le cadre des requêtes mêlant contenu et structure (CAS) de INEX. Les éléments XML retenus sont ceux au sein desquels la requête est la mieux représentée. Notre approche exploite les aspects structurels introduits par l’utilisation du langage XML que ce soit au niveau des documents ou au niveau des requêtes. Au regard des expérimentations réalisées et des résultats obtenus, nous pouvons souligner que :

– les fonctions et paramètres choisis pour la méthode de calcul du score tendent à favoriser l'exhaustivité définie par l'initiative INEX plutôt que la spécificité. En effet, le facteur qui mesure la représentation de la requête dans l'élément XML (c'est-à-dire NT(T,E)/S(T)) a une influence importante sur le résultat de la fonction de score ; ce facteur est liée à l'exhaustivité telle qu'elle est définie dans INEX. Dans le cadre de l'initiative INEX, il serait intéressant de modifier la fonction de score de manière à augmenter le nombre d'éléments sélectionnés ayant une forte spécificité.

– les mesures obtenues en utilisant le jeu de requêtes de type CO de la campagne INEX'2003 étaient globalement supérieures. Ceci suggère que notre méthode de score est plus efficace sur certaines requêtes que sur d'autres. Il sera intéressant d'identifier les catégories de requêtes pour lesquelles la fonction est la plus efficace, les catégories pour lesquelles elle ne l'est pas et de comprendre pourquoi. Des évolutions apportées à la méthode pourraient permettre d'étendre le champ de requêtes sur lesquelles la méthode est efficace ou d'appliquer d'autres méthodes aux autres catégories de requêtes.

– les valeurs des paramètres liés aux indications structurelles ont été fixées de manière arbitraire. Des expérimentations supplémentaires sur les requêtes de type CAS d'INEX nous permettront d'ajuster les valeurs de ces paramètres.

De plus, il serait intéressant d'évaluer le bénéfice apporté à notre méthode par un processus de réinjection de pertinence. Dans un premier temps, il serait possible d'utiliser des informations issues des premiers éléments jugés pertinents par les utilisateurs et de mesurer l'impact sur les résultats. C'est le principe qui a été choisi pour la tâche 'relevance feedback' de la campagne INEX'2004. D'autre part, nous souhaitons étudier un processus de réinjection de pertinence extraite des résultats les mieux classés issus d'une première recherche avec notre système.

Remerciements

Les recherches présentées dans ce papier s'inscrivent dans le cadre du projet intitulé « Aide à la reformulation automatique de requêtes basée sur l'exploration de textes et l'analyse de leurs structures », PAI Alliance N°05768UJ. Les idées exprimées dans ce papier sont cependant personnelles.

6. Bibliographie

[AME 04] Amer-Yahia S., Lakshmanan L., Pandit S. « FleXPath : Flexible Structure and Full-Text Querying for XML », *ACM SIGMOD*, Paris, France, 2004.

[BRA 04] Bray T., Paoli J., Sperberg-McQueen C. M., Maler E., Yergeau Y., Extensible, Markup Language (XML) 1.0. (Third Edition), W3C Recommendation., http://www.w3.org/TR/REC-xml/, 2004.

[BUX 03] Buxton S., Rys M., XQuery and XPath Full-Text Requirements, http://www.w3.org/TR/xquery-full-text-requirements/, 2003.

[CAR 03] Carmel D., Maarek Y. S., Mandelbrod M., Mass Y., Soffer A., « Searching XML documents via XML fragments », *26th international conference SIGIR*, Toronto, Canada, 2003, p. 151-158.

[CLA 99] Clark J., DeRose S., XML Path Language (XPath), W3C Recommendation, http://www.w3.org/TR/xpath.html, 1999.

[CRO 04] Crouch C. J., Apte S., Bapat H., « An Approach to Structured Retrieval Based on the Extended Vector Model », *2nd INEX Workshop*, Dagstuhl, Germany, 2004, p. 89-93.

[FUH 04a] Fuhr N., Großjohann K., « XIRQL: An XML query language based on information retrieval concepts », *ACM TOIS*, vol. 22, Issue 2, 2004, p. 313-356.

[FUH 04b] Fuhr N., Maalik S., Lalmas M., « Overview of the INitiative for the Evaluation of XML Retrieval (INEX) 2003 », *2nd INEX Workshop*, Dagstuhl, Germany, 2004.

[GRA 02] Grabs T., H.-J. Schek H.-J., « Generating Vector Spaces On-the-fly for Flexible XML Retrieval », *XML and Information Retrieval Workshop - SIGIR*, Tampere, 2002.

[HUB 05] Hubert G., « A voting method for XML retrieval », *Proceedings of the 3rd INEX Workshop*, 2005

[HUB 03] Hubert G., Mothe J., Augé J., Englmeier K., « Catégorisation automatique de textes basée sur des hiérarchies de concepts », *19ièmes Journées de Bases de Données Avancées (BDA)*, Lyon, 2003, pp 69-87.

[KAM 04] Kamps, J., de Rijke, M., Sigurbjörnsson, B., « Length normalization in XML retrieval », *27th International Conference on Research and Development in Information Retrieval (SIGIR)*, New York, 2004, p. 80-87

[LI 98] Li Y., « Toward a qualitative search engine », *IEEE Internet Computing*, vol. 2, n°4, 1998, p. 24-29.

[OGI 04] Ogilvie P., Callan J., « Using Language Models for Flat Text Queries in XML Retrieval », *Proceedings of the Second INEX Workshop*, Dagstuhl, Germany, 2004.

[PAU 00] Pauer B., Holger P., Statfinder, Document Package Statfinder, Vers. 1.8, 2000.

[PIW 04] Piwowarski B., Vu H.-T., Gallinari P., « Bayesian Networks and INEX'03 », *Proceedings of the 2nd INEX Workshop*, Dagstuhl, Germany, 2004.

[PON 98] Ponte J. M., Croft W. B., « A Language Modeling Approach to Information Retrieval », *Research and Development in Information Retrieval*, 1998, p. 275-281.

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

[SAL 75] Salton G., Wong A., Yang C. S., « A vector space model for automatic indexing ».- *Communication of the ACM*, vol. 18, Issue 11, 1975, p. 613-620.

[VRI 04] A. P. de Vries, G. Kazai, M. Lalmas, « Evaluation Metrics 2004 », *Pre Proceedings of the 3rd INEX Workshop*, 2004, p. 249-250.