.

Reformulation de Requêtes par Structure en RI dans les Documents XML

Lobna Hlaoua

IRIT-SIG, 118 route de Narbonne, F-31 062 Toulouse Cedex 4, France
{hlaoua@irit.fr}

RÉSUMÉ. *La reformulation de requêtes permet d'enrichir une requête initiale en fonction de jugements de pertinence afin d'exprimer d'avantage les besoins de l'utilisateur. De nouvelles problématiques sont soulevées lorsque la reformulation s'effectue sur des corpus de documents semi-structurés de type XML. Les différentes approches qui ont été développées sont en général basées sur le contenu seul des éléments. Notre contribution consiste à mettre en oeuvre une nouvelle approche permettant d'étendre la requête initiale avec une structure générique et des mots-clés. Cette approche peut être appliquée à la fois sur des requêtes structurées ou non structurées. Nos expérimentations ont suivi le protocole d'évaluation INEX et montrent l'intérêt de notre proposition pour certains types de requêtes.*

ABSTRACT. *Relevance Feedback (RF) is a technique allowing to enrich an initial query according to the user feedback. The goal is to express more precisily the user's needs. Some open issues appear when considering semi-structured documents like XML. Most of the existing RF approaches are applied in the content of elements. We prpose a new approach that is able to extend the initial query by adding a generative structure and keywords. This approach is applied to both structured and un-structured queries. Experiments are carried out with INEX campaign and results show the interest of our method for some query types.*

MOTS-CLÉS : *reformulation, document XML, structure générique, contenu, RI.*

KEYWORDS: *Relevance Feedback, XML document, generative structure, content, IR.*

1. Introduction

La réinjection de pertinence, une des techniques de la reformulation de requêtes, connue aussi sous le nom de Relevance Feedback (RF), permet d'enrichir une requête initiale selon des informations extraites des éléments jugés pertinents par l'utilisateur. Un nouveau challenge est d'appliquer la technique de RF pour la recherche dans des documents semi-structurés de type XML. Lorsqu'il interroge ce type de collection, l'utilisateur, s'il n'a aucune idée de la structure des documents, formule des requêtes à base de simple mots clés (on parle alors des reqêtes CO (Content Only). En résultat, le système renvoie des éléments de différentes granularités (section, paragraphe, référence, titre, etc.). Cependant, il est probable que l'utilisateur ne s'intéresse qu'à quelques types d'éléments. Notre objectif est d'affiner la requête initiale en spécifiant la structure (c.à.d. le type des éléments qui pourraient contenir les informations pertinentes) et en ajoutant des mots clés. La question qui se pose est alors : Comment extraire la structure à partir des éléments pertinents de différents types ?
Dans cet article, nous commençons par présenter l'état de l'art sur la reformulation des requêtes structurées. Nous détaillons notre approche dans la section 3. Nous présentons enfin dans la section 4 les expérimentations effectuées dans le contexte d'INEX et les impacts de différentes approches.

2. Etat de l'art

Dans la littérature de la reformulation des requêtes en recherche d'information dans les documents XML, on distingue deux principales approches :

– une RF orientée contenu basée sur le même principe que la RF en RI classique, mais qui considère des termes extraits des éléments ayant différentes granularités. On y trouve notamment les travaux de Y. Mass [MAS 04]. La RF appliquée sur un modèle vectoriel étendu, consiste à utiliser l'algorithme de Rocchio [ROC 71] sur les requêtes de type CO et a pu donner une amélioration qui n'a pas dépassé les 4% (d'après les évaluation de INEX 2005 [INE05a]). Crouch [CRO 04] a également appliqué l'algorithme de Rocchio [ROC 71] sur un modèle vectoriel basé sur la propagation d'exhaustivité et de spécificité. Aucune amélioration n'est cependant observée.

– une RF qui consiste à enrichir la requête en rajoutant des contraintes structurelles, nous citons Mihajlovic et al [MIH 04]. La technique de RF consiste à extraire le nom d'un document, auquel un élément pertinent a plus de chance d'appartenir ainsi que le type d'élément. L'amélioration est d'ordre 5%. Schenkel et Theobald [SCH 05] ont proposé de structurer les requêtes CO en spécifiant pour chaque mot clé la structure dans laquelle il a plus de chance de se trouver. L'amélioration est intéressante.

3. Contribution

Nous proposons deux approches : une approche orientée structure qui consiste à trouver à partir des éléments jugés la structure générique qu'on l'injecte dans la

requête initiale et une approche orientée structure et contenu qui combine la première avec celle orientée contenu. Cette dernière consiste à extraire les mots clés à rajouter à la requête en appliquant l'algorithme de Rocchio [ROC 71].

3.1. *Extraction de la structure générique*

On appelle structure générique une structure qui peut être commune à un grand nombre d'éléments pertinents. Notre algorithme consiste à apparier la structure de chaque élément pertinent avec le reste des structures des éléments jugés pertinents. En résultat, on obtient un ensemble des Structures Communes appelé SC.
Soient E^p l'ensemble des éléments pertinents et e_i^p un élément $\in E^p$. e_i^p est caractérisé par un chemin simplifié p_i (exemple : */article/bdy/section*) et un poids w_i initialisé par une constante au début de l'algorthme. Pour chaque élément $e_i^p \in E^p$, et pour chaque $e_j^p \in E^p - \{e_i^p\}$, nous appliquons la fonction SCA *(Smallest Common Acestor)* qui permet d'extraire le chemin du plus petit ancêtre commun de e_i^p et e_j^p tout en calculant son poids. Le chemin sera par la suite ajouté à l'ensemble des Structures communes SC. L'algorithme de la fonction SCA est le suivant :

$SCA(e_i^p, e_j^p)$
Début
si $p_i.first = p_j.first$, alors
  si $p_i.last = p_j.last$, alors si $\exists e_p(p_p, w_p) \in CS / p_p = p_i$
                                  alors $w_p \leftarrow w_p + w_j$
                                  sinon $w_i \leftarrow w_i + w_j$
                                  $CS \leftarrow sp_i$
  sinon
  si $head(p_j) \neq null$, alors $p'_j \leftarrow head(p_j)$
                                  $w'_j \leftarrow w_j/2$
                                  $SCA(e_i^p(p_i, w_i), e_j'^p(p'_j, w'_j))$
                                  sinon $SCA(e_j^p, e_i^p)$
Fin

avec $p.first$ et $p.last$ respectivement la première et dernière balise du chemin p, et $head(p)$ une fonction permettant de réduire le chemin p en lui attribuant celui de l'ancêtre direct, (c.à.d. supprimant la dernière balise de la structure). Par exemple, $head(/article/bdy/section) = /article/bdy$. La structure ayant le plus grand score, sera ensuite utilisée sous une forme simplifiée qui correspond au $p.last$.

3.2. *Exemple*

On considère pour une requête donnée, trois éléments jugés pertinents e_k^p, e_l^p et e_m^p auxquels correspondent les structures : S_k, S_l et S_m (Voir Figure 1 ; nous considérons dans la structure que les noms des balises). Des poids w_k, w_l et w_m sont affectés aux

structures, avec w_k=w_l=w_m=constante=1. Soit l'ensemble SC initialement vide dans lequel on rajoutera les structures génériques.
L'extraction de la structure générique se résume en trois étapes (Voir Figure 1) :

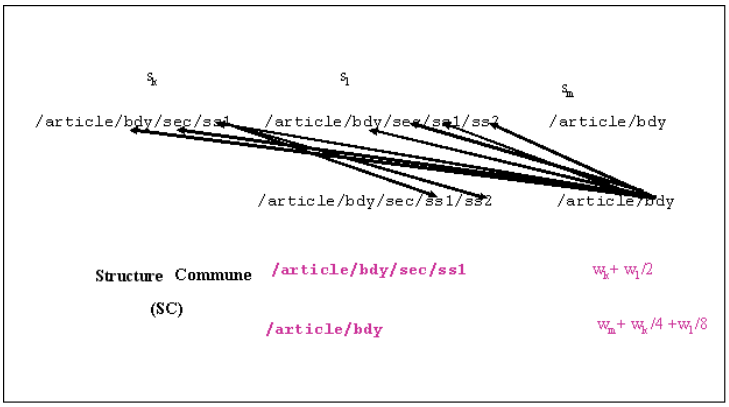

Figure 1. *Recherche d'une structure générique*

1- Appariement de S_k et S_l, avec en résultat l'ajout de S_k dans l'ensemble SC avec un score $w_k + w_l/2$.
2- Appariement de S_k et S_m,avec en résultat l'ajout deS_m dans l'ensemble SC avec un score $w_m + w_k/2^2$.
3- Appariement de S_l et S_m, avec l'incrémentation du score de S_m de $w_l/2^3$.
S_k(*/article/bdy/sec/ss1*) a le score le plus élevé et est sélectionnée comme la structure la plus générique. Dans ce qui suit, les requêtes sont formulées selon le langage de requête NEXI (Narrowed Extended XPath) [TRO 04].
Soit les deux requêtes initiales Q1 et Q2 respectivement de type CO et CAS.
Q1 : *"recherche d'information"* et Q2 : *"article(about(.,"recherche d'information")"*.
Si on reprenons le résultat de l'exemple précédent, les nouvelles requêtes sont respectivement : *Q1' "sec(about(.,"recherche d'information")"*. et *Q2'"article(about(.,"recherche d'information") OR sec(about(.,"recherche d'information")"*.
Dans le cas de reformulation orientée structure et contenu, la requête finale comportera le top *k* des termes selon l'algorithme de Rocchio [ROC 71], qui seront rajoutés aux termes originaux de la requête initiale. L'ensemble de termes sera conditionné par la structure extraite comme la structure la plus générique.

4. Expérimentations

Nous évaluons nos expérimentations grâce la compagne d'évaluation INEX (INitiative for the Evaluation of XML Retrieval) [INE05b]. Cette dernière propose une collection composée de plus de 17000 documents provenant de 21 revues IEEE Computer Society parues de 1995 à 2004. Les jugements de pertinence pour chaque requête

sont effectués par les différents participants. La pertinence d'un élément est évaluée selon deux dimensions : l'exhaustivité et la spécificité. Nous utilisons pour les évaluations les mesures proposées par G. Kazai et M. Lalmas [KAZ 05] : la moyenne des Gains Cumulés Normalisés (*MAnxCG[i]*) par rapport au gain idéal (i est le rang des éléments cumulés) et la moyenne non interpolée d'effort-précision (*MAep*).
Pour comparer, on a défini AR *(Amélioration Relative)* calculée comme suit :

$$AR = (MAP(RF\_run) - MAP(base\_run))/MAP(base\_run) \qquad [1]$$

où MAP dans l'equation 1 est soit MAnxCG@1500, soit MAep, *RF-run* est le résultat obtenu avec reformulation, et *base-run* est le résultat de base.
Les éléments jugés pertinents correspondent aux éléments très exhaustifs. Nous avons testé les deux approches (orientée structure et orientée structure et contenu) sur les différents types de requêtes : CO, CO+S (requête à base des requêtes CO avec une structure) et VVCAS (requête de type CAS dont la pertinence ne dépend pas des contraintes structurelles). Pour chaque mesure, on utilise deux fonctions d'agrégation : stricte et généralisée qui expriment les deux dimensions de pertinence (exhaustivité et spécificité). Dans ce qui suit, nous représentons les valeurs des améliorations relatives aux résulats de base obtenus par le système XFIRM [SAU 04]. L'extension *cs* (resp. *c10s*) désigne les résultats obtenus en appliquant la RF par structure (resp. RF combinée en ajoutant les 10 premiers termes par l'algorithme de Rocchio).
Nous remarquons dans le tableau 1 que la structure a un impact très positif dans le cas des requêtes COS (AR gen(MAep =1,579, MAnxCG@1500=0.36)) ce qui montre que l'injection de la structure générique est efficace pour affiner la requête initiale. Ceci n'est pas prouvé dans le cas des requêtes de type CO et VVCAS. L'impact négatif peut être expliqué par le fait que l'utilisateur n'a pas de préférence au niveau des structures.

	MAep strict	MAep gen	MAnxCG@1500 strict	MAnxCG@1500 gen
AR-VVCAS-RF-cs	-0.0063	-0.1810	-0.4200	-0.2914
AR-VVCAS-RF-c10s	0.0985	0.0451	0.01031	0.0319
AR-COS-RF-cs	1.5790	1.0821	0.3630	0.8394
AR-COS-RF-c10s	0.4588	0.5537	-0.1067	0.4773
AR-CO-RF-cs	-0.5391	-0.3397	-0.3540	-0.3685
AR-CO-RF-c10s	-0.7736	-0.7451	-0.8441	-0.6684

Tableau 1. *Comparaison des deux approches de reformulation*

La reformulation combinée s'avère intéressante dans des requêtes VVCAS et COS mais pas dans le cas des requêtes CO. En effet d'après le tableau 1, la valeur de AR est de l'ordre 40% pour les requêtes COS et environ 7% pour les requêtes VVCAS ce qui confirme à la fois les résultats de la reformulation des requête en RI classique et le fait que la structure générique permet d'affiner une requête initiale. Elle est négative pour les requêtes CO ce qui nous pousse à d'autres méthodes d'extraction et/ou d'autres façons d'injection de la structure et des mots clés.

5. Conclusion

Nous avons présenté une nouvelle approche de RF basée sur l'extraction d'une structure générique, que nous avons combinée avec une approche orientée contenu basée sur l'algorithme de Rocchio. Nous avons comparé les deux approches sur des requêtes de type CO, COS et VVCAS selon le protocole d'évaluation INEX. En général, les résultats de la reformulation des requêtes COS montrent une bonne amélioration. Dans le cadre de notre participation à la tâche RF de INEX 2005, nous sommes classés les premiers. Les résultats de la reformulation des requêtes VVCAS sont meilleurs quand la reformulation combinée est utilisée. La reformulation des requêtes de type CO n'apporte aucune amélioration. Ceci peut être expliqué par le fait qu'il n'y a pas de structure spécifique qui peut satisfaire les besoins de l'utilisateur, ce qui nous amène à penser à étendre le champs des contraintes structurelles (considérer plus d'une structure générique). Un de nos futurs travaux est d'améliorer l'extension du contenu en tenant compte de la sémantique des éléments (titre, section, référence, etc.) à partir desquels on extrait les mots clés les plus significatifs.

6. Bibliographie

[CRO 04] CROUCH C., MAHAJAN A., BELLAMKONDA A., « Flexible XML Retrieval Based on the Vector Space Model », *INEX 2004 Workshop Proceedings*, Germany, December 2004, p. 292,302.

[INE05a] *INitiative for the Evaluation of XML Retrieval*, disponible sur http ://inex.is.informatik.uni-duisburg.de :2005/tracks/rel/, 2005.

[INE05b] *INitiative for the Evaluation of XML Retrieval*, disponible sur http ://inex.is.informatik.uni-duisburg.de :2005/, 2005.

[KAZ 05] KAZAI G., LALMAS M., « INEX 2005 Evaluation Metrics », *INEX 2005 Workshop Pre-Proceedings*, Germany, November 2005.

[MAS 04] MASS Y., MANDELBROD M., « Relevance Feedback for XML Retrieval », *INEX 2004 Workshop Proceedings*, Germany, December 2004, p. 303,310.

[MIH 04] MIHAJLOVIC V., RAMIREZ G., DE VRIES A., HIEMSTRA D., BLOK H., « TIJAH at INEX 2004 Modeling Phrases and Relevance Feedback », *INEX 2004 Workshop Proceedings*, Germany, December 2004, p. 276,291.

[ROC 71] ROCCHIO J., « Relevance feedback in information retrieval », *The SMART retrieval system-experiments in automatic document processing*, Prentice Hall Inc, 1971, p. 313,323.

[SAU 04] SAUVAGNAT K., « XFIRM, un modèle flexible de recherche d'information pour le stockage et l'indexation de documents XML », *Actes de CORIA'04*, Toulouse, France, Mars 2004, p. 121,142.

[SCH 05] SCHENKEL R., THEOBALD M., « Relevance Feedback for Structural Query Expansion », *INEX 2005 Workshop Pre-Proceedings*, Germany, November 2005, p. 260,272.

[TRO 04] TROTMAN A., SIGURBJÖRNSSON B., « Narrowed Extended XPath I(NEXI) », *INEX 2004 Workshop Proceedings*, Germany, December 2004, p. 16,40.