Accéder l'information dans une collection de documents XML :

Requêtes et Recherche plein texte

Anne-Marie Vercoustre,
Directeur de Recherches,
INRIA, Rocquencourt,
Anne-Marie.Vercoustre@inria.fr

Introduction

XML a été développé pour permettre la description uniforme de données et de documents et permettre un accès à l'information plus précis que ce qui peut se faire avec HTML. En effet, HTML est utilisé d'avantage pour présenter l'information que pour la décrire, rendant l'accès à l'information peu précise pour les utilisateurs et pour les applications. XML peut être utilisé pour échanger des données entre systèmes d'information, mais aussi, plus traditionnellement comme en SGML, pour représenter la structure logique de n'importe quel type de documents contenant du texte libre, comme des articles scientifiques, des rapports ou des pages Web. Alors que de larges collections de documents XML commencent à se constituer, se développent dans le même temps des langages et des systèmes pour chercher l'information dans ces collections. L'espoir est que la présence des tags permette à l'utilisateur de spécifier plus précisément ce qu'il cherche, et au systèmes de retourner des documents ou parties de documents qui répondent mieux aux besoins de l'utilisateur.

Données XML versus documents XML

Il découle de ce qui précède qu'un langage de requête pour XML doit couvrir deux aspects:

- L'aspect centré document doit permettre d'interroger une collection de documents d'une façon flexible prenant en compte la structure des documents, mais aussi son contenu. On doit pouvoir faire une recherche libre à la Goggle, mais aussi permettre des recherches plus précises utilisant la structure, par exemple demander les articles publiés par un auteur donné en 2000. L'important, dans ce contexte, est de classer les réponses par ordre de préférence, c'est à dire en commençant par ceux qui sont supposés répondre le mieux à la question.
- L'aspect centré données concerne les applications qui échange des données dans un format spécifiques dépendant de l'application. Typiquement il s'agit d'applications de commerce électronique qui utilisent des données telles que bons de commande, factures, catalogues de produits, etc. Un langage supportant ces aspects offrira non seulement des opérateurs pour sélectionner l'information, mais aussi des opérateurs pour regrouper ou restructurer les résultats dans un nouveau document. Ces opérateurs sont très similaires à ceux trouvés las les langages pour les bases de données tels que SQL.

Xquery, le langage de requêtes proposé par le W3C comme standard prend avantage à la fois du contenu et de la structure spécifiée dans la collection pour retourner des éléments

XML qui satisfassent *strictement* les conditions logiques spécifiées dans la requête, dans un ordre quelconque[1]. Par exemple, si l'on veut rechercher des figures décrivant l'architecture Corba, on peut écrire une requête précisant que cette chaîne de caractères doit se trouver dans la légende de la figure, mais il n'est pas possible d'être un peu plus vague, par exemple demander les figures au sujet de "Corba Object Broker Architecture interoperability communication protocols" et laisser le système retourner les figures jugées les plus proches de la requête. Au minimum, il manque dans XQuery un opérateur de similarité.

L'enjeu est de combiner la flexibilité des moteurs de recherche plein texte, tout en tirant le maximum davantage de la connaissance de la structure comme dans les bases de données.

Extension de Xquery au plein texte

Un groupe de travail du W3C sur Xquery a récemment publié les conditions que doit remplir un langage de requête XML pour la recherche plein texte [1]. L'objectif est de définir un langage, basé sur Xquery et Xpath, incorporant la recherche plein texte et classement des résultats par pertinence. Les conditions suivantes sont identifiées:

- Possibilité de requête sur le texte à l'intérieur des tags: mot, phrases, forme normalisée de mots (stemming), stopwords, proximité, opérateurs booléens, etc.
- Recherche sur des structures arbitraires, combinaison de prédicats sur différentes parties de la structure du document; possibilité d'ignorer les sous structures; une requête peut retourner un nœud arbitraire (élément XML)
- Le classement des réponses est rendu possible par une fonction "score"qui mesure la similarité entre un fragment de document et la requête (ou partie de requête) et renvoie une valeur entre 0 et 1. Le choix de cette fonction est laissé au soin du concepteur de chaque système, et l'utilisateur d'un système doit pouvoir, en principe, remplacer cette fonction par une autre.

La proposition, bien qu'allant dans la bonne direction, souffre encore d'imprécision et de limitations dont certaines sont identifiées dans [6]. En particulier, le langage ne permet pas de laisser indéfini, et à la charge du système, le choix de la granularité du ou des éléments à retourner à l'utilisateur. Or ceci est justement un des principaux objectifs en recherche d'information: retourner à l'utilisateur l'information dont il a besoin, et sous la forme la plus appropriée. Dans certains cas il s'agira d'une figure ou d'un paragraphe, dans d'autre d'une section ou de tout le document, mais il n'est pas toujours possible de le déterminer au moment de la requête.

L'initiative INEX

INEX (Initiative for the Evaluation of XML Retrieval) est un effort international organisé à l'initiative de réseau d'excellence DELOS pour les Bibliothèques Electroniques. INEX, après seulement trois ans d'existence, comprend une cinquantaine de groupes participant à l'évaluation de différentes approches et systèmes de recherche d'information pour XML [2].
INEX utilise une collection de 12107 articles de IEEE, publiés entre 1995 et 2002 et transcrits en XML, totalisant 494 mégaoctets. Chaque année, depuis 2002, un nouvel ensemble de 60

[1] On peut classer les réponses, par exemple par ordre alphabétique, mais pas en fonction de leur plus ou moins adéquation avec la requête, puisque toutes les réponses vérifient strictement et de façon égale la requête.

objets de recherche (topics) est défini par les participants et utilisé pour l'évaluation. Pour plus de détails sur la méthode utilisée pour l'évaluation, on se reportera aux papiers cités en référence. Il est plus instructif d'analyser ici les deux types de *topics* utilisés, d'une part ceux portant sur le contenu seulement (CO topics), et ceux portant à la fois sur le contenu et la structure (CAS topics).

Dans le premier cas, une requête envoyée au système de recherche d'un participant INEX est assez similaire à une requête pour un moteur de recherche standard. Mais le résultat retourné, au lieu d'être une liste ordonnée de documents, est une liste de fragments de documents (éléments XML), dont la granularité, pas nécessairement uniforme, est déterminée par le système en question. L'objectif d' INEX est ici d'évaluer dans quelle mesure l'utilisation de la structure XML permet de retourner à l'utilisateur des réponses pertinentes plus fines que le document complet.

Les *CAS topics*, quant à eux, définissent non seulement l'information recherchée, mais d'éventuelles contraintes sur le document qui la contient, et/ou sur le type d'élément à retourner à l'utilisateur. Les objectifs d'évaluation portent ici 1) sur la précision dans le classement des réponses par rapport à leur contenu, 2) de l'adéquation de la granularité des réponse par rapport au besoin de l'utilisateur, 3) de la possibilité de retourner des éléments sémantiquement *voisins* de ceux demandés (une image au lieu d'une figure, par exemple).

Les objectifs 2) et 3) sont d'autant plus importants que les utilisateurs ont peu de chance de connaître la DTD décrivant les documents. Il faut donc que les contraintes portant sur la structure puissent être relâchées, plutôt que de ne renvoyer aucun document à l'utilisateur.

Le but d'INEX est à la fois de développer une méthodologie d'évaluation, des outils d'évaluations (collections, ensemble de requêtes), et de tester des approches et systèmes particuliers de recherche pour XML.

Approches pour les systèmes de recherche d'information

Les approches proposées par les participants d'INEX peuvent être classées en trois groupes:

- Les approches qui étendent les modèles de recherche plein texte [7].
- Les approches orientées bases de données qui étendent les systèmes classiques pour permettre de traiter les documents XML, ou utilisent des systèmes natifs XML qui généralement incorporent les langages standard XML, tels que Xpath, XSL et Xquery. Parmi les systèmes natifs, on peut citer eXist [3] ou Xylème [8].
- Les approches hybrides, qui cherchent à combiner le meilleur des deux approches, par exemple en utilisant un system plein texte pour faire une présélection (ordonnée) des meilleurs documents, et un system natif XML pour raffiner les réponses, et éventuellement extraire les éléments spécifiques recherchés. Des heuristiques peuvent aussi être utilises pour déterminer la meilleure granularité des réponses [4],[5].

Quelques expériences réalisées en 2003 [5], on montré 1) que l'approche plein texte donne de meilleurs résultats que l'utilisation d'une base XML native, pour les *topics* pour lesquels les réponses sont les documents entiers (CO et certains Cas topics), 2) que notre approche hybride améliore les résultats très nettement dans le cas des requêtes sur le contenu (CO), et moins nettement pour celles portant aussi sur la structure (CAS).

Conclusion

Bien qu'il faille prendre les résultats de ces évaluations avec prudence (il s'agit d'un collection particulière, de requêtes particulières et de méthodes dévaluation en cours d'élaboration), il est intéressant de noter qu'un système natif XML n'est pas nécessairement l'outil le mieux adapté à la recherche d'information orientée documents XML[2].

Les extensions proposées pour Xquery vont permettre plus de flexibilité pour la recherche plein texte, ainsi que le classement des réponses, et donc certainement améliorer les performances des systèmes natifs XML pour cette recherche plein texte.

Toutefois il reste encore beaucoup de travail pour implémenter des systèmes réellement adaptés à l'utilisateur, qui le plus souvent ne connaîtra pas la DTD des documents, et capables de retourner les résultats avec le niveau de granularité approprié aux besoins.

Références

[1] S.Buxton and M.Rys (editors), "*Xquery and Xpath Full-Text requirements*", W3C Working Draft, May 2003. http://www.w3.org/TR/xquery-full-text-requirements/

[2] N. Fuhr, S. Malik and M. Lalmas, "'Overview of the Initiative for the Evaluation of XML retrieval (INEX) 2003"'. In N. Fuhr, M. Lalmas and S. Malik (Eds.) Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Dagstuhl, Germany, December 15-17, 2003. 2004. http://inex.is.informatik.uni-duisburg.de:2003/

[3] W. Meier, "'eXist: An Open Source Native XML Database. In A. B. Chaudri, M. Jeckle, E. Rahm and R. Unland (Eds.) Web, Web-Services, and Database Systems. NODe 2002 Web- and Database-Related Workshops, Erfurt, Germany, October 7-10, 2002. Lecture Notes in Computer Science 2593 Springer, 2003. http://exist.sourceforge.net/

[4] J. Pehcevski, J. Thom and A-M. Vercoustre, "RMIT INEX Experiments: XML Retrieval using Lucy/eXist". In N. Fuhr, M. Lalmas and S. Malik (Eds.) Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Dagstuhl, Germany, December 15-17, 2003. 2004. http://inex.is.informatik.uni-duisburg.de:2003/proceedings.pdf

[5] J. Pehcevski, J. Thom and A-M. Vercoustre , "Enhancing Content-And-Structure Information Retrieval using a Native XML Database", to appear in *Proceedings of the First Twente Data Management Workshop (TDM'04*), The Nederlands, 21 June 2004.

[6] J. Pehcevski, J. Thom and A-M. Vercoustre**, "**XML-search Query Language: Needs and Requirements", *in Proceedings of the ninth Australian World Wide Conference (Ausweb2003*), Gold Coast, Australia, 2003. http://ausweb.scu.edu.au/aw03/papers/thom/paper.html

[7] A-M Vercoustre, J. A. Thom, A. Krumpholz, I. Mathieson, P. Wilkins, M. Wu, N. Craswell and D. Hawking, "'CSIRO INEX experiments: XML Search using PADRE. In N. Fuhr, N. Goevert, G. Kazai and M. Lalmas (Eds.) *Proceedings of the 1st Workshop of the INitiative for the Evaluation of XML Retrieval (INEX) 2002*, Dagstuhl, Germany, December 8-11, 2002. ERCIM, 2003. http://qmir.dcs.qmul.ac.uk/inex/Workshop.html

[8] Vincent Aguilera, Sophie Cluet, Pierangelo Veltri, Dan Vodislav, Fanny Wattez, Querying XML Documents in Xyleme, in *Proceedings of the ACM SIGIR 2000 workshop on XML and Information* Retrieval, Athens, 2000.

[2] Il faut aussi noter qu' INEX ne s'intéresse pas pour l'instant aux performances des systèmes évalués.