TUNING SEARCH ENGINE TO FIT XML RETRIEVAL SCENARIO

Gilles Hubert[1,2], Josiane Mothe[1,2], Kurt Englmeier [3]
(1) IRIT/SIG-EVI, Université Paul Sabatier, 118 route de Narbonne F-31062 Toulouse cedex 9
(2) ERT34, Institut Universitaire de Formation des Maîtres, 56 avenue de l'URSS, F-31078 Toulouse cedex
(3) LemonLabs GmbH, Germany

hubert@irit.fr, mothe@irit.fr, KurtEnglmeier@computer.org

Keywords: Information Retrieval, XML, configurable engine, retrieval scenarios.

Abstract: XML usage is growing to describe documents. Consequently, systems to search in XML collections are necessary. Various proposals of systems for XML retrieval intend to provide solutions to handle XML documents. This paper describes an XML approach based on direct contribution of the components constituting an information need. The search engine is largely configurable in order to be adapted to different context of search. Beyond being globally adapted to a collection of documents an important objective is to define a search engine that can be adapted to different retrieval scenarios and to identify how to adapt it. This paper presents first experiments on INEX testbeds that show how the engine can be adapted to better respond to different retrieval scenarios.

1 INTRODUCTION

Documents particularly scientific publications are more and more described using XML language (Bray et al., 2004). Consequently, systems handling documents tend to move to XML management. Thus, in the information retrieval (IR) domain many systems evolve to exploit the structure of documents and combine textual search and structural search. IR systems generally manage whole documents (i.e. indexing units and retrieval units are whole documents) even if works on passage retrieval (e.g. paragraphs, phrases) have been undertaken.

With structured documents, new possibilities of search are provided to users since different granularities of elements can be managed. Users can indicate structural constraints on the elements constituting the response. These new possibilities multiply the kinds of search and introduce numerous retrieval scenarios. In addition to have systems adapted to XML collections it would be interesting to have systems adapted and even adaptable to different retrieval scenarios. Moreover it would be interesting to know how a system can be adapted to respond as well as possible to a given retrieval scenario. In this context, INEX (Fuhr et al., 2003) is an initiative which aims at providing means for the evaluation of XML retrieval systems. INEX defines different retrieval scenarios.

This paper presents a configurable search engine for XML retrieval that aims at being adapted to various contexts of search notably different retrieval scenarios. These possibilities of engine adaptation have been experimented using INEX testbeds. The paper is organized as follows. Section 2 summarizes the principal approaches proposed in the domain of XML retrieval and situates our search engine. Section 3 describes the principles of the retrieval engine. The engine adaptability is introduced in section 4. Section 5 presents and discusses the experiments realized within the INEX framework to study the adaptation capabilities of our search engine. Section 6 is the conclusions.

2 RELATED WORK

Many approaches related to XML IR are derived from works on database query languages introducing vague correspondence for predicates on content or structure. For example, XIRQL (Fuhr and Großjohann, 2004) introduces a ranking principle of document result based on a probabilistic computation from weighting of query indexing

terms and document indexing terms. XIRQL also introduces data types to which can be associated vague predicates and structural vagueness. Other works propose structural relaxation of XPath (Clark and DeRose, 1999) queries such as FlexPath (Amer-Yahia et al., 2004). The underlying principle is to built a set of queries where structural constraints are enlarged and then to mix and reorder the results.

In the IR domain, the vector space model (Salton et al., 1975) and similarity measures such as cosine have inspired approaches for XML documents. For example, (Carmel et al., 2003) propose query decomposition into XML fragments. Pairs (term, context) constitute indexing units. A context is an XPath of an XML element. The cosine measure is extended to combine context similarity and term similarity. (Crouch et al., 2003) propose an approach where a vector representing a document is a set of sub-vectors for different classes of information. The similarity between extended vectors is defined as a linear combination of similarities between corresponding sub-vectors.

Language modelling (Ponte and Croft, 1998) has been also used for XML documents. For example, (Sigurbjörnsson et al., 2005) propose a multinomial language model with smoothing using document indexes at different levels (article, element). A length normalization of XML elements is proposed to avoid the bias introduced by smoothing in the results. A hierarchical language model is also proposed by (Ogilvie and Callan, 2003) where XML documents are represented by trees. Each XML node (tag in the document) is estimated using a linear interpolation of its content, its children models and its parent model.

Otherwise, (Piwowarski et al., 2003) use bayesian networks to describe documents. Scores of elements are computed recursively through the network from the biggest root element of the network (corpus) to the leaves (smallest XML components).

Few approaches deal with adaptation possibilities. Few have studied their capacities to be adapted to different search contexts. (Liu et al., 2004) present an approach permitting a configurable indexing and ranking for XML information retrieval. This approach gave significant results with INEX 2003 testbed.

Our approach belongs to vector model approaches. However, our underlying principle is directly based on addition of individual contributions of elements defining a query.

3 SEARCH ENGINE PRINCIPLES

Our objective is to propose a search engine responding to information needs an end user can express when handling XML documents. Information need can be “traditional” regarding the textual content such as looking for elements dealing about a list of concepts. It can also be more precise including XML structure such as looking for sections dealing about concepts within articles dealing about other concepts. Like commonly in the IR domain, the objective is to define a search engine which permits to respond as well as possible to the information need expressed by a user without necessarily satisfying all the indications defining this need. Our approach is not based on a “usual” similarity calculus. It is based on a combination of contributions of query elements and mainly on the contribution brought by every term constituting the query. The elements getting the highest scores constitute the result given to the user in response to the query. Other works such as (Geva, 2005) in the context of XML collections arrived to solutions based on similar guiding principles however differing from heuristics used, score propagation, and XML structure handling.

Moreover, the objective is to propose a search engine that takes into account the elements defining an information need in an incremental manner and with a variable strength given to each element. The search engine is largely configurable to try to respond to various contexts of retrieval (e.g. different kinds of collections, different retrieval scenarios). For example, the engine has been used for automatic categorisation (Augé et al., 2003).

3.1 Scoring principles

Document indexing is performed automatically at the element level (i.e. leaf nodes of documents with textual content) to provide the most possibilities to scoring principles between documents and queries.
Documents are represented as sets of triplets (id, term, occ) where id identifies an XML element (including the document identifier and the XPath locating the node containing the term in the document) and occ is the number of occurrences of the term in the textual content of the node. Term extraction involves notably stop word removal and optional processes such as stemming. A similar extraction process is performed on queries.
To define our scoring method, we tried to determine different characteristics which intervene in the decision to consider an element as relevant. The characteristics we identified are the following:

- the importance of a term in the element,
- the term importance to represent the information need,
- the global query importance in the element.

These characteristics have been modelled and combined via three factors to define the function that estimates the relevance of an element as follows:

$$Score(T,E) = \left(\sum_i f(t_i,E) \cdot g(t_i,T) \right) \cdot p(T,E) \quad (1)$$

Where T is the topic, t_i is a term representing the topic T, and E is an XML element.

Firstly, the function adds the contributions of the query concepts. This principle allows giving relevance to elements dealing about either only one concept or several concepts. The sum promotes elements containing several concepts. However, depending on the different chosen functions, dealing strongly about one concept can be evaluated higher than dealing lightly about many concepts. Secondly, the function estimates globally the relevance of an element according to a query.

For previous works related to automatic categorisation (Augé et al., 2003) functions *f*, *g*, *h* were defined. For XML retrieval the functions *f*, *g*, *h* have been adapted according to the context:

- The function $f(t_i,E)$ that measures the importance of a term in an XML element is based on the number of occurrences of the term in the element.
- The function $g(t_i,E)$ that measures the importance of a term in a topic representation is based on the frequency of the term in the topic. The frequency is moderated by the number of XML elements containing the term.
- The function $p(T,E)$ that measures the global presence of a topic in an XML element is based on the number of terms describing the topic and that appear in the XML element.

The scoring function is then defined as follows:

$$Score(T,E) = \underbrace{\left(\sum_i \underbrace{n_i}_{f} \cdot \underbrace{\frac{f_{i,T}}{e_i}}_{g} \right)}_{} \cdot \underbrace{\varphi^{\frac{n_{T,E}}{n_T}}}_{p} \quad (2)$$

Where

n_i is the number of occurrences of the term t_i in the element E

$f_{i,T}$ is the frequency of the term t_i in the query T

e_i is the number of elements containing the term t_i

φ is a real ($\varphi > 0.0$)

$n_{T,E}$ is the number of common terms between the topic T and the element E

n_T is the number of distinct terms in the topic T

When φ is set to 1.0 the function *p* has no effect on the final score. The influence of the function *p* on the final score increases with the value of φ. Using a function power intends to clearly distinguish the elements containing a lot of terms describing the topic and the elements containing few query terms.

The scoring method is completed by principles that handle term preferences (indications of term interest or disinterest), coverage (representation minimum of the query in the element to select it), and structural preferences (preferences on content localization and preferences on structure of target elements). Since this paper does not focus on these principles they are not detailed. Details can be found in (Hubert, 2005).

3.2 Score propagation

The XML hierarchical structure has to be treated. The hypothesis on which is based our search engine is that an element containing a component estimated as relevant is also relevant. Our approach takes into account this hypothesis propagating the score of an element to the elements it composes. The score propagated to the composed elements can be decreased applying a reducing factor.

$$\forall\, E_a \text{ ancestor of } E$$

$$Score(E_a,T) = Score(E_a,T) + \alpha^{\frac{d(E_a,E)}{d(E_R,E)}} \cdot Score(E,T)$$

Where α is a real coefficient ($0.0 \leq \alpha \leq 1.0$)

E, E_a, and E_R are XML elements

$d(E_a,E)$ is the distance between E_a and E in the path associated to E (e.g. in the path /article/bdy/s/ss1/p, d(bdy,p)=3)

$d(E_R,E)$ is the distance between the root E_R and E in the path associated to E

This process tends to consider a composed element less relevant than the element it is composed of. However, an element composed of several relevant elements can obtain a score greater than one of its components. The coefficient α allows varying the score propagation of an element in its ancestors from no propagation (α = 0.0) to total propagation (α = 1.0).

4 ENGINE ADAPTABILITY

The proposed search engine is largely configurable. In addition to changing the functions *f, g, p* (cf 3.2) in the scoring function, the value of numerous coefficients can be changed providing possibilities to adapt the search engine. This paper focuses on the query presence coefficient φ (cf. 3.2), and the score propagation coefficient α (cf. 3.3). These coefficients can be related to different aspects of XML retrieval. This tends to permit to adapt the retrieval process to XML retrieval scenarios that can be related to different criteria such as:

- Size of target elements: a user may be interested firstly in short elements or on the contrary he may prefer bigger elements. The size of target elements can be influenced varying the propagation coefficient.
- Query coverage: a user may prefer having few elements in the results but dealing about a major part of the query. On the contrary, he may prefer having a longer list of resulting elements even dealing about less concepts. The query coverage is indirectly integrated in the scoring method. The more concepts appear in an element the greater is the score. This is due to the sum of concept contributions and to the query presence factor.
- Element focus: a user can be more interested in elements focusing on one concept of the query than elements dealing lightly about all the concepts and inversely. The element focus can vary modifying the propagation.

5 EXPERIMENTS

Training done in the context of INEX in 2004 and 2005 led to a configuration of our search engine adapted to the context of INEX notably the collection characteristics. Since this configuration globally gave relatively good evaluation results, it was chosen for additional studies evaluating the influence of given parameters on our retrieval engine. These additional experiments intended to estimate the capacity of our search engine to be adapted to different types of searches.

5.1 The INEX framework

INEX provides testbeds (collection + topics + assessments) and evaluation methods for XML retrieval. Participants can evaluate and compare their results. The experiments presented are based on the INEX 2005 testbed. INEX 2005 documents correspond to a set of articles from the IEEE Computer Society marked up in XML. INEX introduces two types of queries on content only (CO) and mixing content and structure (CAS). For each type of queries, different tasks are defined (e.g CO.Thorough, VVCAS) modelling different retrieval scenarios. Relevance is defined according to two dimensions: exhaustivity and specificity. Exhaustivity describes the extent to which an element discusses the topic. Specificity describes the extent to which an element focuses on the topic. INEX 2005 defines different evaluation metrics: normalized extended cumulative gain (nxCG), effort-precision (ep), Q and R. Details on these metrics can be found in (Kazai and Lalmas, 2005). The results presented in this paper are based on the main overall indicator MAep (Mean Average ep) of the official system-oriented evaluation based on the ep measures. Two quantization functions are defined to model different user preferences. The strict quantization corresponds to searching only fully specific and highly exhaustive elements while the generalised quantization corresponds to having interest in all relevant elements. Experiments were done on CO.Thorough and VVCAS subtasks that are similar tasks on different sets of topics.

5.2 Studied parameters

The studied parameters in the experiments presented in this paper are query presence factor and score propagation. We tried to evaluate the influence of these parameters regarding the different quantizations and different tasks that is to say different retrieval scenarios.

The different runs are labelled in order to describe the configuration of the experiments as follows:

- xpY indicates the value Y of the coefficient φ used for the function measuring the presence of the query in the element,
- pr0Z indicates the value 0.Z of the coefficient α used for score propagation.

5.3 Influence of propagation

Experiments were done to evaluate the possibility to better respond to a given retrieval scenario when varying score propagation. According to the definition of the scoring principle as presented in section 3.3, weak propagation supports leaf nodes of XML documents or nodes near leaves. The first hypothesis was that in the context of INEX this corresponds to support highly specific nodes and leads to obtain better evaluation results for strict quantization. Increasing propagation includes higher

nodes in the hierarchical structures of XML documents. The second hypothesis was thus that in the context of INEX this corresponds to include in results less specific nodes and leads to obtain worse evaluation results for strict quantization but better results for generalized quantization. However, it is expected that a too strong propagation can deteriorate results for both strict and generalized quantizations since highly specific elements can disappear from the results to the profit of less specific elements. The experiments can show the threshold from which this occurs.

Table 1. Influence of score propagation without query presence factor for Thorough task

Metric : ep/gr (MAep), Task: Thorough		
Run	**Quantization**	
	strict	**generalized**
pr01xp1	**0,038**	0,037
pr03xp1	0,031	0,052
pr05xp1	0,017	**0,062**
pr07xp1	0,011	0,056
pr09xp1	0,008	0,047

Table 2. Influence of score propagation with strong query presence factor for Thorough task

Metric : ep/gr (MAep), Task: Thorough		
Run	**Quantization**	
	strict	**generalized**
pr01xp1000	**0,033**	0,047
pr03xp1000	0,031	0,059
pr05xp1000	0,028	0,066
pr07xp1000	0,023	**0,067**
pr09xp1000	0,016	0,063

The results confirm that increasing score propagation deteriorates the results for strict quantization whatever the query presence factor. For generalized quantization, increasing propagation improves the results before to deteriorate them when propagation becomes too strong.

The best results seem to be obtained for score propagation between 0.5 and 0.7. Applying significance test (paired t-test) between results shows significant differences between runs except between the runs pr05xp1000 and pr07xp1000 for generalized quantization.

5.4 Influence of query presence

Experiments were also done to evaluate the possibility to better respond to a given retrieval scenario when varying query presence factor. According to the definition of the scoring function as presented in section 3.2, a high query presence factor supports XML nodes containing several query terms even if they appear frequently in the collection. However, nodes containing the most query terms (i.e. highly exhaustive) remain top ranked. In the context of INEX this corresponds to promote partially exhaustive nodes.

For generalized quantization results are expected to be better with a query presence factor since more exhaustive nodes are top ranked. However, a too strong query presence factor can deteriorate results since it can promote too many partially exhaustive nodes. For strict quantization, a query presence factor can compensate the elimination of specific nodes caused by a too strong score propagation. With weak score propagation a query presence factor is not expected to improve evaluation results since it can promote partially exhaustive nodes instead of highly exhaustive ones.

Table 3. Influence of query presence factor with intermediate score propagation for Thorough task

Metric : ep/gr (MAep), Task: Thorough		
Run	**Quantization**	
	strict	**generalized**
pr06xp1	0,013	0,060
pr06xp50	0,020	**0,069**
pr06xp500	0,024	0,068
pr06xp1000	0,024	0,068
pr06xp3000	**0,025**	0,067

Table 4. Influence of query presence factor with weak score propagation for Thorough task

Metric : ep/gr (MAep), Task: Thorough		
Run	**Quantization**	
	strict	**generalized**
pr01xp1	**0,038**	0,037
pr01xp50	0,034	0,032
pr01xp500	0,033	0,036
pr01xp100	0,033	0,047
pr01xp3000	0,032	**0,048**

The results confirm that with score propagation (Table 3) to increase the query presence factor leads to better evaluation results for strict quantization. For generalized quantization a slight query presence factor leads to better results while continuing to increase this factor seems to deteriorates the results.

With weak score propagation (Table 4) the results are better for strict quantization without query presence factor. On the contrary for

generalized quantization to increase the query presence factor leads to better evaluation results.

Applying significance test (paired t-test) between results shows significant differences between runs except between the runs pr01xp50 and pr01xp500 and between pr06xp50, pr06xp500 and pr06xp1000. For strict quantization, significance test shows significant differences only for pr06 runs except between pr06xp500, pr06xp1000 and pr06xp3000.

5.5 Synthesis

Regarding the results when varying the main parameters and regarding the different quantizations modelling different retrieval scenarios, we can identify how to adapt the search engine to each case. When searching only XML elements highly exhaustive and fully specific (i.e. scenario: thorough, strict) the adapted configuration of the search engine is with weak score propagation (α = 0.1) and without query presence factor (φ = 1). When searching all the relevant XML elements (i.e. scenario: thorough, generalized) the adapted configuration of the search engine is with intermediate score propagation (α = 0.6) and with weak query presence factor (φ = 50).

Additional experiments were performed on CAS topics defined in INEX 2005 according to VVCAS evaluation. Regarding evaluation, the VVCAS task criteria are similar to the CO.Thorough task. These experiments confirm the search engine behaviour according to the studied parameters. Regarding quantizations same configurations that for CO.Thorough task lead to the best results.

6 CONCLUSIONS

In this paper, we proposed a search engine for XML retrieval. This engine is based on the addition of contributions brought by each component of the user's information need. The search engine is largely configurable intended to be adapted to different contexts such as different retrieval scenarios. Through experiments using the INEX framework we have evaluated the influence of different parameters on the effectiveness of the search engine. The experiments confirm that the engine presented can be adapted to better respond to different retrieval scenarios and how to adapt it to a given scenario.

However, experiments have been analysed globally using average precision. Like other participants to INEX, our search engine has variable effectiveness at query level. Additional studies have to be carried out at the query level to define criteria that permits to adapt the search engine per topic. Some works (Sigurbjörnsson et al., 2005)(Pehcevski et al., 2005) tried to define categories among the INEX topics. It would be interesting to study our search engine according to these categories. Fusion of different results obtained with different configurations of our search engine is also a considered future work.

REFERENCES

Amer-Yahia S., Lakshmanan L., Pandit S., 2004. FleXPath: Flexible Structure and Full-Text Querying for XML, *ACM SIGMOD*, Paris, pp. 83-94.

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

Bray T., Paoli J., Sperberg-McQueen C. M., Maler E., Yergeau Y., 2004. *Extensible, Markup Language (XML) 1.0. (Third Edition)*, W3C Recommendation.

Carmel D., Maarek Y. S., Mandelbrod M., Mass Y., Soffer A., 2003. Searching XML documents via XML fragments, *26th international conference SIGIR*, Toronto, pp. 151-158.

Clark J., DeRose S., 1999. *XML Path Language (XPath), W3C Recommendation*.

Crouch C. J., Apte S., Bapat H., 2003. An Approach to Structured Retrieval Based on the Extended Vector Model, *2nd INEX Workshop*, Dagstuhl, pp. 89-93.

Fuhr N., Großjohann K., 2004. XIRQL: An XML query language based on information retrieval concepts, ACM TOIS, vol. 22, Issue 2, pp. 313-356.

Fuhr N., Maalik S., Lalmas M., 2003. Overview of the INitiative for the Evaluation of XML Retrieval (INEX) 2003, *2nd INEX Workshop*, Dagstuhl, pp. 1-7.

Geva S., 2005. GPX – Gardens Point XML Information Retrieval at INEX 2004, *LNCS 3493, INEX'04, 3rd International Workshop*, Dagstuhl, p. 211-223.

Hubert G., 2005. A voting method for XML retrieval, *LNCS 3493, INEX'04, 3rd International Workshop*, Dagstuhl, p. 183-196.

Kazaï G., Lalmas M., 2005. INEX 2005 Evaluation Metrics, *Pre Proceedings of the 4th INEX Workshop*, pp. 401-406.

Liu S., Zou O., Chu W. W., 2004. Configurable indexing and ranking for XML information. *27th International Conference SIGIR,* Sheffield, pp. 88-95.

Ogilvie P., Callan J., 2003. Using Language Models for Flat Text Queries in XML Retrieval, *2nd INEX Workshop*, Dagstuhl, pp. 12-18.

Pehcevski J., Thom J. A., Tahaghoghi S. M. M., 2005. Hybrid XML Retrieval Revisited, *LNCS 3493, INEX'04, 3rd International Workshop*, Dagstuhl, pp. 153-167.

Piwowarski B., Vu H.-T., Gallinari P., 2003. Bayesian Networks and INEX'03, *2nd INEX Workshop*, Dagstuhl, pp. 33-37.

Ponte J. M., Croft W. B., 1998. A Language Modeling Approach to Information Retrieval, *21st International Conference SIGIR,* Melbourne, pp. 275-281.

Salton G., Wong A., Yang C. S., 1975. A vector space model for automatic indexing*,* Communication of the ACM, vol. 18, Issue 11, pp. 613-620.

Sigurbjörnsson B., Kamps J., de Rijke M., 2005. Mixture Models, Overlap, and Structural Hints in XML Element Retrieval, *LNCS 3493, INEX'04, 3rd International Workshop*, Dagstuhl, pp. 196-210.