HOW INTEGRATE STRUCTURAL DIMENSION IN RELEVANCE FEEDBACK IN XML RETRIEVAL

Inès Kamoun Fourati, Mohamed Tmar and Abdelmajid Ben Hamadou
High Institute of Computer Science and Multimedia, University of Sfax, Sfax, Tunisia

Keywords: Relevance feedback, XML, INEX, Line of descent matrix.

Abstract: Contrarily to classical information retrieval systems, the systems that treat structured documents include the structural dimension through the document and query comparison. Thus, the retrieval of relevant results means the retrieval of document fragments that match the user need rather than the whole document. So, the structure notion should be taken into account during the retrieval process as well as during the reformulation by relevance feedback way.
In this paper we propose an approach of query reformulation based on structural relevance feedback. We start from the original query on one hand and the fragments judged as relevant by the user on the other. Structure hints analysis allows us to identify nodes that match the user query and to rebuild it during the relevance feedback step. The main goal of this paper is to show the impact of structural hints in XML query optimization. Some experiments have been undertaken into a dataset provided by INEX (INitiative for the Evaluation of XML retrieval, an evaluation forum that aims at promoting retrieval capabilities on XML documents) to show the effectiveness of our proposals.

1 INTRODUCTION

The goal of information retrieval systems (IRS) is to satisfy informational needs of a user. This need is expressed in the form of a query to be matched to the all documents in the corpus to select those who could answer to the user's query. Because of the ambiguity, and the incompleteness of his query, the user is in most cases confronted with the problems of silence or noise. To overcome this problem, there must be alternatives to the initial query so as to improve the research. Among the most popular patterns in information retrieval (IR), we cite the relevance feedback (RF) which, since the first attempts in IR has become a crucial phase. It is based on the judgments of relevance of the documents found by the IRS and is intended to re-express the information needs from the initial query in an effort to find more relevant documents.

Due to the great importance of structured information, XML documents cover a big part not only on the web, but also on modern digital libraries, and essentially on Web services oriented software (Bray et al., 2000). This standardization of the Web to XML schemas presents new problems and hence new needs for customized information access. Being a very powerful and often unavoidable tool to customized access to information of all kinds, information retrieval systems arise at the forefront of this issue.

However, the traditional IRS do not exploit this structure of documents, including the RF phase. However, a structured document is characterized by its content and structure. This structure possibly completes semantics expressed by the content and becomes a constraint with which IRS must comply in order to satisfy the user information needs. Indeed, the user can express his need by a set of keywords, as in the traditional IRS, and can add structural constraints to better target the sought semantics.

Thus, taking into account the structure of the documents and that of the query by the information retrieval systems handling structured documents is necessary in the feedback process.

We propose in this paper to evaluate the impact of structure handling in query reformulation process by structural relevance feedback way. The structure hints in the user query are taken into account at first (before any content treatment) in the query reformulation process, the query structure could be devoted to some modification based on the structure of the relevant judged document fragments. Thus, we put the emphasis on the structure by analyzing the structure

Kamoun Fourati I., Tmar M. and Ben Hamadou A..
HOW INTEGRATE STRUCTURAL DIMENSION IN RELEVANCE FEEDBACK IN XML RETRIEVAL.
DOI: 10.5220/0003300305770582
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 577-582
ISBN: 978-989-8425-51-5

features and relation that are the most significant to the relevance feedback process.

This paper is organized into four sections. The second section gives a survey of related work in RF in XML retrieval. The third section presents our approach in this context. In the fourth section, we present the experiments and the obtained results. The fifth section concludes.

2 RELATED WORK

Schenkel and Theobald (Schenkel and Theobald, 2005) describe two approaches which focus on the incorporation of structural aspects in the feedback process. Their first approach reranks results returned by an initial, keyword-based query using structural features derived from results with known relevance. Their second approach involves expanding traditional keyword queries into content-and-structure queries. Official results, evaluated using the INEX 2005 (Fuhr et al., 2006) assessment method based on rank-freezing, show that reranking outperforms the query expansion method on this data.

Sauvagnat et. al. (Hlaoua et al., 2007),describe their experiments in relevance feedback as follows: The "structure-oriented" approach first seeks to identify the generic structure shared by the largest number of relevant elements and then they use this information to modify the query. A second method, called "content-oriented", utilizes terms from relevant elements for feedback. A third method involves a combination of both approaches. Official results show improvement in some case but are not consistent across query types.

Mass and Mandelbrod (Mass and Mandelbrod, 2004) propose an approach that determines the types of the most informative items or components in the collection (articles, sections, and paragraph for INEX) and creates for each type its index. The automatic query reformulation process is based on identifying its best elements from an ordered list to select the most relevant ones. The scores in the retrieved sets are normalized to enable comparison across indices and then scaled by factor related to the score of containing article. They use the Rocchio algorithm (Rocchio, 1971) associated with the lexical affinity.

Among these approaches, only a few consider that RF in the query structure is necessary. It is common to rewrite the query based on its structure, and the content of the relevant elements, without any modification of the query structure itself. In our approach, we consider the structural RF is necessary, particularly if the XML retrieval system takes into account the structural dimension in the matching process. Since we use an XML retrieval system that matches the structure in addition to the content (Aouicha, 2009), we assume that the structure reformulation could improve the retrieval performance.

3 STRUCTURAL RELEVANCE FEEDBACK: OUR APPROACH

In our approach we propose to integrate structural dimension in RF in XML retrieval. So, we focus on the structure of the original query and that of document fragments deemed relevant to the user structure hints. An example is shown in figure1.

Indeed, this study allows us to reinforce the importance of these structures in the reformulated query to better identify the most relevant fragments to the user's needs. The analysis of structures allows us to identify the most relevant nodes and the involved relationships.

In our approach we essentially manipulate the structure of an XML tree, so, we propose to present some XML tree's basics notions . Then, we present our approach, which is based on two major phases. The first aims at representing the query structure and the judged relevant fragment one in single representative structure. The second is focused on query rewriting.

3.1 The XML Tree

An XML document is composed by a set of structural elements e called *doxels* (or *Document Element*). The set of these *doxels* is called $\mathcal{E}$. There is 3 kinds of elements in XML document:

- Elements $\mathcal{E}^{/sana}$: associated with alphanumeric label. Example: document, section, body, paragraph,...)
- Attributes $\mathcal{E}_{@}$: prefixed by @. An attribute is associated with only one element in $\mathcal{E}_{/}$.
- Data $\mathcal{E}_{\#}$ represented by #PCDATA. These element contain data.

In XML retrieval we focus essentially on the structure of documents and we consider *doxels* of $\mathcal{E}_{/}$. This structure is defined by a set of links between *doxels*. The link $l \in \mathcal{L} = \mathcal{E} \times \mathcal{E}$ relies two *doxels* and defines a direction. If the link is from e to e', we note (e, e').

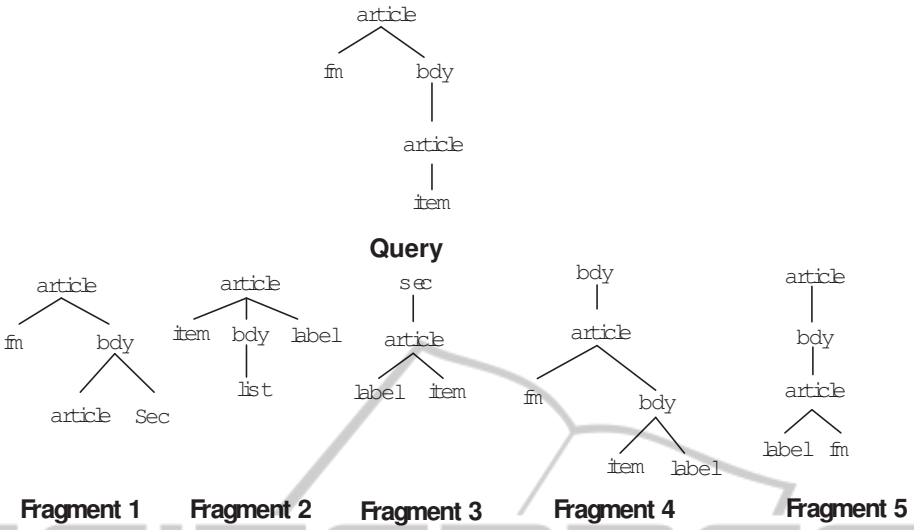


Figure 1: Example of query structure and relevant fragments.

3.2 Query and Relevant Fragment Representation

According to most approaches of relevance feedback, the query construction is done by building a representative structure for relevant objects and another structure for irrelevant ones, and then build a representation close to the first and far from the second.

For example, the Rocchio's method (Rocchio, 1971) considers a representative structure of a document set by their centroid. A linear combination of the original query and the centroids of the relevant documents and irrelevant ones can be assumed as a potentially suitable user need.

Although simplistic, the Rocchio's method is the most widespread. But Rocchio's method is adapted to the case where documents are full text, a context in which each document is expressed by a vector and where the documents embody structural relations, the vector representation becomes simplistic. Therefore the reconstruction of a unified structure causes new problems.

As for us, we believe that the structure is an additional dimension. A unique dimension is not enough to encode the structural information (one dimension vector), thus we need to encode all documents into two dimensions, by using matrices rather than vectors.

That reasoning has led us to traduce the documents and the query in a matrix format instead of a wighted term vector. Those matrices are enriched by values calculated from transitive relationship function. Then, the representative structure of query and judged relevant fragments (that we call S) is constructed under a matrix form.

3.2.1 Line of Descent Matrix

We build for each document a matrix called *line of descent matrix* (LDM), which must show all existing ties of kinship between different nodes. This representation should also reflect the positions of the various nodes in the fragments as they are also important in the structural relevance feedback. For an XML tree (or subtree) A, we associate the matrix defined by M_A:

$$M_A[e,e'] = \begin{cases} P & \text{if } e' \in son(e) \\ 0 & \text{otherwise} \end{cases}$$

Where P is a constant value which represents the weight of the descent relationship and $son(e) = \{e'/(e,e') \in (L)\}$

As for us, we represent each of the relevant fragments and the initial query in the LDM form. The value of the constant P for the query LDM construction is greater than that used for the construction of other LDMs (which represent the relevant fragments) to strengthen the weight of the initial query edges following the principle used in the Rocchio's method which uses reformulation parameters having different effects (1 for the initial query, α for the relevant documents centroid and β for the non relevant documents centroid where $0 \leq \alpha \leq 1$ and $-1 \leq \beta \leq 0$).

Note that no complexity analysis is here needed because of the low number of relevant judged documents comparing to the corpus size. In our experiments, we undertake the relevance feedback in a pseudo-feedback way on the top 20 ranked documents resulting from the first round retrieval. In the other

hand, the total number of tags is over 160 in all the collection (INEX'05 collection) and about 5 in a single fragment, so the matrix size can not exceed $5 * 5$.

3.2.2 Setting Relationship between a Node and its Descendants

XML retrieval is usually done in a vague way (Mihajlovic and Ramirez, 2005). A fragment can be returned even if the structural conditions of the query are not entirely fulfilled. This means that if a fragment of an XML document is similar but not identical to the query, it can be returned. The information retrieval systems now has to query with tolerated differences (a few missing elements or more additional ones) between the query structure and the document. Consequently, we believe that the most effective way to bring this tolerance is to assure that one element is not only connected to its child nodes, but to all of its direct and indirect descendants. A relationship between nodes in the same line of descent is weighted by their distance in the XML tree.

For example, in fragment 3 (represented in figure 1) the node *sec* is the parent of *article* and the latter is the parent of *label*, the descent link between *sec* and *label* is weighted with a value that depends on the weight of the link between *sec* and *article* and that between *article* and *label*.

So, we propose TR function which is a transitive relationship on the weights of the nodes edges with a common ancestor. The resulted value will be added to the weight of the edge itself in the LDM as follows:

$$\forall (e,e',e'') \in N^3, \text{ if } (n' \in son(n) \text{ and } n'' \in son(n')) \text{ then } M_A[e,e''] \leftarrow M_A[e,e''] + TR(M_A[e,e'], M_A[e',e''])$$

where N is the set of all different nodes in the tree A and M_A is its LDM.

TR function is more detailed in (Fourati et al., 2006). After some research, we use the following function as a meeting of these criteria:

$$TR(x,y) = \frac{x \times y}{\sqrt{x^2 + y^2}}$$

As for us, this transitive relationship will be applied to each LDM of each fragment judged as relevant and also to the LDM of the query. The figure 2 illustrate an XML tree and associated LDM.

3.2.3 Matrix S Construction

To represent the query structure and judged relevant fragment one in single representative structure we consider $F = \{A_1, A_2 \ldots A_n, Req\}$ where Req is the initial query and A_i are the relevant judged fragments, the query structure is built starting from the cumulated LDM S:

$$\forall (e,e')^2 \in B^2, S[e,e'] = \sum_{A \in F} M_A[e,e']$$

If a column contains several low values, then the node will tend to appear as a leaf node in the reformulated query. If on the contrary one row contains several low values, then the node will tend to be seen as a root node in the reformulated query. If, in addition, the corresponding column contains several high values, otherwise, the node will tend to appear as an internal node. Thus, in order to build the new query structure, we can determine the new root.

3.3 Structural Query Rewriting

3.3.1 Root Identification

The structure query construction starts by identifying its root. The root is characterized by a high number of child nodes and a negligible number of parents. For example, to find the root we simply return the element R, which has the greatest weight in the rows of the matrix S and the lowest weight in its columns. The root R is then such that:

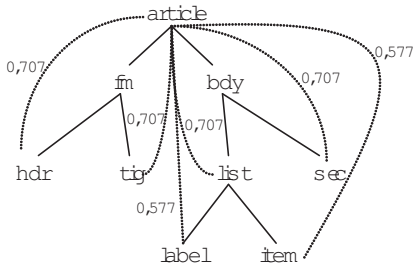
$$R = \arg\max_{e \in B} \sum_{i=0}^{i=n} S[e,e'].\log\left(\frac{\sum\limits_{e' \in B} S[e',e]}{\sum\limits_{(e',e'') \in B^2} S[e',e'']} + 1\right)$$

The argument to maximize reflects that the candidate nodes to represent the root should have as maximal low values as possible in the relative row ($\sum\limits_{e' \in B} S[e,e']$) and as minimal low values as possible in the column ($\sum\limits_{e' \in B} S[e',e]$) relatively to the total sum of the matrix values ($\sum\limits_{(e',e'') \in B^2} S[e',e'']$). In our example the element *article* will be the root of the new query.

3.3.2 Building the New Query Structure

Once the root has been established from the matrix S, we proceed to the recursive development phase of the tree representing the structure of the new query. The development of the tree starts by the root R, and then by determining all the child nodes of R, the same operation is performed recursively for the child nodes of R until reaching the leaves elements. Each element e is developed by attributing to it its potentially child nodes e' ($e' \neq e$) whose $S[e,e'] > Threshold_n$.

We assume that $Threshold_n$ is calculated from the mean average μ_n and the standard deviation σ_n



	1	2	3	4	5	6	7	8	9
1	0	1	1	0.707	0.707	0.707	0.707	0.577	0.577
2	0	0	0	1	1	0	0	0	0
3	0	0	0	0	0	1	1	0	0.707
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	1
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0

Figure 2: Example of LDM (P=1).

of its relative child nodes. This threshold is defined as follows: $Threshold_e = \mu_e + \gamma * \sigma_e$ where $\mu_e = \frac{1}{|N|} \sum_{e' \in N} S[e,e']$ and $\sigma_e = \frac{1}{|N|} \sqrt{\sum_{e' \in N} (S[e,e'] - \mu_e)^2}$ If the value of γ is relatively high, the tree outcome will tend to be shallow and ramified and vice versa. The value of γ allows the estimation for each element of the number of child nodes. The objective of this interval is to reconstruct a tree as wide and deep as the XML fragments from which the query should be inferred. This value is then defined experimentally.

4 EXPERIMENTS AND RESULTS

Our experiments have been undertaken into INEX'05 dataset which contain 16819 articles taken from IEEE publications in 24 journals. The INEX metrics used for evaluating systems, are based on two dimensions of relevance (exhaustivity and specificity) which are quantized into a single relevance value. We distinguish two quantization functions :

- A strict quantization to evaluate whether a given retrieval approach is able of retrieving highly exhaustive and highly specific document components,

$$f_{strict}(s,e) = \begin{cases} 1 & if\ (e,s) = (2,1) \\ 0 & otherwise \end{cases} \quad (1)$$

- A generalized quantization to evaluate document components according to their degree of relevance.

$$f_{generalized}(s,e) = e \times s \quad (2)$$

Official metrics are based on the extended cumulated gain (XCG) (Kazai and Lalmas, 2005). The XCG metrics are a family of metrics that aim to consider the dependency of XML elements within the evaluation. The XCG metrics include the user-oriented measures of normalized extended accumulated gain ($nXCG$) and the system-oriented effort-precision/gain-recall measures (ep/gr). The xCG metric accumulates the relevance scores of retrieved documents along a ranked list.

For a given rank i, the value of $nxCG[i]$ reflects the relative gain the user accumulated up to that rank, compared to the gain he could have attained if the system would had produced the optimum best ranking. For any rank the normalized value of 1 represents the ideal performance.

The effort-precision ep is defined as:

$$ep(r) = \frac{e_{ideal}}{e_{run}} \quad (3)$$

where e_{ideal} is the rank position at which the cumulated gain of r is reached by the ideal curve, and e_{run} is the rank position at which the cumulated gain of r is reached by the system run. A score of 1 reflects the ideal performance where the user needs to spend the minimum necessary effort to reach a given level of gain.

In evaluation, we use the uninterpolated mean average effort-precision denoted as $MAep$ which is calculated as the average of effort-precision values measured at each natural gain-recall points.

To carry out our experiments we only considered the VVCAS (Fuhr et al., 2006) (queries whose relevance vaguely depends on the structural constraints) type queries because the need for reformulation of the query structure is appropriate to the task. We present only the results using generalized quantization which is most suitable for VVCAS queries.

The table 1 shows the results obtained from the research system based on tree matching (Aouicha et al., 2008). This table presents a comparison between the

Table 1: Comparative results before (BRF) and after (ARF) structural RF.

Run	*nxCG*[10]	*nxCG*[25]	*nxCG*[50]	MAep
BRF	0.1778	0.1593	0.1336	0.099
ARF	0.2430	0.2396	0.2195	0.0817

values obtained before RF (BRF), after RF (ARF). Note that BRF represent the result of base run.

We can see through our experiments that our RF approach significantly improves the results. We note that during these experiments we reformulate only the queries structures without changing their original content, and therefore we believe that this reformulation has brought an evolution that could be accentuated by the reformulation of the content.

5 CONCLUSIONS AND FUTURE WORK

We have proposed in this paper an approach to structural relevance feedback in XML retrieval. We proposed a representation of the original query and relevant fragments under a matrix form. After some processing and calculations on the obtained matrix and after some analysis we have been able to identify the most relevant nodes and their relationships that connect them.

The obtained results show that structural relevance feedback contributes to the improvement of XML retrieval. The strategy of the reformulation is based on a matrix representation of the XML trees deemed relevant to the fragments and the original query. This representation preserves the original links of descent and the transformations achieved favors the flexibility of the research.

We plan, in short term and in order to improve our results to reformulate the content of the initial query relying on the terms having the greatest weight in the relevant elements. The selected terms will be injected in the content of the query elements. We plan also to conduct out tests on other corpus notably that of Wikipedia.

REFERENCES

Aouicha, M. B. (2009). *Une approche algébrique pour la recherche d'information structurée*. Thése de doctorat, Université Paul Sabatier, Toulouse, France.

Aouicha, M. B., Tmar, M., Boughanem, M., and Abid, M. (2008). Modèle de recherche d'information structurée basé sur la relaxation de requêtes. *INFORSID*.

Bray, T., Paoli, J., and Sperberg-McQueen, C. M. (2000). World wide web consortium (w3c). extensible markup language (xml) 1.0. *http://www.w3.org/TR/REC-xml*.

Fourati, I. K., Tmar, M., and Boughanem, M. (2006). Reformulation automatique de la requête en recherche d'information structurée. In *INFORSID*, pages 263–274.

Fuhr, N., Lalmas, M., Malik, S., and Kazai, G., editors (2006). *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, volume 3977 of *Lecture Notes in Computer Science*. Springer.

Hlaoua, L., Torjmen, M., Sauvagnat, K. P., and Boughanem, M. (2007). Xfirm at inex 2006. ad-hoc, relevance feedback and multimedia tracks. *International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), Dagstuhl, Allemagne, 18/12/2006-20/12/2006*, pages 373–386.

Kazai, G. and Lalmas, M. (2005). Inex 2005 evaluation metrics. *INEX 2005Workshop Proceedings*, pages 401–406.

Mass, Y. and Mandelbrod, M. (2004). Relevance feedback for xml retrieval. *In INEX 2004 Proceedings, Dagstuhl, Allemagne*.

Mihajlovic, V. and Ramirez, G. (2005). Tijah scratches inex 2005: Vague element selection, overlap, image search, relevance feedback, and users. *INEX 2005 Proceedings*, pages 54–71.

Rocchio, J. (1971). *Relevance feedback in information retrieval*. Prentice Hall Inc., englewood cliffs, nj edition.

Schenkel, R. and Theobald, M. (2005). Relevance feedback for structural query expansion. In (Fuhr et al., 2006), pages 344–357.