**International Conference on Automation, Control, Engineering and Computer Science (ACECS'14)**
**Proceedings - Copyright IPCO-2014, pp.158-163**
**ISSN 2356-5608**

# XML Element Retrieval using terms propagation

Samia Berchiche-Fellag[#1], Mohamed Mezghiche[#2]

*#Université Mouloud Mammeri de Tizi-Ouzou,UMMTO, Université M'Hamed Bougara Boumerdes, Algérie.*

*Tizi-Ouzou, Algérie.*

[1]samfellag@yahoo.fr
[2]mohamed.mezghiche@yahoo.fr

***Abstract*— In this paper, we are interested in content–oriented XML information retrieval which aims to retrieve focused parts of documents (elements) that match the user needs. These needs can be expressed through content queries composed of simple keywords. Our approach is based on terms propagation method, which aim is to assign a set of representative terms for each node of the document to allow an automatic selection of a combination of elements that better answers the user's query. Our method has been tested on the «Focused» task of INEX 2006, and has been compared to structured retrieval model which uses relevance propagation. Obtained results shown a significant improvement in the retrieval process efficiency.**

***KeyWords*— XML, terms propagation, CO query, element, INEX**

## I. INTRODUCTION

Extensible Markup Language (XML) is widely used as a standard document format in many application domains. XML documents are semi-structured documents which organize text through semantically meaningful elements labelled with tags.

Structural information of XML documents is exploited by Information Retrieval Systems (IRS) to return to users the most exhaustive[1] and specific[2] [1] documents parts(i.e. XML elements, also called nodes) answering to their needs. These needs can be expressed through Content queries (CO: Content Only) which contain simples keywords or through Content And Structure queries (CAS) which contain both keywords and structural information on the location of the needed text content.

[1] *An element is exhaustive to a query if it contains all the required information*

[2] *An element is specific to a query if all its content concerns the query*

Most of the structured retrieval models are adaptation of traditional retrieval models. The main problem is that the classical IR methods work on statistics such as term frequency and document frequency at the document level. This does not perform well at the node level as explained in [2], [3], [4].

The challenge in XML retrieval is to return the most relevant nodes that satisfy the user needs. The challenge is greater with CO queries where the user doesn't know anything about the collection structure and express her query in free text. The IRS exploits the XML structure to return the most relevant XML nodes that satisfy the user needs. It is precisely this issue that we propose to deal by providing a method which consists of searching the relevant nodes to user's query composed of simple keywords (CO query) in a large set of XML documents and taking into account the contextual relevance. Our search process is based on a method of terms propagation.

This paper is organized as follows, in section II we introduce an overview of works dealing with structured retrieval models. In section III we present our baseline model, which uses terms propagation method; and finally in section IV we present our experiments results.

## II. RELATED WORK

Several structured retrieval models have adapted traditional IR approaches to address the user information needs in XML collection. Some of these methods are based on the vector space model [5], [3], [6] or on the probabilistic model [7]. Language models are also adapted for XML retrieval [8], [9], as well as Bayesian networks in [10].

IRS dealing with XML documents aim to retrieve the most relevant nodes the user need. For this purpose, several approaches based on propagation methods were proposed. Relevance propagation, terms propagation and weights propagation. In the relevance propagation approach, relevance score of leaf nodes in xml document tree is

calculated and propagated to ancestors. Authors in [11] used linear combination of children's scores called "maximum-by-category » and « summation ».While the relevance propagation in [12] using XFIRM system is function of the distance that separates nodes in the tree. In [13], [14] authors used a method of weights propagation. For computing the eights of inner nodes, the weights from the most specific nodes in the document multiplied with an augmentation factor are propagated towards the inner nodes. Authors in15], [16] , and [17] used terms propagation method. In this case, textual content of leaf nodes in XML document is propagated to their ancestor considering some conditions. Authors in [15] and [17] exploited both structural information and the statistics of term distributions in structured documents. In [16], a leaf node is represented by a set of weighted terms. These terms are propagated to their ancestors by reducing their weight depending on the distance that separates nodes in the tree. As a conclusion, whatever the considered approach, the relevance node's score strongly depends on its descendants' scores.

## III. PROPOSED APPROACH

We consider an XML document *D* as a tree, composed of simple nodes *ni*, leaf nodes *lni* and attributes *ai*. The textual information (terms) is at the leaf nodes *lni*. Weights are assigned to terms in leaf nodes and weights of inner nodes are computed dynamically during the propagation.

Example of such document is given on fig.1.

```
<Article>
<Date ="01/01/2003">
<Heading>
<Title> XSL </Title>
 <Author> Ed Tittel</Author>
</Heading>
<Body>
 <Section>
   <Subtitle> XSLT </Subtitle>
    <Par> eXtensible Stylesheet Language …
</Par>
 </Section>
 <Section>
   <Subtitle> XSL-FO </Subtitle>
   <Par> The semi structured data …</Par>
</Section>
</Body>
</Article>
```

Fig. 1 Example of XML document

Fig 2 bellow is the tree representation of XML document in fig. 1.

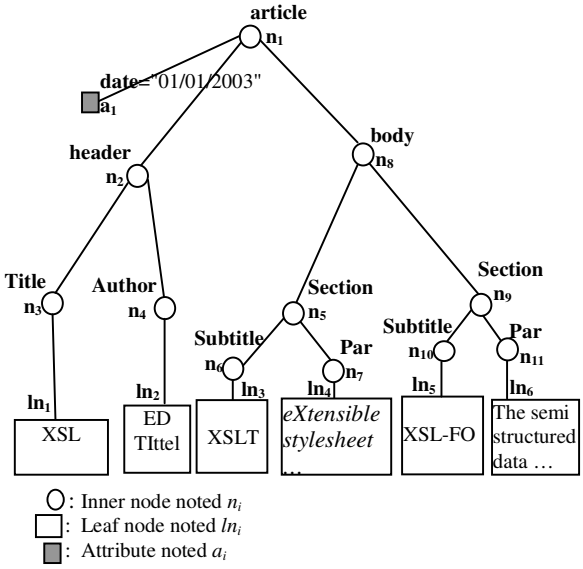


Fig. 2 Tree representation of the XML document in fig.1

### *A. Query processing*

The query processing in our method based on terms propagation is carried out as follows:

- We first assign weight values to terms in leaf nodes,
- We prune the document tree, retaining only informative nodes,
- We Propagate the well distributed terms in the leaf nodes, to their ancestor, in order to identify relevant and informative units
- At the end, we evaluate the relevance score of the identified information units (nodes) and present the results descending scores.

### *B. Weighting terms in leaf nodes*

The first essential step in query processing is to assign weights to terms in leaf nodes *ln*. We used for this purpose the formula (1) which is one of the adaptations from traditional IR to the granularity used in structured IR "*the element"*. Indeed in tradional IR we frequently use *tf×idf* where *Idf* is the inverted document frequency while *ief* is inverted element frequency.

$$w_k = tf_k \times ief_k. \quad (1)$$

Where:

$w_k$: is the weight of term *k* in leaf node *ln*

$tf_k$ : frequency of term $k$ in the leaf node

$Ief_k$: Inverted element frequency = $\log\left(\frac{Ne}{ne}+\alpha\right)$ with $0.5 \le \alpha \le 1$

*Ne*: number of leaf nodes in the document

*ne*: number of leaf nodes containing the term *k* in the document

*C. Evaluation nodes relevance value*

The final step in our query processing is to evaluate the relevance value of nodes $n_i$ according to the query.

Let $q = \{(t_1,w_{q1}), \ldots\ldots, (t_M,w_{qM})\}$ be a query composed of weighted keyword terms. $t_k$ a query term, $w_{qk}$ the weight of $t_k$ in query $q$ and $M$ is the number of terms in the query. Relevance values are computed thanks to a similarity function called $RSV(q,n_i)$ *of* the vector space model (Inner product) as follows:

$$RSV(q, n_i) = \sum_{k=1}^{M} w_{qk} \times w_{nik} \ . \qquad (2)$$

Where $w_{qk}$ and $w_{nik}$ evaluated with formula (1), are respectively the weight of the term *k* in the query *q* and in the node $n_i$.

*D. Terms Propagation*

The main issue in our terms propagation method is : *what terms propagate?*

We introduce in this aim, the concept of *informativeness* of the node. And we define it as follow:

*A node is informative if it carries sufficient information to satisfy a user query*.

The issue is: *how to measure it?* for this purpose we propose to take into account the node's size.

Indeed, a node that contains only the query terms, is specific to this query. However, not informative because it does not provide the required information to the user (eg a title node may be relevant to a query but is not informative). We define for that, a threshold that involves the minimum terms number in a node to be considered informative. It is clear that we have no theoretical way to determine this threshold. We propose to fix it by experiment as is frequently the case in the IR area.

Two cases in the terms propagation, be considered:

*Case A*: nodes whose number of terms is below the threshold.

*Case B:* nodes whose number of terms is greater than the threshold.

*Case A: nodes whose number of terms is below the threshold.*

The document tree is traversed starting from leaf nodes. During the path, when the number of terms in the visited node is below the threshold, the node is removed from the tree and its contents ascended to its parent node. This process is done recursively until reaching (and possibly exceed) the threshold, or reach the root node of the document, or reach an inner node whose terms number of at least one of its child is greater or equal than threshold.

*Case B: nodes whose number of terms is greater than the threshold.*

We consider fundamental hypothesis which expresses that: *"terms of a node well distributed in its child may be representative terms for this node."*

Two cases can occur, a node can have several child nodes, or have only one (leaf node only has no child):

*1. Case of node with several child nodes*

Intuitively, we can think that a term of a node can be representative for its parent node if it appears at least on one sibling node.

This intuition is insufficient; account should be taken to the weight of the term in the node. Indeed, a term of a node may belong to all its child nodes, but if its weight is low compared to the weight of other terms of the nodes. It cannot be discriminant for these nodes.

We consider in this aim another hypothesis which consists to take into account only the terms which average weight in the child nodes where they appear is between the average and the maximum weight of all the terms of child nodes.

*2. Case of node with one child node*

We consider the hypothesis which expresses that "term of a node is representative for its parent. If its weight is between the average and the maximum weight of all the terms of the node.

The term satisfying cases 1 or 2, is removed from its node and ascended to its parent node. Its weight in the parent node is equal to its average weight in the child nodes in case 1, or its weight in case 2.

These hypotheses are formalized as follows:

**1.** Let *e* be a node with several child nodes *e'*. Let *t* be a term of a child node *e'*. *w(t, e')* the weight of term *t* in node *e'*, calculated with the formula (1). *t* can be ascended to *e*, if *t* exists in at least one sibling node of e' and if the average weight of *t* in the child node of *e* where it appears, verify the following condition:

$$w_{avg} \le \underset{e' \in chl(e)}{avg}(w(t,e')) \le w_{max} \qquad (3)$$

Where :

$$w_{avg} = \frac{\sum_{e' \in chl(e)} \sum_{i=1}^{Nte'} w(t_i, e')}{Nt} \ . \qquad (4)$$

$W_{avg}$ : average weight of terms in the nodes *e'* child of *node e*

*chl(e):* child of node *e*

$$\text{avg}_{e' \in cl(e)}\left(w(t,e')\right) = \frac{\sum w(t,e')}{Ne'} \quad (5)$$

*Nte'* : number of terms in the node *e'*

*Nt* : number of terms in all nodes *e'* child of node *e*

*Ne'* : number of nodes *e'* containing the term *t*

$w_{max}$ : Maximum weight of terms in all nodes *e'* child of node *e*

The term *t* is removed from the child nodes *e'* and ascended to its parent node *e* its weight will be:

$$w(t,e) = \underset{e' \in chl(e)}{\text{avg}}(w(t,e')) \quad (6)$$

**2.** Let *e* be a node with only one child node *e'*. Let *t* be a term of node *e'*. *w(t, e')* the weight of term *t* in node *e'*, calculated with the formula (1). *t* can be ascended to *e*, if it satisfies condition (7):

$$w_{avg} \le w(t,e') \le w_{max} \quad (7)$$

Where :

$$w_{avg} = \frac{\sum_{i=1}^{Nte'} w(t_i,e')}{Nte'} \quad (8)$$

$W_{avg}$ : average weight of terms in node *e'*

$w_{max}$ : maximum weight of terms in node *e'*

*Nte'* : number of terms in the node *e'*

The term *t* is removed from the child node *e'* and ascended its parent node *e* with its weight:

$$w(t,e) = w(t,e') \quad (9)$$

Note that during the ascent of term *t* from child node *e'* to its parent node *e*, it may previously be present. In this case, the term *t* is removed from child node(s) e', and its weight in node *e* is equal to the average weight in child node(s) *e'* and the parent node *e* as follow:

$$w(t,e) = \frac{w_{(6)/(9)}(t,e) + w_0(t,e)}{2} \quad (10)$$

where:

$w_0(t,e)$ : initial weight of the term *t* in the node *e*

$w_{(6)/(9)}(t,e)$ : weight should have (if it did not exist) the term *t* in node *e* calculated using the formula (6) or (9).

At the end of the propagation process, the relevance score of the nodes represented by these terms according to the query terms is evaluated, the results are presented descending scores. The results nodes are relevant and informative.

## IV. EXPERIMENTATIONS

Our model has been tested and compared to XFIRM model which uses relevance propagation.

### *A. INEX: Initiative for the Evaluation of XML Retrieval*

We used for our experiments the INEX 2006 collection [18]. The main INEX evaluation purpose is to promote their search in XML documents by providing a test collection, and assessment procedures to allow participants to benchmark their results. The test collection consists of a set XML documents, queries and relevance judgments, and uses a collection made from English documents from Wikipedia. INEX consists of several tasks such as "focused" task,"thorough" task, "Best in context" task... We based our tests on the "focused" task.

### *B. Data Collection*

The collection contains about 659 388 documents and provides a set of 126 queries for evaluation. The features of this collection are presented in table 1.

TABLE 1: FEATURES OF INEX 2006 COLLECTION

| | |
|---|---|
| Collection size | **4.6 GO** |
| DocumentsNumber | **659388** |
| Links number | **16737300** |
| Topicsnumber | **126** |

### *C. Evaluation Protocol*

We experimented 32 queries on INEX 2006 collection. We used the normalized cumulated gain *nxCG[t*] measure which was used in the evaluation of the "focused" task in INEX 2006.

With *nxCG[t]* measure, system performance was reported at several rank cutoff values (t).

For a given topic, the normalized cumulated gain measure is obtained by dividing a retrieval run's xCG vector by the corresponding ideal xCI vector.

$$nxCG[i] = \frac{xCG[i]}{xCI[i]} \quad (11)$$

*xCG[i]* takes its values from the full recall-base of the given topic.

*xCI[i]* takes its values from the ideal recall-base and i ranges from 0 and the number of relevant elements for the given topic in the ideal recall base. 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 that could have attained if the system would have produced the optimum best ranking.

### *D. Results*

We have conduced preliminary experiments with 10 queries and a hundred of documents from INEX 2006 collection. This allowed us to set threshold value to 50. Obtained results of our experiments are presented in table2.

TABLE 2. RESULTS FOR THE « FOCUSED » TASK WITH THE NXCG METRIC AT DIFFERENT CUTOFFS

| | **NXCG5** | **NXCG10** | **NXCG25** | **NXCG50** |
|---|---|---|---|---|
| **XFIRM** | 0,587 | 0,298 | 0,224 | 0,170 |
| **Our Approach** | 0,765 | 0,349 | 0,261 | 0,191 |
| **% improvement** | **23,282** | **14,687** | **14,127** | **11,198** |

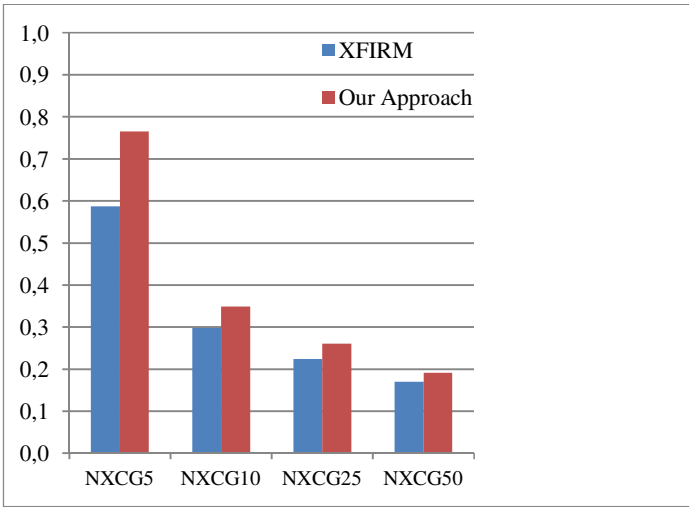


Fig. 3 comparative graph the achieved results of our approach to that of XFIRM

The results obtained with our method which is terms propagation, show a clear improvement and a significant gain compared to XFIRM which is relevance propagation. Improvement in results is observed on different gain values at 5, 10, 25 and 50 documents. Although, the highest performances are observed in the 05 and 10 documents.

We conclude that terms propagation method significantly improves the research results since it based on contextual relevance.

## V. CONCLUSION

We have presented in this paper our contribution to XML element retrieval for retrieving the most relevant part of XML documents that the user needs. We proposed for this terms propagation method, which aim is to not only return the most exhaustive and specific nodes to a user query but mainly informative nodes are returned with the constraint about node's size we imposed. Our method has been evaluated on the «Focused» task of INEX 2006, and has shown a significant improvement in the retrieval process efficiency compared to XFIRM system which used relevance propagation method.

## REFERECENCES


[1] M. Lalmas, *Dempster-Shafer's theory of evidence applied to structured documents: Modeling uncertainty*, In: Proceedings of ACM-SIGIR, pp. 110-118. Philadelphia, 1997.

[2] Y.Mass and M.Mandelbrod, *Retrieving the most relevant XML Component*, In: Proceedings of the Second Workshop of the Initiative for The Evaluation of XML Retrieval (INEX), 15-17. December, 2003.

[3] Y.Mass and M.Mandelbrod, *Component Ranking and Automatic Query Refinement for XML Retrieval*, In: Advances in XML Information Retrieval, LNCS 3493, INEX 2004. December 2004.

[4] Y.Mass and M.Mandelbrod, *Using the INEX Environment as a Test Bed for various user Models for XML Retrieval*, In : *LNCS 3977, INEX 2005, Dagstuhl Germany*, pg. 187-195. November 2005

[5] T.Grabs and H J.Scheck, *Flexible information retrieval from XML with Power DB XML*, In : proceedings of the first annual workshop of INEX, pages 141-148, December 2002

[6] V.Kakade and P. Raghavan, *Encoding XML in vector spaces*, In: Proceedings of ECIR 2005, Saint Jacques de Compostelle, Spain.

[7] N.Fuhr N and S.Malik and M.Lalmas, *Overview of the initiative for the evaluation of XML retrieval (INEX) 2003,* In: Proceedings of INEX 2003 Workshop, Dagstuhl, Germany. Decmber 2003

[8] P.Ogilvie and J.Callan, *Using language models for flat text queries in XMLretrieval*, In: Proceedings of INEX 2003 Workshop, Dagstuhl, Germany, pages 12–18. December 2003.

[9] J.Kamps and M.Rijke and B.Sigurbjornsson, *Length normalization in XML retrieval*, In: Proceedings of SIGIR 2004, Sheffield, England, pages 80–87

[10] B.Piwowarski and GE.Faure and P.Gallinari, *Bayesian Networks and INEX*, In: proceeding in the first annual workshop for the evaluation of Xml Retrieval(INEX),2002

[11] Vo Ngoc Anh, Alistair Moffat, *Compression and an IR approach to XML Retrieval*, In: INEX 2002 Workshop Proceedings, p. 100-104, Germany, 2002.

[12] Karen Sauvagnat, *Modèle flexible pour la recherche d'information dans des corpus de documents semi-structurés*. Thèse Doctorat, Université Paul Sabatier de Toulouse, 2005.

[13] N. Fuhr, K. Grossjohann, *XIRQL, a query language for information retrieval in XML documents*, In: proceedings of SIGIR 2001, Toronto Canada 2001.

[14] N. Gövert, M. Abolhassanni, N. Fuhr, K. Grossjohann, *Content-Oriented XML Retrieval with HyreX*, In: INEX 2002 Workshop Proceedings, p. 26-32, Germany, 2002.

[15] H.cui, J-R.Wen, J-R.Chua, *Hierarchical indexing and flexible element retrieval for structured document*. April 2003.

[16] M.Ben Aouicha, *Une approche algébrique pour la recherche d'information structurée*. Thèse de doctorat en informatique, Université Paul Sabatier, Toulouse, 2009.

[17] S.Berchiche-Fellag and M.Boughanem, *Traitement des requêtes CO (Content Only) sur un corpus de documents XML*, In : Colloque sur l'Optimisation et les Systèmes d'Information, 2010.

[18] L.Denoyer and P. Gallinari, *The Wikipedia XML corpus*, In: SIGIR Forum 40(1), pp. 64–69, 2006