A Universal Model for XML Information Retrieval

Maria Izabel M. Azevedo[1], Lucas Pantuza Amorim[2], and Nívio Ziviani[3]

[1] Department of Computer Science, State University of Montes Claros,
Montes Claros, Brazil
izabel@dcc.ufmg.br
[2] Department of Computer Science, State University of Montes Claros,
Montes Claros, Brazil
lucaspantuza@yahoo.com.br
[3] Department of Computer Science, Federal University of Minas Gerais,
Belo Horizonte, Brazil
nivio@dcc.ufmg.br

Abstract. This paper presents an approach for extending the vector space model (VSM) to perform XML retrieval. The model is extended to support important aspects of XML structural and semantic information such as element nesting level, matching tag names in the query and the collection and the relation between tag names and content of an element. Potential use of the model for heterogeneous as well as for the unstructured collection is also shown. We compared our model with the standard vector space model and obtained a gain for unstructured and structured queries. For unstructured collections the vector space model effectiveness is preserved.

1 Introduction

Studying the structure of a XML [3] document we can observe special aspects on its information organization: the hierarchical structure corresponding to the nesting of elements in a tree and the presence of *markups that describes their content* [1]. The first one is important for information retrieval because the words on different levels of the XML tree may have different importance for expressing the information content in the whole tree. Moreover, if markup describes its content, it must have been conceived semantically related to the information it delimits. This makes the second aspect especially important.

Another important aspect on XML documents is that it introduces a new retrieval unit. We do not have only documents and collections anymore. Now we have elements that can be inside of another element and also contain many others. Consequently, the unit of information to be returned to users can vary. If one element satisfies a query so its ancestor or descendant may also satisfy. Besides, with XML documents, the user can propose queries that explore specific elements.

In the XML environment there are two types of queries, those with structural constraints, called CAS (Content and Structure), and those without constraints called CO (Content Only) [5]. In this paper we propose an extension of the vector space model [8] that considers both aspects (nested structure and markup that describes content) of

N. Fuhr et al. (Eds.): INEX 2004, LNCS 3493, pp. 311 – 321, 2005.

XML structure. This will be done in order to improve the vector space model result for CO and CAS queries, processing retrieval units of varying lengths.

Although the extended model has been conceived to explore the semantic relation between XML markups and content of an element, we demonstrate that it can be applied to non-XML documents. In this case, the vector space model effectiveness will be preserved. It can also be applied to homogeneous collections, where homogeneous structures do not always allow appropriate semantic relation between markups and content of an element. Consequently, our model has universal application, achieve in complete automatic process.

The rest of the paper is organized as follows: Section 2 describes the extension to the vector space model presenting the factor that will explore XML characteristics. Section 3 shows the model applications to different collections. Section 4 presents the results and section 5 concludes the paper.

2 XML Retrieval Using the Vector Space Model

In this section, we introduce the retrieval model. A retrieval unit is defined and also the new *fxml* factor, used to explore XML characteristics.

2.1 Retrieval Units

The first challenge we face when studying XML Information Retrieval is what will be the ideal retrieval unit to be returned to the user, the one that best solve his information needs. In one XML document, there are many possibilities: we can return a whole document or any of its sub-elements. But, what is the best one? The answer depends on the user query and the content of each element. Related work, as [2] and [6], index pre-defined elements as retrieval units.

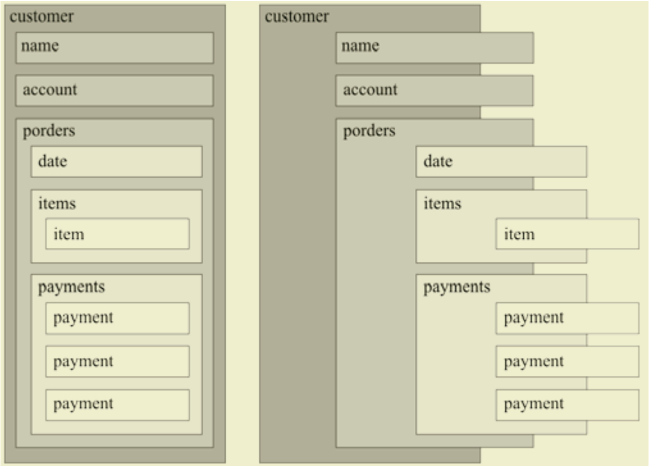


Fig. 1. Retrieval units

In our model we evaluate all possible combinations of elements for each query. To make it possible we should index all components as an information unit. In the example of Fig. 1, our original collection will be expanded from 1 document and 10 subelements to 11 retrieval units, each one with an entry in the inverted list. The statistic of each element will consider all the text inside of its sub-elements.

2.2 Vector Space Model Applied to XML Retrieval

From the vector space model, we have that the relevance of document D to query Q, $\rho(Q,D)$, is given by the cosine measure [7]:

$$\rho(Q,D)=\sum ti\in Q\cap D\frac{w_Q(ti)*w_D(ti)}{\|Q\|*\|D\|} \tag{1}$$

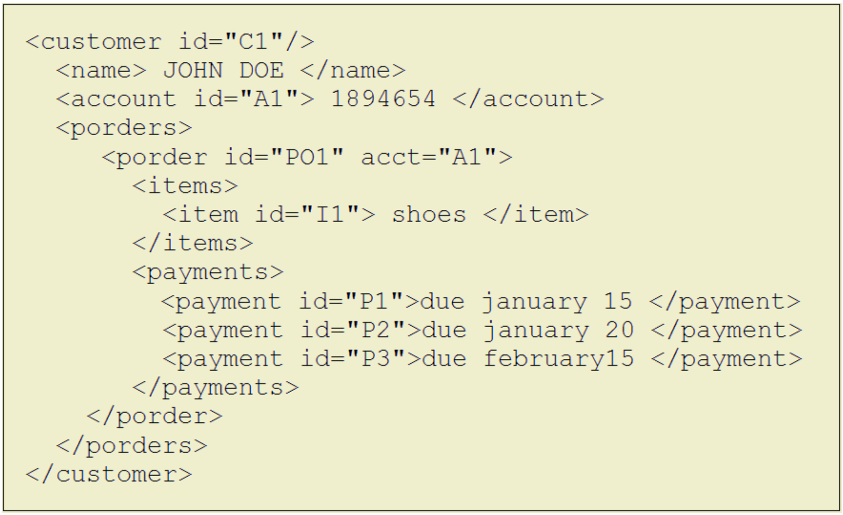
where,

- ti is a term in the collection;
- $W_Q(ti)$ is the weight of the query and does not affect the ranking;
- $W_D(ti)$ is the weight of the document and is given by

$$w_D(ti)=log(tf(ti))*log(N/df(ti)). \tag{2}$$

To adapt this model to XML documents we introduce changes that express their characteristics, as follows. First, we will consider each element as a new retrieval unit. $tf(ti)$ is taking for each of them, becoming $tf(ti,e)$, so:

- $tf(ti,e)$ is the number of occurrences of a term ti in a element e;
- $df(ti)$ is the number of elements that contain ti;
- N is the total number of elements in the collection.



```
<customer id="C1"/>
  <name> JOHN DOE </name>
  <account id="A1"> 1894654 </account>
  <porders>
     <porder id="PO1" acct="A1">
       <items>
         <item id="I1"> shoes </item>
       </items>
       <payments>
         <payment id="P1">due january 15 </payment>
         <payment id="P2">due january 20 </payment>
         <payment id="P3">due february15 </payment>
       </payments>
    </porder>
  </porders>
</customer>
```

Fig. 2. One XML Document

Now, we consider the nested structure of a XML document. One term will be counted for elements where it appears in textual content and for the ancestor elements of *e*, as showed bellow for the example XML document in Fig. 2.

Thus:

- *tf(*`january, payment id = "P1"`*)* = 1;
- *tf(*`january, payment id = "P2"`*)* = 1;
- *tf(*`january, payments`*)* = 2;
- *df(*`january`*)* = 3.

So *ρ(*`january,payments`*)* will be greater than *ρ(*`january,payment`*)*. As elements in a higher level in the XML tree include the contents of sub-elements, *tf(ti,e)* will increase. This problem has been treated in [2], using the concept of augmentation. We will apply this idea, using a factor (*fnh*) that will reduce the weight of terms contribution depending on their position on XML tree as explained later on.

2.3 Retrieval Model Extension

At this point, we have already defined how the standard vector space model statistics will be calculated, just adapting them to the new retrieval unit, present on XML documents. In the following, we will describe one new factor (*fxml*), which will explore XML characteristics pointed in the introduction, assigning different weights for each term. Thus:

$$\rho(Q, D) = \sum ti \in Q \cap D \frac{w_Q(ti) * w_D(ti, e) * fxml(ti, e)}{\|Q\| * \|D\|} \tag{3}$$

where,

$$fxml(ti,e) = fnh(ti,e) * fstr(ti,e) * focr(ti,e) \cdot \tag{4}$$

The Nesting Factor, denoted by fnh, expresses the relevance of terms considering their position on the XML tree, and is given by:

$$fnh(ti,e) = 1/(1 + nl) \tag{5}$$

where,

- *nl* is the number of levels from element *e* to its sub-element containing term *ti*.

The nesting factor can vary between the following two values:

- *fnh(ti,e)* = 1, for terms directly in element *e*,
- *fnh(ti,e)* = 1/*nd*, *nd* being the depth of the XML tree.

This factor will reduce the term contribution for distant elements (upwards) in XML tree.

The Structure Factor, denoted by *fstr*, expresses how the query structural constraints are satisfied by the context[1] of an element and is given by:

$$fstr(ti,e) = (common\_markups + 1)/(nr\_qmarkups + 1) \tag{6}$$

where,

- *common_markups* is the number of markups present in the query structural constraints and also in the context of element *e* that contains *ti*;
- *nr_qmarkups* is the number of markups in the query structural constraints.

The structure factor can vary from the following to values:

- *fstr(ti,e)* = 1/(*nr_qmarkups*+1), when no structural constraints appears in the context of *ti*,
- *fstr(ti,e)* = 1, when all query's structural constraints markups appears in the context of *ti*.

This factor will valorize a context that better satisfies structural constraints present in the query. It is important on the CAS query, where users express elements that will better fit their information need. For CO queries it will be equal to 1, and will not influence the relevance equation.

The last factor, Co-occurrence Factor, denoted by *focr*, expresses the semantic relation between markups and their content, and is given by:

$$focr(ti,e) = cf(ti,e) * idf(ti) * N * icf(e) \tag{7}$$

where,

- *cf(ti,e)* is the number of times the markup of element *e*, denoted by *m*, delimits a textual content containing term *ti*. In other words, number of co-occurrences of term *ti* and markup *m* in the collection;
- *idf(ti,e)* is the inverse of the number of elements *e* that contain *ti*;
- *icf(e)* is the inverse of the number of times markup *m* appears in the collection;
- *N* is the total number of elements in the collection.

Then, *cf(ti,e)*idf(ti,e)*, is the reason between the number of times term *ti* appears with *m* for the numbers of elements containing *ti* in the collection. And *icf(e)*N*, express the popularity of markup *m* in the collection. So, the co-occurrence factor takes into account the co-occurrence of terms and markups, considering the popularity of markups. Concluding, the XML factor (*fxml*) explores XML characteristics looking for the semantic of terms, looking for information behind words.

3 Applications of the Extended Model

In this section we show the application of our model in unstructured, homogeneous and heterogeneous collections, analyzing the fxml factor.

[1] Context is the element position on the XML tree, represented by its complete path, from the root to the element containing the textual content.

3.1 Non-XML Documents

Considering that a real world collection may contain XML documents and non-XML documents, we will demonstrate that the same model can be applied on those documents, preserving the vector space model effectiveness.

Examining Formula 3, we conclude that to satisfy this condition, XML factor must be equal to 1. To demonstrate it, we will consider that the whole content of a non-XML document will be delimited by one special markup, for example `<article>` and `</article>`. This special markup will convert a non-XML document in a XML document, with only one element. So it will be processed as any other XML document, but XML factor will be equal to 1.

Next we will analyze each of the three *fxml* factors, for non-XML documents.

Fnh. In one non-XML document there is only one level where all textual content is, so *nl* = 0 and *fnh(ti,e)* will be equal to 1.

Fstr. For a non-XML document, the numerator will be equal to 1 because it has no markup, the denominator will depend on the query type. Thus:

- For CO: *q_markups* = 0 and *fstru(ti,e)* = 1;
- For CAS: *nr_qmarkups* may vary depending on the number of structural constraints in the query.

One non-XML document will never satisfy the CAS query structural constraints because it is not structured, then its relevance will be decreased compared with those that can satisfy query constraints.

Focr. For one non-XML:

- *cf(ti,e)*, the number of times *ti* appears with markup *m*, will be the number of times *ti* appears in the collections because all documents have the same special markup `<article>` and only this markup. So *cf(ti,e)* is the inverse of *idf(ti)*, making *cf(ti,e)*idf(ti)* = 1;
- *icf(e)*, the inverse number of times markup *m* appears in the collection, will be equal 1/*N*, the number of documents in the collection, because all documents in the collection will have the same special markup `<article>`. So *N*icf(e)* will be equal 1, making *focr(ti,e)* =1.

Non-XML documents are a special case and the model will converge to the vector space model.

3.2 Homogeneous Collections

A homogeneous collection is defined as a collection where all documents have the same DTD. In this section we will analyze the implications it has on our model. We now discuss each of the three *fxml* factor, for homogeneous Collections.

Fnh. This factor will affect the relevance of elements, reducing the term contribution for distant elements (upwards) in XML tree.

Fstr. This factor will be analyzed only for CAS queries because for CO queries it will be always 1, as stated in Section 4.

As all documents have the same DTD, they have the same elements and so *fstr(ti,e)* will be equal for all elements. Any document will have the same probability to have an element return to user. Within one document, those elements with more similarity with the query structural constraints will have greater relevance. Also they will have better chance to be returned to the user.

Focr. In homogeneous collections, all documents have the same markups, and not always there will be an appropriated semantic relation between markups and the content of an element. Examining INEX [4] homogeneous collections, for example, we observe that its markups describe information structure (<p>, <bdy>, <fm>, <bm>) rather than information content. So this factor, will probably not affect much the relevance ranking.

3.3 Heterogeneous Collection

A heterogeneous collection is defined as a collection where documents may have different DTD. Our Model does not use any information that comes from DTD. It just indexes elements, terms and markups, collecting statistics that measure the relation between them, so we do not need to make any change in dealing with heterogeneous collections. But it is important to analyze how the heterogeneity of markups will influence the relevance ranking of our model. We now discuss each of the three *fxml* factor, for heterogeneous Collections.

Fnh. This factor will affect the relevance of elements, reducing the term contribution for distant elements (upwards) in XML tree.

Fstr. As stated before, this factor is always 1 for CO query. So let us analyze CAS queries. CAS queries impose a structural constraint, and will have greater relevance to those elements that satisfy them. So, documents with DTDs similar to the structure of the query will be ranked first. If a user asks for:

```
//article[about(.//author, John Smith)],
```

one element as:

```
<author> John Smith </author>,
```

or even

```
<author>
   <first name> John</first name>
   <last name> Smith</last name>
</author>,
```

will be better ranked than one as

```
<title> John Smith Biography </title>.
```

The reason is because the first two have markups present in the structure constraint of the query and the third has not. This will come across the information need of the user.

But one element with a markup <author> will be better ranked than one with <writer> on its DTD. It happens because both markups have the same semantic, but they are not equal and only <author> is present in the structure constraint of the query. It will affect the ranking, but will not avoid that the element with <writer> can be returned to the user.

Focr. This factor tries to explore the fact that markups describe their content. Considering that in a heterogeneous collection different DTD will allow better relation between each document structure and its content, it will help to explore different meanings of the same words in different contexts.

But here appears the following language problem: which markup is semantically closer to John Smith, <author> or <writer>? <author> or <autor> ?

Factor *focr* also ponders the frequency of markups in the collection by *N*icf(e)*. So, if <author> is more widespread than <writer> and <autor>, it will have more chance to appear with John Smith. But it will be compensated by *icf(e)*, in a similar way that common terms in one collection will be reduced by *idf(t)*, in standard vector space model.

Concluding, our model can deal with heterogeneous collections, to answer unstructured or structured queries. The model uses statistical measures of markups and terms, and do not need to map the structure of one DTD onto the others.

4 Results

The proposed model was run over the homogeneous and heterogeneous INEX collections [4]. For the homogeneous collection, the effect of each term of *fxml* factor was observed.

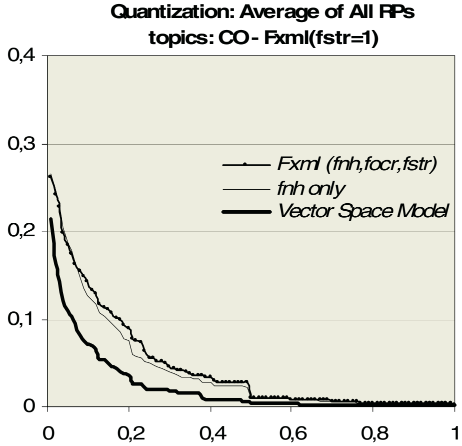


Fig. 3. Recall/Precision Curves comparing Vector Space Model and Adapted Model

The *fnh* factor as expected, improves considerably the vector space model effectiveness as expected, as shown in Fig. 3. Upwards elements accumulate all sub-elements contribution and without this consideration many of them would have been better ranked than more important sub-elements.

We also compared different values of Fnh, concluding that when it changes for elements in different levels of a XML tree, then precision improves a little bit. Subsequently, we introduced the Focr factor and observed a small improvement, observe the small improvement as shown in Fig. 4, which can be imputed to the fact that in homogeneous collections this factor will not vary much, because all documents have the same structure.

For CAS queries the factor Fstr was introduced and also caused some improvement as shown in Fig. 5.

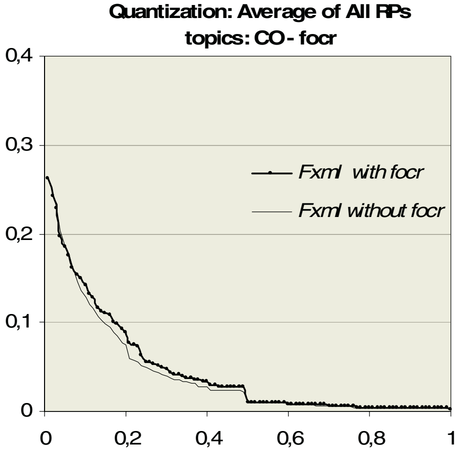


Fig. 4. Recall/Precision Curves changes with *Focr*

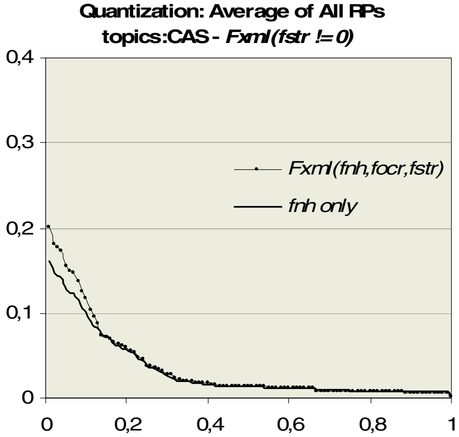


Fig. 5. Recall/Precision Curves changes with *Fstr*

CAS queries results are worse than CO results. It raises a question: can the structure constraint of a query help to improve the precision of the results? To answer this question we should compare the results of CAS queries using fstr, which uses structure information to improve performance, with the results without it, for the same set of queries. Fig. 5 confirms that structure constraint of a query improves precision. So, why CAS queries results are worst? Maybe because the CAS queries express a more specific information need, and receive more strict assessment. But this question remains open.

We submitted runs to the INEX Iniciative for heterogeneous collections, but as its assessments were not concluded, we have no Recall/Precision Curves. It follows a sample of an answer to a query showing results from many sub-collections, confirming that our model can deal with different DTDs.

For query:

```
//article[about(.//author, Nivio Ziviani)],
```

we get the following answer:

```
<topic topic-id="2"> ...
<result>
  <subcollection  name="ieee" />
  <file>co/2000/ry037</file>
  <path>/article[1]/fm[1]/au[1]</path>
  <rank> 3</rank>
</result> ...
<result>
  <subcollection  name="dblp" />
  <file>dblp</file>
  <path>/dblp[1]/article[177271]/author[4]</path>
  <rank> 6</rank>
  </result> ...
<result>
  <subcollection  name="CompuScience" />
  <file>exp-dxf1.xml.UTF-8</file>
  <path>/bibliography[1]/article[23]/author[1]</path>
  <rank> 30</rank>
</result>
 ...
<result>
  <subcollection  name="hcibib" />
```

```
    <file>hcibib</file>
    <path>/file[1]/entry[229]/article[1]/author[1]</path>
    <rank> 139</rank>
  </result>
```

5 Conclusions and Future Work

We have shown a universal model for dealing with information retrieval on XML documents. It can be applied to non-XML documents, homogeneous and heterogeneous collections, to answer structured (CAS – content and structured) and no-structured (CO – content only) queries. The major contribution of this work is its universal application, achieved in a completely automatic process.

All introduced factors behave as expected and our results are close to the average of other INEX participants. The average precision stays around 0.05 for CO and 0.04 for CAS queries and needs to be improved, demanding further investigation. *Fstr* factor should be better adjusted to query constraints. For an appropriated assessment of *Focr* factor it would be better to have a real heterogeneous collection, with documents from different knowledge areas, as biology, geography, etc., and including XML documents originated from databases.

References

1. S. Abiteboul, P. Buneman and D. Suciu. *Data on the Web – From Relations to Semistructured Data in XML*. Mogan Kaufmann Publishers, San Francisco, California, 2000, pp. 27-50.
2. M. Abolhassani, K. Grobjohann and N. Fuhr. Content-oriented XML Retrieval with HyREX. In *INEX 2002 Workshop Proceedings*, Duisburg, 2002, pp.26-32.
3. T. Bray, J. Paoli, C. M. Sperberg-McQueen and E. Maler. *Extensible Markup Language (XML) 1.0.* 2nd ed. http://www.w3.org/TR/REC-xml, Cct 2000. W3C Recommendation 6 October 2000.
4. N. Fuhr and M. Lalmas. *INEX document Collection*. http://inex.is.informatik.uni-duisburg.de:2004/internal/, Duisburg, 2004.
5. G. Kazai, M. Lalmas and S. Malik. INEX'03 Guidelines for Topic Development. In *INEX 2003 Workshop Proceedings*, Duisburg, 2003 pg. 153-154.
6. M. Mandelbrod and Y. Mass. Retrieving the most relevant XML Components. In *INEX 2003 Workshop Proceedings*. Duisburg, 2003, pp. 58-64.
7. B. Ribeiro-Neto e R. Baeza-Yates. *Modern Information Retrieval*. Addison Wesley, 1999, pp. 27-30.
8. G. Salton e M. E. Lesk. *Computer evaluation of indexing and text processing*. Journal of the ACM. 15(1), 1968, pp. 8-36.