IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITP4291

System of Information Retrieval in XML Documents

Saliha Smadhi
Laboratoire d'Informatique, Universite de Pau, Saliha.Smadhi@univ-pau.fr

ABSTRACT

The Extensible Markup Language (XML) is considered as a new standard for data representation and exchange on the web. XML opens opportunities to develop a new generation of information retrieval system (IRS) to improve the interrogation process of document bases on the Web.

We propose an approach to retrieve units (or subdocuments) of relevant information from XML documents. Our work focuses instead on end-users who have not expertise in the domain and of that the structure is them unknown (like a majority of the end-users). This approach supports keywords based searching like classical IRS and integrates structured searching with the search attributes notion.

It's based on an indexing method of document tree leafs which authorize so a content-oriented retrieval. The retrieval subdocuments are ranked according to their similarity with user's query. We use an similarity measure which is a compromise between two measures : exhaustiveness and specificity.

INTRODUCTION

The World Wide Web (WWW) contains huge amounts of information that is available at web sites, but it's difficult and complex to retrieve pertinent information. Indeed, a large part of this information is often stored as HTML (HyperText Markup Language) pages that are only viewed through a web browser.

This research is developed in the context of the MEDX project (Lo & al, 2001) of our team. We use XML as a common structure for storing, indexing and querying a collection of XML documents.

Our aim is to propose the suited solutions which allow to the end-users not specialist of the domain, to search and extract portions of XML documents (called units or subdocuments) which satisfy their queries.

The extraction of documents portion can be realized by using XML query languagues (XQL, XML-QL, …) (Robie, 1999), (Deutsch & al., 1999).

An important aspect of our approach concerns the indexation which is realized on leaf elements of the document tree and not on the whole document.

Keywords are extracted from domain thesaurus. A thesaurus is a set of descriptors (or concepts) connected by hierarchical relations, equivalence relations or association relations. Indexing process results are stored in a ressources global catalog that is exploited by the search processor.

This paper is organized as follows. Section 2 discusses the problematic of relevant information retrieval in context of XML documents. In section 3, we present the model of XML documents indexing. The section 4 presents the similarity measure adopted and the retrieval strategy of relevant parts of documents. Section 5 discusses related work and concludes the paper. An implementation of SIRX prototype is currently under way in Python language on Linux Server.

INFORMATION RETRIEVAL AND XML DOCUMENTS

The classical retrieval information involves two principal issues, the representation of documents and queries and the construction of a ranking function of documents.

Among IR models, the most models used are the boolean model, vector space model and probabilist model. In the vector space model, documents and queries are represented as vectors in the space of index terms. During the retrieval process, the query is also represented as a list of terms or a term vector. This query vector is matched against all document vectors and a similarity measure between a document and a query is calculated. Documents are ranked according to their values of similarity measure with a query.

XML is a subset of the standard SGML. It has a richer structure that it's composed mainly of a elements tree that forms the content. XML can represent more useful information on data than HTML. An XML document contains only data as opposed to an HTML file, which tries to mix data and presentation and usually ignores structure. It preserves the structure of the data that it represents, whereas HTML flattens it out. This meta markup language defines its own system of tags representing the structure of a document explicitly. HTML presents information and XML describes information.

A well-formed XML document doesn't impose any restrictions on the tags or attribute names. But a document can be accompanied by a DTD (Document Type Definition) which is essentially a grammar for restricting the tags and structure of a document. An XML document satisfying a DTD is considered as valid document.

The Document Object Model (DOM) is simply a set of plans or guidelines that enables the user to reconstruct a document right down to the smallest detail.

The structure of a document can be transformed with XSLT (XSLT, 1999) and its contents displayed by using the XSL (eXtensible Style Language) language or a programming language (python, java,…). XSL is a declarative language which model refers the data by using patterns. It is limited where one wants retrieve data with specific criteria as one can realize that with the query language XQL (or OQL) for relational databases (or objects). This extension is proposed by :

- two languages coming from the database community : XML-QL (Florescu & al., 2000), Lorel (Abiteboul & al, 1997)
- XQL (Robie, 1999) from the Web community.

Requirements for a System of Relevant Information Retrieval for XML Documents

We propose an approach for information retrieval with relevance ranking for XML documents of which the basic functional requirements are :

- To support keyword based searching and structured searching (by proposing a set of search attributes) by end-users who have no expertise in the domain and of that the structure is them unknown (like a majority of the end-users);
- To retrieve a relevant parts of documents (called subdocuments) ranked by their relevancy with the query;
- To navigate in the whole document.

In order to satisfy the essential requirements of this approach, we have opted for :

a – using of domain thesaurus,

b – definition of an efficient model of documents indexing that extend the classic "inverted index" technology by indexing document structure as well as content.

This paper appears in *Issues and Trends of Information Technology Management in Contemporary Organizations,* the proceedings of the Information Resources Management Association International Conference. Copyright © 2002, Idea Group Inc.

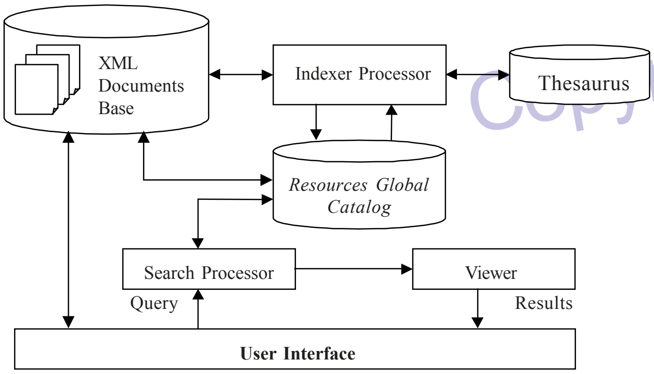
c – integration of search attributes that concern a finite number of sub-structures types, which like to make searchable.

d – propose an information retrieval engine with ranking of relevant document parts.

Architectural Overview of SIRX

We present an overview of the **S**ystem of **I**nformation **R**etrieval in **X**ML documents (SIRX) showing its mains components (Figure 1).

Figure 1: The general architecture of SIRX



The main architectural components of the system are the following:

1 - User interface : It' s used to facilitate the interaction between the user and the application. It allows the user to specify his query. It displays also retrieved documents or parts of documents ranked by relevance score. It does'nt suppose an expertise or an domain knowledge of the end-user.

2 - Search processor : Allows to retrieve contents directly from Resources Global Catalog on using the various index and keywords expressed in input query .

3 - XML documents base : stores XML documents well-formed in their original formats.

4 - Thesaurus : The domain thesaurus contains the set of descriptors (keywords) which allow to index documents of this domain.

5 - Indexer processor : for every XML document, the indexer processor creates indexes by using the thesaurus and the XML documents base. These indexes allow to build the resources global catalog.

6 – Resources Global Catalog : It's an indexing structure that search processor uses to find the relevant document parts. It is exploited mainly by the search processor.

7 - Viewer: Displays retrieved document parts . The results are recombined (XML + XSL) to show the document to the user in an appropriate manner into HTML.

THE MODEL OF XML DOCUMENTS INDEXING

In our approach that is based on vector space model, we propose to index the leafs of the document tree (Shin & al, 1998) the keywords that correspond to the descriptor terms extracted from domain thesaurus (Lo & al., 2000). Indexing process results are structured by using the XML language in meta-data collection which is stored in *"resources global catalog"* (Figure 2) This catalog is the core of the SIRX system. It encapsulates all semantic content of XML documents base and thesaurus.

Elementary Units and Indexing

In classic information retrieval, the documents are considered as atomic units. The keyword search is based on a classic index structures that are inverted files. An classic inverted file contains <keyword, document> pairs meaning that the word can be found in the document. This classical approach allows to retrieve the whole document. It is not necessary to forget that documents can often be quite long and in many cases only a small part of documents may be relevant to the user's query. It is so necessary to be able to retrieve only the part of document may be relevant to the end-user's query.

To accomplish this objective, we extend the classic inverted file by making unit structure explicit. The indexing processor extracts terms from thesaurus and calculates their frequencies in each element at the text level.

Every elementary unit is identified in an unique way by a *access-path* showing his position in the document. The form of this index is <keyword, unit, frequency>

where :

(i) keyword is a term appearing in the content of element or values of an attribute of this document

(ii) unit specifies the access path to element content that contains keyword. The access path is described by using XPath (Xpath, 1999) compliance syntax.

(iii) frequency is the frequency of the keyword in the specfied unit.

This indexation method allows a direct access to any elementary unit which appears in the result of the query and regroups results of every document by using XSLT.

Search Attributes

Methods of classical information retrieval propose a function of search from signaletic metadata (author, title, date, …) that concerns mostly characteristics related to whole document. To be able to realize searches on sub-structures of a document, we propose to integrate a search based on the document structure from a finite number of element types, which like to make searchable from their semantic content. These specific elements are called *search attributes*. They are indexed like keywords, in resources global catalog. Every search attribute has the following form : <identifier, unit> where identifier is the name (or tag) of the search attribute under which it will appear to the user and unit indicates the access path to a elementary unit (type 1) or an another node (type 2) of document on that will carry this structural search based on it content. Search attributes names are available at the level of user's interface.

In the following example, the tag of elementary unit is 'title' and 'author' is the name of an attribute of the tag 'book'.

```
<info idinfo="title"  path="//title"/>
<info idinfo="author" path="//book/@author"/>
```

The query result depends on type of search attribute.

If the indexed search attribute is an elementary unit then the returned result is the node that is the father of this unit.

If the indexed search attribute is a node different from elementary unit then the returned result is this node.

Queries examples :

Query 1 : title = 'dataweb'. This query returns following result : all the names of documents of which value of <title> contains 'dataweb' text.

Query 2 : author = 'smadhi'. This query returns following result : all the sub-structures (at first level) which have for name 'book' and for that the attribute 'author' contains 'smadhi' text.

Resources Global Catalog

The resources global catalog is defined as a generalized index that allows to maintain for SIRX, to efficiently support keyword searching and sub-structure searching. It i's used by search processor use to find the relevant documents (or parts of documents).

It's represented by an XML document which describes every XML document that is indexed by indexing processor. This catalog is described in XML according the Figure 2.

Figure 3 illustrates the structure of this catalog.

Keyword Weights

In vector space model, documents and queries are represented as vectors weighted terms (the word term refers to keyword) (Salton &

Figure 2: The catalog DTD

```
<!ELEMENT catalog(doc*)>
<!ELEMENT doc(address, search-attributes, keywords)>
<!ATTLIST doc iddoc ID #REQUIRED>
<!ELEMENT search-attributes(info*)>
<!ELEMENT info (#PCDATA)>
<!ATTLIST info idinfo ID #REQUIRED)
<!ATTLIST info path CDATA #REQUIRED>
<!ELEMENT address(#PCDTA)>
<!ELEMENT keywords(key*)>
<!ELEMENT key (#PCDATA)>
<!ATTLIST key idkey ID #REQUIRED>
<!ATTLIST key path CDATA #REQUIRED>
<!ATTLIST key freq CDATA #REQUIRED>
```

Figure 3: An example of resources global catalog

```
<catalog>
<doc iddoc="d1" >
   <address>c:/SRIX/mcseai.xml</address>
   <search-attributes>
         <info idinfo="title"  path="//title"/>
         <info idinfo="author" path="//book/@author"/>
   </search-attributes>
   <keywords>
         <key idkey ="k1" path="//dataweb/integration"
   freq=2>xml </key>
         <key  idkey  ="k2"  path="//  mapping/@base"
   freq=1>xml </key>
         …
   </keywords>
</doc>
<doc iddoc="d2" >
   <address>c:/SRIX/cari2000.xml</address>
   <search-attributes>
         <info idinfo="title"  path="//title"/>
         <info idinfo="author" path="//book/@author"/>
   </search-attributes>
   <keywords>
         <key idkey ="k25" path="//architecture/integra-
   tion" freq=2>web </key>
         <key  idkey  ="k26"  path="// architecture/inte-
   gration" freq=2>dataweb </key>
         …
   </keywords>

</doc>
….
</catalog>
```

al, 1988), (Yuwono & al. 1996) . In our approach each indexed elementary unit j of document i is represented by a vector as follows :

$$U_j^i = (w_{j1}^i, w_{j2}^i, ,..., w_{jk}^i, ..., w_{jp}^i)\ ,\ k = 1, 2, ...,\ p$$

- *nu* : number of elementary units *j* of document *i*
- *p :* number of indexing keywords
- w^i_{jk}: weight of the *k*th term in the *j*th elementary unit of the *i*th document

We use the classical *tf.idf* weighting scheme (Salton et al., 1988) to calculate w^i_{jk} .

$$w^i_{jk} = tf^i_{jk} \times \ idf_k$$

- $tf^{\,i}_{jk}$: the frequency of the *k*th term in the *j*th elementary unit of the *i*th document
- idf_k : the inverse document frequency of the index term *tk*. It is computed as a function of the elementary unit frequency by the following formula :

$$idf_k = log(tnu/nu_k)$$

- *tnu* : the total number of elementary units in the document base
- nu_k : the number of elementary units which the *k*th term occurs at least once.

RELEVANT INFORMATION RETRIEVAL

SIRX supports two ways to retrieve parts of documents:

a) Querying by search attributes; it authorizes a search based on a document structure from a list of search attributes proposed to user. It allows to retrieve documents or parts of documents according the type search attributes (see § 3.2) . This aspect is not detailed in this paper.

b) Querying by content with keywords; It allows to retrieve documents or parts of documents.

In this section we describe the search process of relevant information retrieval that involve two issues : generation of query vector and computing the similarity between vector query and each elementary unit vector.

The adopted model of data rests mainly on the use of the catalog in memory central for an exploitation, during the process of interrogation by as set of end-users.

Query Processing

A user'query is a list of one ore more keywords which belong to the thesaurus. When the user input a query, the system generates a query vector by using the same indexing method as that of the element unit vector. A query vector *Q* is as follows:

$Q = (q_1,\ q_2,\ ...,\ q_k,\ ...,\ q_m)$ with *m£<p*

Query terms q_k (*j=1…m*) are weighted by the *idf* value where *idf* is measured by $log(tnu/nu_k)$.

Retrieval and Ranking of Relevant XML Information Units

The search process returns the relevant elementary units of an XML document. These information units are ranked according to them similarity coefficients measuring the relevance of elementary units of an XML document to a user's query.

In the vector space model, this similarity is measured by cosine of the angle between the elementary unit vector and query vector.

On considering the two vectors $\boldsymbol{U_i}$ and $\boldsymbol{Q}$ in the euclidean space with scalar product noted <,> and norm noted ||.||, the similarity is (Smadhi 2001):

$$Sim\,(U_i, Q) = \frac{\sum_{j=1}^{m} q_j w_{ij}}{\sqrt{\sum_{j=1}^{m} q_j^2}\sqrt{\sum_{j=1}^{m} w_{ij}^2}}$$

This measure as others (Salton & al., 1988), (Wang & al. ,1992) are based on the following hypothesis: more a document looks like the query more they susceptible to be relevant for the user. We question this hypothesis because query and the document do not play a symmetric role in the search for information (Simonnot & Smail, 1996), (Fourel, 1998). It is necessary to note that the user expresses in his query only characteristics of the document which interests it at the given moment. It is necessary to take into account two important criteria: the exhaustiveness of the query in the document and the specificity of the document with regard to the query (Nie, 1988).

Now, we show how to spread this measure of similarity to take into account these two criteria.

A measure is based on the exhaustiveness if it estimates the degree of inclusion of the query Q in the unit U_i. Conversely, a measure based on the specificity measures the degree of inclusion of U_i elementary unit in the query Q.

We propose the two following measures :

a) The exhaustiveness measure noted mexh

$$mexh(U_i, Q) = \frac{\cos(U_i, Q)\|U_i\|}{\|Q\|}$$

b) The specificity measure noted mspec

$$mspec(U_i,Q) = \frac{\cos(U_i,Q)\|Q\|}{\|U_i\|}$$

These two measures have intuitively a comprehensible geometrical interpretation because mexh(Ui,Q) represents the norm of the vector projection Ui on the vector Q. In dual way, mspec(Ui,Q) represents the norm of vector projection Q on the Ui vector. Then similarity measure became :

$$Sim(U_i,Q) = \sqrt{mspec(U_i,Q)mexh(U_i,Q))}$$

Experiments Results

The reference collection that we built is not very important. This collection has 200 XML documents which correspond to articles extracted from proceedings of conference. First estimates seem to us very interesting: the measure of similarity that we proposed allowed us to improve about 20 % the pertinence of restored subdocuments. These tests are realized on Linux Server using Dell computer with 800Mhz Intel processor with 512 MB RAM.

RELATED WORK AND CONCLUSION

Many works are done to propose methods of information retrieval in XML documents. Among various approaches (Luk & al. 2000), database-oriented approach and information retrieval-oriented approach seems the most used.

In database-oriented approach some query language, like XIRQ (Fuhr & al. 2000), XQL, XML-QL are proposed but these languages are not suitable for end-users in spite of the intregation of a keyword search into XML quey language (Florescu & al., 2000). Xset (Zhao &al., 2000) suppose to have knowledge about document structure. If XRS (Shin & al. 1998) propose an interesting indexing method at the leaf elements but it presents an inconvenience with the use of DTD.

Our approach propose, like XRS, an indexing at the leaf elements and it extends inverted index with XML path specifications. It takes into account also the structure of XML document. Moreover we introduce a particular measure of similarity which is an compromise between two measures : exhaustiveness and specificity.

This new approach allows to retrieve parts of XML documents with relevance ranking.

REFERENCES

Abiteboul S., Quass D., McHugh D., Widom J., and Wiener,J. (1997). The Lorel query language for semi-structured data. Journal of Digital Libraries, pp.68-88.

Deutsch A. , M.F. Fernandez M.F., Florescu D. and Levy A. (1999). A query Language for XML. WWW8/Computer Networks 31, pp.1155-1169.

Florescu D., Manolescu I. and Kossman D. (2000). Integrating Search into XML Query Processing. In Nineth International WWW Conference.

Govert N., Lalmas M. and Fuhr N. (1999). A probabilistic description-oriented approach for categorising Web documents. In Proc. Of the 9th international conference on Information and knowledge management, ACM, New York, 475-782.

Fuhr N. , and Grossjohann K. (2000). XIRQ : An extension of XQL for information retrieval. In Proc. of ACM SIGIR 2000 Workshop on XML and Information Retrieval.

Hayashi Y., Tomita J. and Kikui G.. (2000). Searching Text-rich XML Documents with Relevance Ranking. In Proc. of ACM SIGIR 2000 Workshop on XML and Information Retrieval.

Luk R., Chan A., Dillon T. and Leong H.V.. (2000). A Survey of Search Engines for XML Documents, In Proc. of ACM SIGIR 2000 Workshop on XML and Information Retrieval.

Lo M. and Hocine A. (2000). Modeling of dataweb : An approach based on the integration of semantics of data and XML. Proc. of the Fifth African Conference on the search in Computing Sciences. Antananarivo, Madagascar.

Lo M., Hocine A. and Rafinat P. (2001). A designing model of XML-Dataweb. In Proceedings of International Conference on Object Oriented Information Systems (OOIS'2001), Calgary (Canada), pp.143-153.

Nie J. (1988). An Outline of a general model for information retrieval systems. In ACM SIGIR International Conference on Research and Development in Information Retrieval, pp.495-506.

Robie J. (1999). The design of XQL, 1999. Available : http://www.texcel.no/whitepapers/xql-design.html.

Salton G., Bukley D. (1988). Term weighting approaches in automatic text retrieval. Information Processing and Mangement 24(5), pp.513-523.

Shin D. , Chang H. , and Jin H. (1998). Bus : An effective indexing and retrieval scheme in structured documents. In Proc. of Digital Libraries'98, pp.235-243.

Simonnot B. and Smail M. (1996). Modèle flexible pour la recherche interactive de documents multimedias. In Proc. Inforsid'96, Bordeaux, pp.165-178.

Smadhi S.(2001). Search and ranking of relevant information in XML documents. In Proc. IIWAS 2001, Linz, Austria, pp. 485-488.

XSLT(1999). http://www.w3.org/TR/1999/REC-xslt-19991116

Xpath(1999). http://www.w3.org/TR/1999/REC-xpath-19991116

Wang Z.W., Wong S.K. and Yao Y.Y. (1992). An analysis of vector space models based on computational geometry. In AMC SIGIR International Conference on Research and Development in Information Retrieval, Copenhagen (Danemark), pp. 152-160.

Yuwono B. and Lee D.L. (1996). WISE : A World Wide Web Resource Database System. IEEE TKDE, Vol.8, N°4, pp. 548-554.

Zhao B. Y., Joseph A.. (2000). Xset : A Lightweight XML Search Engine for Internet Applications. http://www.cs.berkeley.edu/ravenben/xset/html/xset-saint.pdf