We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

24

XML retrieval

Ben Aouicha Mohamed, Tmar Mohamed, Boughanem Mohand and Abid Mohamed
ENI Sfax
Tunisia

1. Introduction

The intensive growth of the volume of the data available on the web has generated numerous problems of access to the data/information. The tools permitting to facilitate personalized access to relevant information, and in particular research systems of information, must take the appropriate action. The problem to face is data storage i.e. index management. Many techniques have appeared in order to help store the information in a way which makes its enquiry simple and harsh. In this context, it is the volume of the information which is highlighted, without the need to evaluate the traditional research modals of data.

Nevertheless, the appearance of new documents' formats requires the adaptation of these modals to be able to find the relevant information. In fact, content alone is not enough to decide about the relevance of the document if it is structured. The XML document is what matters in this problem. The notion of the structure has been made to complete the content of the document and must be dealt with within a research method of information, known as research of structured information/data. The traditional research modals of data processed the structure of documents as relevant only in a marginal way. With the new requirements set by research of structured information/data, they can not separate between storage techniques and structured data query which makes the reputation of database management systems. On the other hand, database management systems process only the structure: All the data is supposed to be atomic and consequently no content process is realised. The community of database and information research has put a long time dealing with this topical problematic. We find approaches based on the content (document-centric). They mainly process content taking into consideration structure as an additional dimension. We also find approaches based on structure (data-centric).They prioritize data structure and process the content by generating similar queries of SQL.

According to this technique, there is much enthusiasm about the content or the structure. The crossing to the scale contrary to any expectation shows that structure damages the quality of response. The current data research techniques give much importance to structure, but always in a marginal way.

An XML document is assimilated to a tree where the nodes represent XML embedded visible commands and can contain content elements. Contrary to the classic data research, a query may be formulated in XML. The access to XML documents consists in finding the

fragments of a tree relevant to the query and having an illustration within the tree relative to the document in terms of content and structure: A good research system of structured data must take into consideration these two aspects in a flexible way i.e. the similarity in terms of content and / or structure must be a graded measure.

The classic XML query languages are based on the exact comparison. They do not allow providing binary results, but in data researching (classic or structured), we try to organize the documentary granules according to their potential relevance.

The appearing document/query must be realised in the same way as the documentary granules, whom structure shows slight differences with the query structure, get a score. They can equally be considered as the reverse of the strain necessary to the incremental construction of a tree out of another.

The comparison of trees was the concern of many initiatives. They were all based on the most basic updating methods which each is provided according to a certain cost. The construction cost is the cumulative cost of all the necessary operations for the construction. We start from the representation of a tree and then we transform it into another with the minimum of possible strain. While in an experimental situation, algorithms used to compare literature trees have shown not to be adapted to data structured research.

Less complicated than the algorithms of comparing trees, the Levenstein distance, initially suggested to measure the distance between two character chains rather than trees, is the alternative solution, but it remains costly since it is based on the dynamic programming which is costly itself.

The processing of content is equally problematic. In fact, the textual/ word content does not always appear in node leaves. However, an internal node can be relevant even if it does not contain any indexing terms because relevance does not arise only from structure. In order to restore order within the nodes so that the leaves are not privileged compared to internal nodes, the latter must have a score relevant to the content. In literature, we distinguish two common principles. The majority of RI models use score propagation: We propagate and spread the score of a node relevant to a query (in terms of content) to its ancestors. The other models are based on the spreading / propagation of content: Instead of spreading the score, we spread the content of a node to another via descendents 'links and we calculate its score independently.

Only the distance between XML documents is taken into consideration during the propagation (depth), however, there exist other criterians (related to the width of a tree or a fragment of a tree) that may have an important impact on the quality of the system response.

The processing of both content and structure are the major problems of data structured research. It is essential to obtain relevant scores at a scale highly graduated. It is at this level that occurs a problem: how to combine both scores to be able to supply a sorted list of XML elements. Otherwise, every XML element must get a unique score obviously depending on its relevance according to the structure and content, but according to what and how ? A linear combination seems to be a solution, whereas facing the experimental evaluation, other more complicated combinations must be experienced, and many other parameters must be taken into consideration.

The traditional problematic related to the evaluation of data relevance compared with a query is still of topical importance. However, it is getting complicated and it implies other issues within the framework of XML documents, notably concerning structure.

Queries orientated to content, which are far from being easy for the user, dictate for data research systems to decide about the appropriate granularity of the data to be sent. Data units must be the most possibly exhaustive and specified compared to the query. Unlike traditional data research, relevance within the framework of structured data research is in fact expressed according to two dimensions: exhaustivity and specificity. Exhaustivity allows to measure to which point data units respond to the users 'needs. Specificity, on the other hand, allows to measure to which point data unit content focuses on the users' requirements. Research and sorting models of data units must therefore take into consideration those two dimensions explicitly, which is not of course the case of the suggested approaches in literature and notably database orientated approaches. Within the framework of query, two cases are possible. In the first case, the user can express conditions regarding the structure of the document, but can not precise the type of data units he wishes to resent using the system. This problematic, in which structured data can be used only as an indication to help find the relevant information and not as an indication of what the user wants, was not dealt with in literature. The second case is concerned about queries for which the type of the element to be resent is specified by the user. Other relevance notions come therefore into play. The size of specificity has no longer a meaning since the user precise the data granularity he needs. Meanwhile, the content of structure elements as well as path commands present within the query should be able to be processed in an indefinite way. Otherwise, a degree of relevance must be allocated to the elements.

2. Issues and problems of structured information

XML documents have updated the problem of information retrieval. In traditional IR or unstructured, the documentary unit returned to the user is the entire document. The first challenge of the query in XML documents deals essentially with the concept of documentary unit; in fact, in the case of XML documents, any item can be returned as a response to the request. Therefore, there is no standard unit (whole document). If our request for document shown in Figure 2.1 is photography, we can return the title element, the element section or the article: the result will be a granule document (a portion of the document).

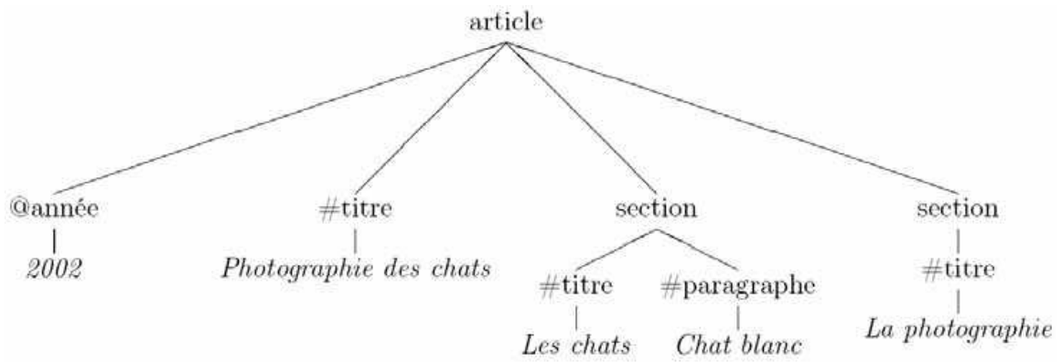

Fig 2.1 Example of XML document

2.1 Queried Information

A structured (SIRS) is expected to set a criterion to select the most appropriate of an XML document [1]: such a system should always seek the most specific of a document responding to a query. This motivates a strategy for recovery that returns the smallest unit containing the information sought, but not below this level. However, it can be difficult to implement this principle algorithmically. Consider a query seeking the information unit containing the word photography, in the example in Figure 2.1 we have two title elements containing that word. But in this case, the title which contains only the photography is very specific [2][7]. However, items returned must not lack exhaustivity: if multiple items are very specific but comprehensive low, they can be grouped to form a less specific but very comprehensive. Deciding what level of the tree potentially responds to a request is difficult. We must consolidate knots, but the problem of this is that parts of the document may not make sense for the user because they are not logical.

Because of the redundancy caused by the small size ,it is common to restrict the set of elements that are eligible to be returned. So the restriction policies must :

- Remove all small items,
- Avoid all kinds of things which users do not look at,
- Avoid all kinds of things that users do not generally consider appropriate (if relevance assessments are available)
- Keep only the types of items that a designer or a library system considered as useful.

In most of these approaches, result sets contain small parts. Thus, we can remove a few items in a stage of post-treatment to reduce redundancy. Alternatively, we can group several smaller items together. If conditions are accentuated in the need for information, average scanning of the elements takes longer time than the sweeping of the smaller.

Thus, if the section and paragraph occur in the list of results, they would be sufficient to show the section. The advantage of this approach is that the paragraph is presented together with its context (i.e. section). This context may be useful in the interpretation, subsection (for example, the reported source of information) is preserved if the paragraph alone satisfies the query. If the user knows the schema of the collection and may indicate the type of item desired, then the problem of redundancy is tolerated as long as some elements have the same nested type.

2.2 Expression of information needs

In structured IR (SRI) users must indicate the items they interview. Therefore the interface formulation is more complex than a search box of information needs consisting of simple lists of key words as in classical IR. We can also assist the user by providing suggestions based on the structures most frequently in documents of the collection. The complaints contain an information content (as in traditional IR) but may also contain structural information used in the RIS. Query languages [29][14] used for structured queries are typically based on XPath. However, users prefer relaxed structural constraints, to better target the units returned.

3. Approaches to IR in XML documents

Access to XML documents has been apprehended from two important angles [3] the data-driven approach that uses techniques from the database (data-centric), and approach-oriented(document-centric) adapting the techniques developed by the RI. table 3.1 illustrates the principles of each community for the treatment of structured documents.

	DBMS	IRS
Be care in information	Precis	Blur
Result	Exact	Inaccurate
Request	SQL	Key words
Model	Set theory	IR Models

Table 3.1 Title of table, left justified

3.1 The data-driven approaches

Data-driven approaches overlay realized the integration of XML in a relation object. The overly allows you to transform XML documents into tables and vice versa. The aim of these approaches is to use the wealth of IR textual keywords in a query of the database. At this stage, many query languages have been developed to query XML documents like XPath and XQuery. All these languages can integrate predicates in terms of content and structural information of XML documents in an efficient way, but they are limited because they treat the text in a binary(present\absent),or it has been shown in IR text that the weighting keywords is required for the graduation of the document relevance and thus the return of an ordered list of results.

3.2 Document–oriented approaches

RI-driven approaches consider XML documents as collection of text documents with tags and relations between these tags. The main purpose of this community focuses on the use of the information carried by the document structure to improve search results and changes the granularity of the latter (structural elements instead of entire document).

The classic IR models (Boolean model, vector and probabilistic) have been extended to take into account the coexistence of the content and structure. They are concerned with effective treatment of textual content carried by the structural elements in XML documents. The relevance of a document against a query is an aggregation of all relevance values assigned to elements relevant to a query.

The structural dimension is taken into account together with the text size. In several approaches, the textual content, whether weighted or not, is undergoing a spread of the leaves to the ancestors. This technique is highlighted with a view to appear in each element of the text content according to its descendants, and thereafter to evaluate the score of each depending on the content. There is a similar approach which is to inversely calculate a score in the leaves and spread to the ancestors [19].

4. Techniques for indexing structured documents

Indexing structured documents appears more complex compared to flat documents. In fact, the indexing of documents is just flat out words representative of each document.
The document indexing service structure however is more complex with reference to the co-existence of textual information and the structural information, so a structured document indexing is to find a way representing these two types of information. One of the main challenges in the RIS is to find a way representing these two types of information. One of the main challenges in the RIS is to find a way to represent structural information in an XML document in order to use this index in the calculation of relevance between a document and an XML query. Subsequently, textual information should be represented according to the structural information. In this section we present the different approaches proposed in the literature to address the problem of indexing structured documents.

4.1 Indexing of textual information

The textual information in XML documents is localized in the leaf nodes (of types # PCDATA) approaches for basic data they consider that these nodes have the text, unlike the document-oriented approaches, where the term is weighted to reflect its importance.
The problem of indexation of the text unit is to find this information with that of the structural information to identify the element relevant to answering the query. There are two visions for indexing textual information in documents according to XML IR approaches textual information is processed simultaneously or independently of the structural information. The spread of the terms of the current approaches embody this relationship by spreading leaf nodes to their ancestor [8]. In general, we calculate a weight for each term in the node that contains it, then that word is spread to ancestor nodes by reducing its weight by the distance between the node that contains it and to which the word is spreading indexing of textual information [2]. Separate units: approaches ignore the current stage of indexing the structural relationship between different nodes in the XML document. They consider that nodes are disjointed units [9][11][13] [21] [26].

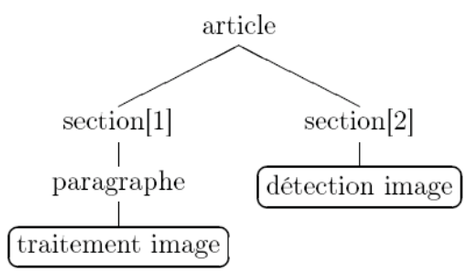

Fig. 4.1 Example of an XML document section

4.2 Indexing of the structural information

We can distinguish between the modes of representation of the documents [25] the following methods :
- Indexing fields: For each term of the document, we indicate the nodes that contain that contain [17]. So for each filed or tag located on the information text. With this method, we

filter during the search, the tags containing the text in question. Table 4.1 illustrates the result of indexing the document shown in table 4.1.

Word	**Field**
Traitement	Paragraphe
Image	Paragraphe
Detection	section2
Image	section2

Table 4.1 Indexing based on fields

Indexing paths instead of the name tag for locating textual information, indexing paths replace the name of the tag for locating textual information element based on XPath [17] as shown in table 4.2.

Word	**Field**
traitement	/article/section1[1]/paragraphe[1]
Image	/article/section1[1]/paragraphe[1]
Detection	/article/section2[1]
Image	/article/section2[1]

Table 4.2 Indexing based on the ways

This accelerates the query process for locating information in multiple tags with the same name. The major drawback of this method is that it does not describe the relationship of the offsprings of different elements of the XML document. Indexing trees: this indexing technique is similar to indexing paths [22], but unlike the latter it can assign to each node the values of pre-order and post order to distinguish the relationship ancestor descendant (hierarchical) [21]. Table 4.3 illustrates an example of indexing based on trees. The index structure of XPath Accelerator [20] allows representing the XML document tree by assigning to each node of increasing values pre-order or post-order as shown in table 4.3.

Word	**Path**
traitement	/article/section1[1]/paragraphe[1] **(3)**
Image	/article/section1[1]/paragraphe[1] **(3)**
Detection	/article/section2[1] **(4)**
Image	/article/section2[1] **(4)**

Table 4.3 Example of indexing based on the trees

5. Query models of XML documents

The classical models of IR have been adapted to the IRS taking into account the structural dimension. Regardless to research models, the matching is preformed using two different approaches

- Either at the level of the elements returned by a spread of terms that are weighted or not.

- Or at the level of the smallest unit of indexing. In this case the elements are returned with a spread of relevance.

5.1 The extended vector model

The extended vector model [10] is an extension of the model vector that separates the information provided by the structure of information content. Taking into consideration the structural information is highlighted by identification of a vector space in which documents (or elements of each document) and queries can be represented by vectors. This identifies a basis in which each vector element or XML query receives coordinates scalars. The basis of the extended vector model is based on the representation of each dimension by a sub-lexical tree. A sub- lexical tree is a path where the XML is a term sheet index, and internal nodes arc the names of XML tags. If we create a separate dimension for each sub-lexical tree that appears in the collection, the dimension of space becomes very large dimensions but many have little use because they usually appear in a single document. We can deal with the structural sub-lexical remaining trees the same as indexing terms in the classical vector model.

This means that we can use the formalism of the vector space for XML query. The difference is that the dimensions of the vector space in the unstructured query are the terms while they are also sub-trees in the structured search. If we want to restrict the structural dimension, we are left with the traditional vector model. We can now represent the XML paths, whether derived from applications or documents, as vectors in this vector space and calculate the similarities between them. A simple measure of the similarity of a path in a c_d document and a path in a query c_q is calculated by the following function c_r:

$$Cr\begin{cases}\dfrac{1+c_q}{1+c_d} & \text{if } c_d \text{ can be produced from } c_q \text{ by adding some element} \\ 0 & \text{so not}\end{cases} \quad (1)$$

The final score of the document by using the cosine measure is given by:

$$sim\_c_r(c_q,c_d)=\sum_{c_k\in C}\sum_{c_l\in C}c_r(c_k,c_l)\sum_{t\in V}wieght(q,t,c_k)\frac{weight(d,t,c_l)}{\sqrt{\sum_{c\in C,t\in V}wieght^2(d,t,c)}} \quad (2)$$

where V is the vocabulary of non-structural dimensions, C is the set of all paths of the XML tree and $weight(q, t, c)$ and $weight(d, t, c)$ are respectively the weight of a term t in a component c in the XML query q and document d. You can use the measures $tf\ x\ idf$ to calculate the weight of a term t. Because of the structure of documents, different types of models have been extended in various ways. Thus we must take into consideration the additional parameters that have arisen with the IRS. To do this, we must then adjust the formulas and inject structure parameters such as the depth of the document, the number of children, etc.

5.2 The extended Boolean model

To allow the expression of more powerful queries to specify the relationship between the terms of the index, the Boolean model has been extended with a new non-commutative binary operator contains. The first operand is of type XPath and the second is a Boolean expression. This model allows applications to be completely specified in terms of content and structural information based on the query language XPath. The research is to extract the title and convert it to boolean query, the elements considered relevant are then ranked by the sum OkapiBM25 [18]. Thief et al. [19] use a combination of methods using a probabilistic regression logistics with an approach based on the Boolean model to assess the relevance of the documents and items. The value of probability of relevance *R* of a component C (component) is calculated as the product of probabilities of relevance of *C* opposite the application *Qbool* by a Boolean model and relevance opposite the application *Qprob* by a probabilistic model:

$$P(R \mid Qbool, C) \times P(R \mid Qprob, C) \tag{3}$$

This combination allows you to restrict ail documents relevant to documents with a Boolean value equal to *1* while allocating a row.

5.3 Probabilistic Model

In the probabilistic model [27], classification of documents is based on the probability that the retrieved document d implies the query q. To extend the probabilistic model for XML documents, the odds must take into consideration the structural information. Two approaches have been developed.

The first approach allows the use of conditional probabilities of joint, for example *P(d | t)* becomes *P (d | p contains t)* where d represents a document or part of the document, *t* is a term and *p* is a path in the tree XML, the second allows to extend the logic to pro and takes into consideration issues related to the structure: this approach is based on the definition of relations between record of tables in a database, and modeling predicates with logical formulas. As for the models together, this formalization does not order a list of documents and makes no room for vagueness of the wording of the request. This problem is solved by Fuhr [8] who focused on the IR in databases. He proposed to combine the approaches of IR and databases. It proposes a probabilistic relational algebra that is a generalization of relational algebra. This algebra is to assign probability weights to record of a relationship. These weights give the probability that a record belongs to a relationship. This approach has two advantages: it allows representing data values with vagueness and classifying documents according to their probability weight.

This method is based on the query language XIRQL [9], and was implemented within the search engine HyRex. The terms are propagated to the nearest indexed node. The weight of relevance of the nodes is calculated through the propagation of the weights of terms in the document tree. The weight of each term decreases by multiplying each factor called growth factor. Considering the structure of the document shown in Figure 5.1, we assign to each term a weight according to its probability of occurrence in a node. We want to calculate the weight of the term model in the root element of the document presented in Figure 5.1 (Article):

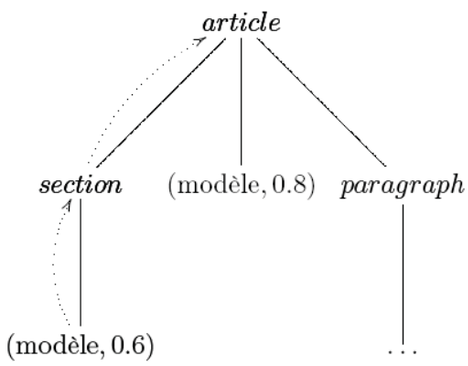

Fig 5.1 Model of increase

Thereafter while introducing the factor of increase *(0.7)* we will have for queries that contain structural conditions, weighted sums of these probabilities are aggregated.

5.4 The inferential model

In information retrieval in XML documents, diagrams inference has been adapted to express the causal relationship between the content and structure. Piworwarski et al. [24] have proposed a probabilistic model based on Bayesian networks where the dependency hierarchy is expressed by conditional probabilities. The probability of relevance of a given the parent p to query q is:

$$P(e = a \mid p = b, q) \frac{1}{1 + e^{F}_{\varepsilon, a, b_q}} \tag{4}$$

Where F is the relevance of the element e as the Okapi model. Query q Structure is decomposed into a set of n elementary subqueries qi . Each of these sub queries reflects an entity structure and a need for information. The final score is given by:

$$rsv(e_i, q) = rsv_{q_i}(e_i, q) \times rsv_{q_n}(e_i, q) \tag{5}$$

De compos, Fernandez and Huete [6] have also proposed a research model based on Bayesian networks where the diagram is inference based on conditional probability. Two types of diagrams are: SID(simple Inference Diagram) and CID (context based Inference Diagram). A Diagram consists of a qualitative component (representing the variables and inferences) and a quantitative component (probabilities of nodes).

5.5 Language Models

Sigurbjornsson et al. [28] have proposed a model combining language models of the element, the document and the collection. To estimate the model language, the authors used two types of index: an index to the elements of the XML document that provides the same

function as a file in reverse RI classic and another (index article) for any document used for statistical calculations.
For each element *e*, we estimate the model language (score) for a query *q* by:

$$P(e \mid q) \alpha P(e) \times P(q, e) \tag{6}$$

5.6 The XIVIR model

XIVIR is an XML information retrieval based on tree matching [3][4][5] . The approach consists of comparing document and query representations, computing a structure and a content score to each document node and then combines them into a final score.

5.6.1 Structure retrieval

When querying an XML corpus, best structure matches should privilege document parts that fulfill as more as possible the query structure. Slight structure differences should be tolerated in order to provide a ranked list of document parts.
Formally, an XML tree is a set of node paths $A \to B$ where node *A* is the parent of node *B*.

To each node, a set of weighted indexing terms is associated. The term weighting formula used will be presented in section 4 and has no effect on structure-based retrieval. The XML tree root is the only one that has no parent, so an XML tree *T* should have the following property:

$$\{N, \forall N', N' \to N \notin T\} = \{root\}$$

The structural retrieval process should look for the deepest and largest sub-tree shared by both representations. To do so, we add to each path $A \to B$ a weight reflecting the importance of the relation between nodes *A* and *B*. According to the parent-child relation, this weight is equal to *1*. The more *A* is distant from *B* in the original path, the less its weight is. We use the weighting function *f* defined by f($A \to B$) = *exp(1 − d(A,B))* where *d(A,B)* is the distance that separates node *A* from node *B*. We denote this path by $A \overset{w}{\to} B$ where
w = *exp(1 − d(A,B))* is the weight of the path $A \to B$.
Figure 5.2 shows how a path $N1 \to N2 \to N3 \to N4$ is extended to a set of weighted paths.
To support flexible structure matching, we start by extending the query and the document trees paths, and then we look for the deepest and widest sub-trees shared by both sets of weighted paths.

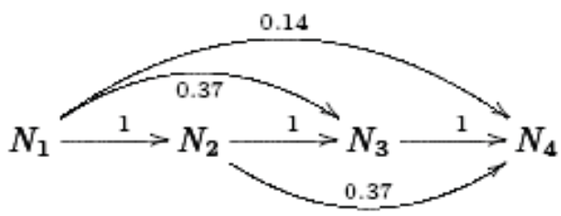

Fig 5.2 Original and additional paths, the original paths are weighted by 1.

We assume that this set of paths is potentially a relevant structure sub-tree, its score depends on the weight subscribed on each path according to the query and its corresponding according to the document. We use the cumulative product of each path weight according to the query by its corresponding according to the document:

$$rsv_s = \sum_{A_q \xrightarrow{w_q} B_q \in E_q \equiv A_d \xrightarrow{w_d} B_d \in E_d} w_q \times w_d \tag{7}$$

where rsv_s denotes the structure-based score of a retrieved sub-tree following the query structural conditions. *Eq (rep. Ed)* is a set of weighted paths generated from the query (resp. document) tree and $T_q \equiv T_d$, shows that T_q is the homologous of T_d.

5.6.2 Text propagation and content-based retrieval

The content-based score is computed for each document node according to a given query. This score is computed independently from the structure-based retrieval as follows:

$$rsv_c(n) = \sum_{t \in n \cap q} w(p,n) \tag{8}$$

where $w(p, n)$ is the weight of term t in node n and $n \cap q$ is the set of terms appearing in both the query and the element node. The term weight in a document node is computed as follows:

$$w(t,n) = ief_p \times \sum_{n \to c_1 \to c_2 \ldots c_k} \frac{tf(t,c_k)}{d(n,c_{k+1})} \tag{9}$$

where $tf(t,c_k)$ is the frequency of term t in node ck and idf_p is the inverse of element frequency of term t and is done by the following:

$$ief_t = \frac{N}{n_t} \tag{10}$$

where N is the total number of elements and n_t is the number of elements containing term t. Equation 9 shows that each node content is propagated to its ancestor nodes. For example, if node A is the parent of node B which contains term t, we assume that node A contains term t and we down weight it in node A by dividing it by $2 = d(A,B)+1$. If node C is the parent of node A, term t is propagated from node B to node C and we divide its weight by $3 = d(C,B) + 1 \ldots$

The approach consists of comparing document and query representations, computing a structure and a content score to each document node and then combine them into a final score.
The evaluation of the performance of a IRS, is generally based on measures of recall and precision. For the assessment of IR in XML documents, there is at present only one companion evaluation: INEX (Initiative for the Evaluation of XML retrieval).

6. INEX evaluation Company

INEX is a program that produces collections, sets of queries [2][11][23], and judgments of relevance. The INEX annual symposium is held to present and discuss research results. This company began in 2002; Table 2.5 illustrates the evolution of data in the companion INEX evaluation. It should be noted that the task CO (Content Only) is always present in the INEX.

6.1. Test Collection

The companion evaluation INEX provides a collection of documents prepared in XML format. The collection contains 12,107 items, about 500 MB, published from 1995 to 2002, this collection was used during the evaluation of the years 2002, 2003 and 2004. In 2005, the collection has been enriched, it contains 16,819 articles covering the period from 1995 to 2004 and totalling approximately 750 MB. The DTD includes 192 different tags and an article is an average of approximately 1500 knots, and is of average depth of 6.9.

6.2. Queries (Topics)

In 2003, to express the information needs, XPath is the formalism that was used. This formalism was complex, the rate of errors in applications reached 63%. In language 20,041th NEXI (narrowed Extended XPath I) was proposed as a formalism for the expression of need, the error rate decreased to 12%. In 2005, the language NEXI was again adopted for the expression of information need. Content only (CO-type queries, Content Only), queries with information about the content and structure (type CAS, Content and Structure)

6.3. The relevance judgments

Assessing the relevance of SRIS goes through a phase of validation of documents returned by the IRS. Each element of the document or the document is considered in full by hand (by participants) for each request. This phase allows to obtain judgments of relevance. A two-dimensional scale was proposed: the first dimension measures the comprehensives and the second dimension measures the specificity of an element for a given query. These two dimensions are multivalued, which allows for elegance in the judgments.
Completeness reflects only the presence or absence of the information sought in an item, even if this information appears in only a small part of the element. For example, the element representing the entire document will be considered as highly comprehensive even if only one paragraph in the whole document is very relevant to the application and that the rest of the document is not. One can distinguish for this endpoint four levels of completeness

- Not exhaustive(0): the element does not address the subject of the complaint;
- Slightly comprehensive(1): the marginal addresses the topic of the request;
- Fairly comprehensive (2): the subject of the complaint is largely addressed in the element;
- Highly exhaustive (3): the subject of the complaint is dealt with comprehensively in the item.

Specificity is totally related to the evolution of structured documents. This measure examines the degree to which the element returned by the system processes all the information sought. One can distinguish for this endpoint four levels of specificity

- Not specific (0): the element returned contains no relevant;
- Low specific (1): only a small part of the information contained element is in the relevant information;
- Fairly specific (2): the largest part in the element is of the relevant information ;
- Very specific (3): the element returned contains only relevant information.

7. References

[1] Abolhassani, M. & Fhur. N. (2004). *ECIR*, Applying the divergence from randomness ISBN, approach for content-only search in xml documents, pp 409–419

[2] Anh, V. N. & Moffat. A. (2002). Compression and an ir approach to xml retrieval, *Proceedings of INEX 2002 Workshop*, Dungstuhl, Germany

[3] Ben Aouicha, M.; Tmar, M.; Boughanem, M. & Abid, M. (2009), Experiments on Element and Document Statics for XML Retrieval based on tree matching, *International Journal of Computer and Information Science and Engineering, IJCISE*, Vol 3 n°1

[4] Ben Aouicha, M.; Tmar, M.; Boughanem, M. & Abid, M. (2008). XML information retrieval based on tree matching, *IEEE International Conference on Engeneering of Computer-Based Systems, ECBS*, 31 Mars-4 Avril, 2008, belfast, Ireland

[5] Ben Aouicha, M. Modèle de recherche d'information strcuturée basé sur la relaxation de requêtes, *Congrès INFormatique des ORganisations et des Systèmes d'Information et de Décision, INFORSID*, 27-30 Mai 2008, Fantainebleau, France

[6] Campos, L. M.; Fernández-Luna, J. M. & Huete, J. F. (2008). Collaborative recommendations using bayesian networks and linguistic modeling. *ICAISC*, pp 1185–1197

[7] Fernàndez, M. F.; Jim, J.; Morton, K.; Onose, N. & Siméon, J. (2007). Highly distributed xquery with dxq. *SIGMOD*: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp 1159–1161

[8] Fuhr, N. & Rölleke, N. (1997). A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1), pp 32–66

[9] Fuhr, N. & Großjohann, K. (2001). Xirql: A query language for information retrieval in xml documents, *SIGIR*, pp 172–180

[10] Fox, E. A. (1985). Composite documents extended retrieval. *In Proceedings of ACM SIGIR*, pp 45–53, Montréal

[11] Govert, N.; Abolhassani, M.; Fuhr, N. & Grossjohann, K. (2002). Content-oriented xml retrieval with hyrex. *Proceedings of INEX 2002 Workshop*, Dungstuhl, pp 26–32, Germany

[12] Grust, T. (2002). Accelerating xpath location steps. *In Proceedings of the ACM SIGMOD International Conference on MAnagement of Data*, Madison, Wiscon, USA

[13] Gutierrez, A.; Motz, R. & Viera, D. (2000). Building databases with information extracted from web documents. *SCCC '00: Proceedings of the XX International Conference of the Chilean Computer Science Society*

[14] Huck, G.; Macherius, I. & P. Fankhauser. (2000). Pdom : Lightweigt persistency support for the document object model. *In Succeeding with object Databases*, John Wiley

[15] Jones, S.; Hancock-Beaulieu, M.; Robertson, S.; Walker, S. & Gatford, M. (1994). Okapi attrec 3. *In Proceedings of the 3rd Text REtrieval Conference*

[16] Kamps, J.; Rijke, M. & Sigurbjörnsson, B. (2004). Length normalization in XML retrieval. *Proc. SIGIR*, pp 80–87,

[17] Kazai, G.; Lalmas, M. & Roelleke, T. (2002). Focused document retrieval, *International Symposium on string processing and information retrieval*, Lisbon, Portugal

[18] Luka, R. W.; Leong, H.; Diloon, T. S.; SHAN, A. T.; Croft, W. B. & Allan, J. (2002). A survey in indexing and searching xml documents, *JASIST*, 53(6): pp 415–437

[19] Larson, R. R. (2002). Cheshire ii at inex: Using a hybrid logistic regression and Boolean model for xml retrieval, *INEX Workshop*, pp 18–25

[20] Lalmas, M. & Piwowarski, B. (2005). Inex 2005 relevance assessment guide. https://inex.is.informatik.uniduisburg.de/2005/pdf/relevance Assessment2005

[21] Piwowarski, B. & Gallinari, P. (2005). A bayesian framework for xml information retrieval: Searching and learning with the inex collection. *Inf. Retr.*, 8(4): pp 655–681

[22] Ogilvie, P. & Callan, J. (2003). Using language models for flat text queries in xml Retrieval, *Proceedings of INEX 2003 Workshop*, pp 12–18, Dungstuhl, Germany

[23] Rölleke, T.; Lalmas, M.; Kazai, G.; Ruthven, I. & S. (2002). Quicker. The accessibility dimension for structured document retrieval, *ECIR*, pp 284–302

[24] Rijisbergen, C. J. V. (1979). Information Retrieval. *Dep of computer Science*, University of Glasgow

[25] Sigurbjörnsson, B.; Kamps, J., & Rijke, M. (2003). *Proceedings of INEX 2003 Workshop*. An element-based approach to XML retrieval., Dagsthul, Germany

[26] Sauvagnat, K. & Boughanem, M. (2004). Le langage de requête xfirm pour la recherche d'information dans des documents XML: de la recherche par simples mots-clés à l'utilisation de la structure des documents. *Congrès Informatique des Organisations et Systèmes d'Information et de Décision, INFORSID*, pp 107–124

[27] Sauvagnat, K.; Boughanem, M. & Chrisment, C. (2006). Answering content and structure-based queries on XML documents using relevance propagation. *Information Systems*, 31(7): pp 621–635

[28] Sauvagnat, K. & Boughanem, M. (2004). The impact of leaf nodes relevance values evaluation in a propagation method for xml retrieval, *SIGIR*, pp 19–22

[29] Shin, D.; Jang, H. & Jin, H. (1998). Bus : an effective indexing and retrieval scheme in structured documents. *In Proceedings of the ACM International Conference on Digital libraries*, pp 235–243, Pittsburgh, Pennsylvania, USA

IntechOpen

Recent Advances in Technologies

Edited by Maurizio A Strangio

ISBN 978-953-307-017-9

Hard cover, 636 pages

Publisher InTech

Published online 01, November, 2009

Published in print edition November, 2009

The techniques of computer modelling and simulation are increasingly important in many fields of science since they allow quantitative examination and evaluation of the most complex hypothesis. Furthermore, by taking advantage of the enormous amount of computational resources available on modern computers scientists are able to suggest scenarios and results that are more significant than ever. This book brings together recent work describing novel and advanced modelling and analysis techniques applied to many different research areas.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ben Aouicha Mohamed, Tmar Mohamed, Boughanem Mohand and Abid Mohamed (2009). XML Retrieval, Recent Advances in Technologies, Maurizio A Strangio (Ed.), ISBN: 978-953-307-017-9, InTech, Available from: http://www.intechopen.com/books/recent-advances-in-technologies/xml-retrieval

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen