ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

Information Retrieval Systems in XML Based Database – A review

Preeti Pandey[1], L.S.Maurya[2]

Research Scholar, IT Department, SRMSCET, Bareilly, India [1]

Associate Professor, IT Department, SRMSCET, Bareilly, India [2]

ABSTRACT: **XML - the eXtensible Markup Language has emerged as a new standard for data representation and exchange over the Internet. It will become a universal format for data exchange on the Web and that in the near future we will find vast amounts of documents in XML format on the Web. As a result, it has become crucial to sort large collections of XML documents and retrieve relevant information from the collection efficiently and effectively. In our paper we are focusing on reliability of a software component. XML as a data interchange format helps in communicating across heterogeneous platform and hence establish a reliable communication with easy to maintain individual software components. We have a corpus of XML documents. The stop words are removed to remove the unimportant terms. The next step is stemming, after that parsing of the documents is done to generate the structural terms and thus structural terms are stored. Then searching of index structure is done to retrieve the documents based on some ranking criteria, in response to user query.**

Keywords: **Stopword, stemming , ranking, indexing**

I. INTRODUCTION

Information Retrieval (IR) is the study of methods for capturing, representing, storing, organizing, and retrieving unstructured or loosely structured information[2]. This process involves several stages starting with representing the data and ending with returning results to the user. Intermediate steps include search and match operations, ranking mechanisms, and filtering processes. In an IR environment, a successful search approach is one that is able to provide the most relevant results to the user in a conceivable amount of time.

XML - the eXtensible Markup Language defines a set of rules for encoding documents in a format that is both human-readable and machine readable As a result, it has become crucial to sort large collections of XML documents and retrieve relevant information from the collection efficiently and effectively.

A distributed system consists of multiple autonomous computers that communicate through a computer network. In distributed computing, each processor has its own private memory, information is exchanged by passing messages between the processors.

So, EAI (Enterprise Application Integration) is used to integrate different application (distributed), by using RMI, CORBA and DCOM using RPC. But the drawback associated with this was not protocol independent and maintenance cost was much because of extra overhead.

Further enhancement came in form of ESB (Enterprise Service Bus) only provides Services and events it becomes totally general and enables application well beyond the scope of mere integration. EAI & ESB together ensure distributed communication with guaranteed message delivery by routing and transport mechanism.

Further development of XML and web services helped to ensure interoperability of different technologies and integration, but the problem associated with this was tightly coupled in nature, the component once bonded could not be used separately with different components. To overcome this problem we are planning to use XML based model for reliable communication so that we could improve maintenance overhead.

In our paper, we describe information retrieval models in next section. In section III, we explain about information retrieval system, what model or structure we are generating for finding relevant information from the XML Based Database. We described two algorithms for stemming – Lovins and Porter algorithm[1]. We are using here Porters algorithm as it is simple and a new version of Lovins algorithm. Porters develop his website for easy implementation of stemming process. Results and Discussion is discussed in section IV. In section V, Conclusion includes the analysis of relational databases and IR models. We interpret some of the work of Information Retrieval system for future in section VI. Lastly we mentioned our references from where we got ideas[7].

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 9, November 2012

II. INFORMATION RETRIEVAL MODELS

An IR model specifies the details of the document representation, the query representation, and the retrieval functionality. A web IR model involves the representation of the ranking functionality as well. The fundamental IR models can be classified into boolean, vector, and probabilistic model.

A. Boolean Model:

In the boolean model, a document is associated with a set of keywords. Queries are also expressions of keywords separated by AND, OR, or NOT/BUT. The retrieval function in this model treats a document as either relevant or irrelevant. Due to the fact that the web is rather redundant, the possibility of returning millions of web results having the same rank makes it difficult to assign display priorities to those documents when this model is implemented.

B. Vector Space Model:

This is the most recognized IR model. A degree of similarity between the query vector and all documents represented by the vector space is measured for ranking purposes. XML retrieval requires taking into consideration the structural context of terms. We make use of the vector space model to represent this structural context as shown in fig.1. XML retrieval requires taking into consideration the structural context of terms.

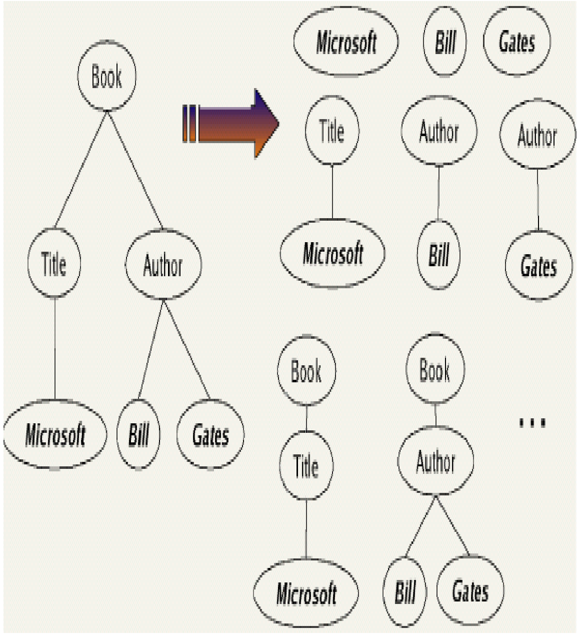

Fig.1 A mapping of an XML document (left) to a set of structural terms (right)

C. Probabilistic Model:

A probabilistic IR model is based on the assumption that there is a set of documents that represents the model answer to the user query[3]. In this model, a preliminary set of documents is selected and monitored by the user. This is done by using interactive interfaces with immediate user feedback on the way to achieving the model answer set. In other words, approaches based on this model rely heavily on feedback from their users during query time.

The probabilistic retrieval model is based on the Probability Ranking Principle, which states that an information retrieval system is supposed to rank the documents based on their probability of relevance to the query, given all the evidence available [Belkin and Croft 1992]..

III. RELATEDWORK

XML information retrieval is an approach for providing more focused information than traditionally offered by search engines. As earlier relational based database are better for handling large volumes of data within a system. They have mature management systems which efficiently and reliably maintain large quantities of structured data. There is no equivalent XML management system, and using XML documents directly for storing and maintaining large volumes of data can prove both inefficient and unreliable.

XML documents contain both the data and the informative relationship structuring of that data in a way that both machines and people can read. An XML document can be electronically transmitted from one party to another and all the information is carried with it, so they are self describing. Since XML does a very good job of delivering self-describing data feeds it has become a key standard in Service Oriented Architectures (SOA) and Web Services

We divide our work into five steps :
A. Stopword Removal
B. Stemming
C. Indexing
D. Ranking
E. Query Evaluation

A. Stopword Removal:

Stop words can be viewed as a type of signal noise which interrupts the ability to quickly ascertain the relevance of search results or the meaning and importance of words in a document. By filtering out such words, the message becomes clearer or more useful. Filtering of stop words to reduce index size or to assist users in providing search queries that will provide better results. All stop words, for example, common words, such as *a* and *the*, are removed from multiple word queries to increase search performance.

B. Stemming:

Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form — generally a written word form[1].

A stemmer for English, for example, should identify the string "cats" (and possibly "catlike", "catty" etc.) as based on the root "cat", and "stemmer", "stemming", "stemmed" as based on "stem". A stemming algorithm reduces the words "fishing", "fished", "fish", and "fisher" to the root word, "fish".

The first ever published stemmer was written by Julie Beth Lovins in 1968. This paper was remarkable for its early date and had great influence on later work in this area. A later stemmer was written by Martin Porter and was published in the July 1980 issue of the journal Program. This stemmer was very widely used and became the de-facto standard algorithm used for English stemming.

There are two widely used stemming algorithms: Lovins algorithm and Porter's algorithm.

Lovins algorithm specifies 260 suffix patterns and uses an iterative heuristic approach. The design of the algorithm was much influenced by the technical vocabulary with which Lovins found her working (subject term keywords attached to documents in the materials science and engineering field). The subject term list may also have been slightly limiting in that certain common endings are not represented (ements and ents for example, corresponding to the singular forms ement and ent), and also in that the algorithm's treatment of short words, or words with short stems, can be rather destructive. The Lovins algorithm is noticeably bigger than the Porter algorithm, because of its very extensive endings list. But in one way that is used to advantage: it is faster. It has effectively traded space for time, and with its large suffix set it needs just two major steps to remove a suffix, compared with the eight of the Porter algorithm.

The Porter's algorithm is a simpler version than Lovins algorithm. It uses 60 rules that are organized into sets. Conflicts within a set of rules are resolved before applying the next set. The rules are also separated into five distinct phases numbered 1 to 5 as shown below in fig. 2. They are applied to the words in a document from phase 1 moving on to phase 5.Each phase will remove a type of suffix of the word. After the five stages, the stem of the word will be left. Since Porter's algorithm is simpler and faster, we employed Porter's algorithm in our implementation.

During the process, all stop words and words having less than three letters will also be dropped, and any upper case characters will be changed to lower case[4].

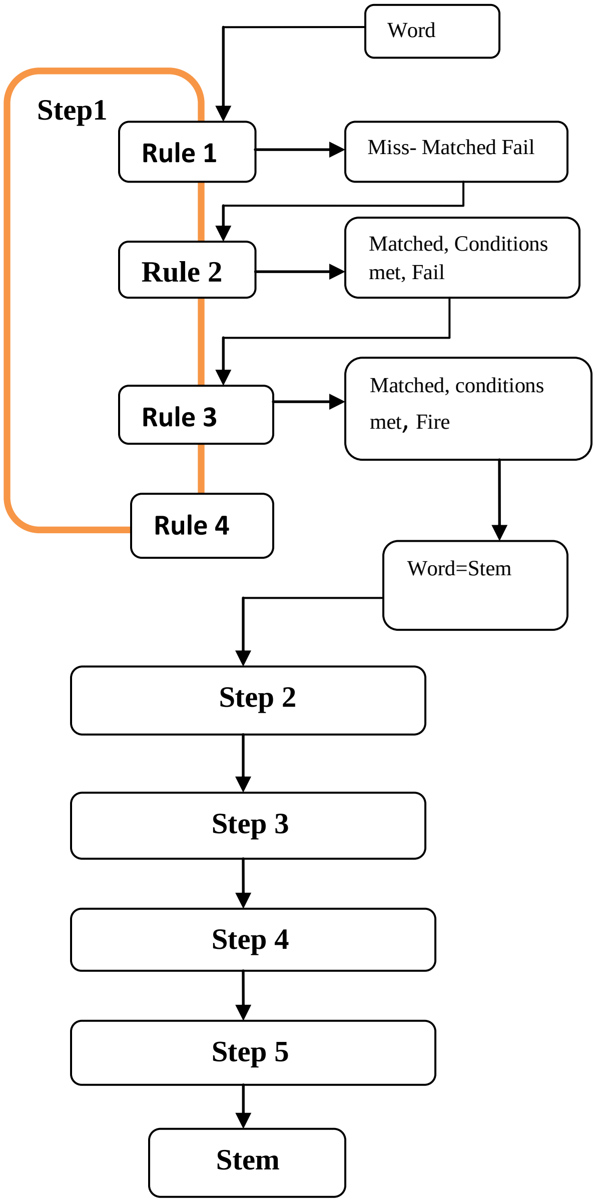

Fig 2. Porter Algorithm

C. Indexing:

This step involves the estimation of the distribution ("within document frequency") of all terms (literals and tags). Any path and term duplication is removed so that the resulting XML summary tree contains each path at most once. Thus the summary tree is smaller than the original XML document and it contains only those literal terms and tags that are important as far as indexing is concerned[6]. The term distribution may be in addition used to generate weights for each term in the summary tree in order to facilitate ranking.

D. Ranking:

This processing step deals with the loading of the summary trees into the index structure[6]. This involves the separation of content data from path data as shown below in fig.3. Content data is raw text aimed to be stored in the inverted file and path data is structured text aimed to be Stored in the path index. The path index is a hierarchy of tags, which records every single path in the collection.

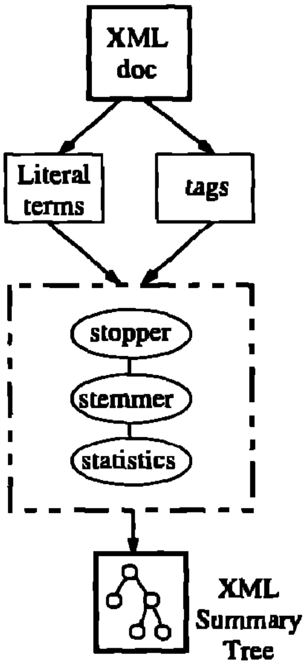

Fig 3. XML Document Normalization Process

E. Query Evaluation:

Structural terms are also generated for the user query after preprocessing (stop word removal and stemming) as shown in fig.4.These structural terms are then to be matched with the structural terms generated from XML documents according to formula to evaluate context – resemblance

$$\mathbf{cr}\ (q, d) = (1 + |q|)/(1 + |d|) \qquad \text{---}\ \ (1)$$

Where q and d are number of nodes in the query path and document path, respectively. The context resemblance function returns 0 if query path cannot be extended to the match of document path. Its value is 1.0 if q and d are identical .

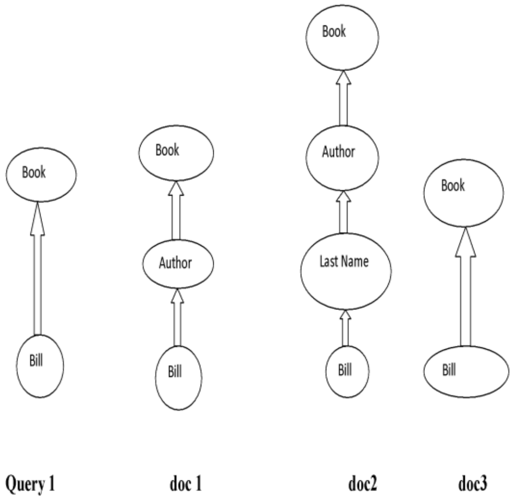

Fig.4. Query-document matching for extended queries. Extended queries match a document if the query path can be transformed into the document path by insertion of additional nodes. The context resemblance function cr is a measure of how similar two paths are. Here we have cr(q, d1) = 3/4 = 0.75, cr(q, d2) = 3/5 = 0.6 and cr(q,d3)=1

IV. RESULT

In this paper, an Information Retrieval system is developed for XML documents. XML documents are preprocessed and structural terms are generated for each document. These terms are then indexed .Also a ranking method is applied to rank the documents in response to a user query, ranked results are presented for the output Stopword Removal removes the unnecessary memory space and helps to increase quick search in the database. Using of stopword removal and stemming makes the

XML based database more reliable and efficient especially for heterogeneous platforms.

VI. CONCLUSION

XML significance is useful in web designing. Various Information Retrieval systems were analyzed. XML does a very good job of delivering self-describing data feeds it has become a key standard in Service Oriented Architectures (SOA) and Web Services. Sophisticated solution is provided. Relevant information retrieval model is developed for retrieving relevant information.

Tag transformation might be useful in order to handle variant words using sound-like methods. Eliminating suffixes and prefixes from the tags may further facilitate the content regularity. Devising techniques for handling proximity measures between tags can further increase the accuracy of XML information retrieval.

Scope for further research is towards tag similarity searching in order to address the problem of proximity searching and achieve more effective XML document ranking.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude towards our chairman Shri Dev Murti, Prof. Prabhakar Gupta(Dean), and Prof. S.P.Gupta (Director General). Without these members this manuscript is not possible. Under their cooperation we are able to make this manuscript successful.

We are also thankful to our Trust Administrator Er. Subhash Mehra. He helps us throughout this research paper.

Last but not the least we are heartily thankful to your institution which bestowed this opportunity to us.

REFERENCES

[1] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze “An Introduction to Information Retrieval” Cambridge University Press Cambridge, England, pp.195-237,April 2009.

[2] Gheorghe Muresan ,Bing Bai Conference RIAO2007 “Exploring Interactive Information Retrieval: An Integrated Approach to Interface Design and Interaction Analysis”, Pittsburgh PA, U.S.A. - Copyright C.I.D. Paris, France, pp. 1-3, June2007.

[3] Surajit Chaudhuri, et.al,“Probabilistic Information Retrieval Approach for Ranking of Database Query Results” ACM Transactions on Database Systems, Vol. 31, No. 3, pp. 1134–1168. September 2006.

[4]Willett Peter,“The Porter stemming algorithm: then and now.” Program: electronic library and information systems, Vol.40 (3). pp. 219-223,2006

[5] Norbert Fuhr and Kai Großjohann, “XIRQL: An XML Query Language Based on Information Retrieval Concepts” ACM Transactions on Information Systems, Vol.22, No.2, pp. 4-20, April 2004.

[6] Evangelos Kotsakis “Structured Information Retrieval in XML documents” ACM SAC2002, Madrid, Spain, pp. 664-666, 2002.

[7] Masatoshi Yoshikawa and Toshiyuki Amagasa, “XRel: A Path-Based Approach to Storage and Retrieval of XML Documents Using Relational Databases” ACM Transactions on Internet Technology Vol.1, No.1, pp. 2, August 2001.

[8] John D. Musa, “Software Reliability Engineering: More Reliable Software Faster and Cheaper”, Tata McGraw- Hill publication 2nd Edition,ISBN-13:978-0-07-060319-6, pp. 2-16, 265-284, 2005.

[9] Kothari, C.R., “Research Methods-Methods and Techniques”, New Age International(P) Limited, Publishers 2nd Edition, ISBN(13): 978-81-224-2488-1,pp. 1-52, 2004.