Interactive Search for Finding Duplicate Detection by Using Fuzzy Techniques in XML Data

R. Viswanath Reddy, G Pradeep Reddy
Mtech 2nd year , Dept of cse, jntuacea
Lecturer, Dept of cse, jntuacea

Abstract

Searching XML data has important utility in the real world. This is because the data in XML format can be understood by all development platforms across the world. It is a neutral format which is device independent and protocol independent. Therefore storing data in XML format has become a common practice in the industry. Searching such data needs the knowledge of contents. If the user is not aware of the content of the XML data, he cannot search effectively. To overcome this problem, Feng and Li proposed fuzzy type-ahead search for information retrieval from XML data sources by using query keywords. Their method has features such as high quality search and efficient indexing to make search process faster. In this paper we proposed a framework for searching XML data that enhances the fuzzy type ahead search process by combining it with both LCA and MCT based methods. The proposed framework thus makes XML search more robust and user friendly. We built a prototype application that demonstrates the verification of theory. The experimental results discovered to the proposed methodology is effective and feasible to build real world search applications on XML data sources.

Index Terms –XML, indexing structures, keyword search, fuzzy type-ahead search, LCA, MCT

INTRODUCTION

XML data is widely used as it supports all development platforms. When data is stored in XML format, there are traditional methods that make use of query languages to retrieve required information from XML data sources. The query languages include XQuery and XPath. These methods are efficient in retrieving data easily. However, novice users will not be able to frame queries easily with these languages as their syntax is not easy. There were many researches that contributed to the study of challenges pertaining to querying XML data in the genuine earth[1], [2], [3], [4], [5], [6], [7], [8], [9]. With keyword query applications, user feels easy to enter search words and get the required results. This is because the keyword search makes it simple and the users need not to remember the syntaxes of XQuery and XPath. Even they need not to have knowledge on the structure of XML data. One important drawback with keyword search applications is that user needs to have some knowledge on the underlying data. Otherwise user will not be able search properly. Thus those search interfaces are neither user friendly nor efficient. IN order to solve this problem, various features came into existence. They include Autocomplete which automatically fills the words started by end users. Almost all search engines have come up with this facility. The autocomplete treats multiple keywords as single word for searching. This is its drawback. To overcome this problem Bast and Weber [10] and [11] proposed a solution known as Complete Search that finds related answers with different keywords. However, it does not provide approximate search facility. Recently in [12] fuzzy type ahead search was introduced for text documents. This will help users to provide inputs easily as the system gives hints on data. This kind of search is also present in databases [13]. However the existing methods that are meant for searching XML documents do not have the facility of type – ahead search. XML document has data in tree format. Search such content needs efficient query processing approach.

As seen in figure 1, it is evident that the XML file has content that is hierarchical in nature. It is formed as a tree. In [14] fuzzy type ahead search was introduced. It helps users to know the words available in the XML documents beforehand so as to make efficient searches. This is particularly useful as XML data is generally not known to end users. The authors have provided friendly interface to make fuzzy ahead search. It has algorithms and also indexing structures to achieve this.

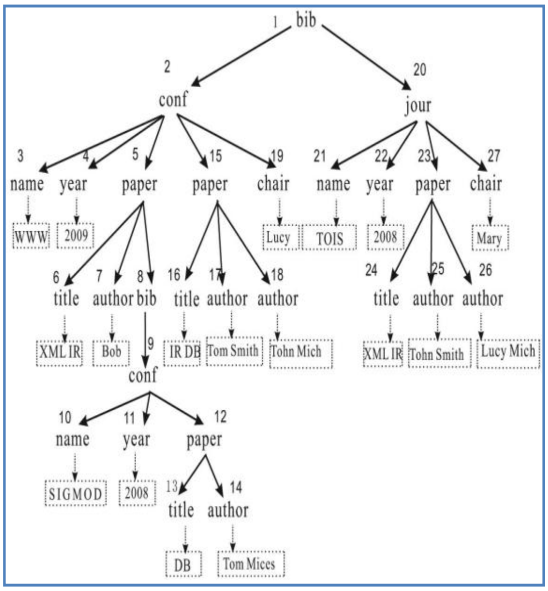


Fig. 1 –Tree structure of an XML document

In this paper we proposed a framework for searching XML data that enhances the fuzzy type ahead search process by combining it with both LCA and MCT based methods. The proposed framework thus makes XML search more robust and user friendly. We built a prototype application that demonstrates the evidence of theory. The observed outcome discovered that the proposed methodology is effective and feasible to build real world search applications on XML data sources. The remainder of this paper is structured as follows. Section II provides details of proposed framework. Section III presents prototype implementation details. Section IV presents experimental results while section V concludes the paper.

PROPOSED FRAMEWORK

The proposed framework is an extension to the concepts introduced in [14]. The framework is meant for improving fuzzy type-ahead search. This is basically used to help users to make efficient queries without having knowledge of underlying data in XML data sources. Our approach is to make use of LCA and MCT methods along with fuzzy type ahead search in order to improve the performance further.

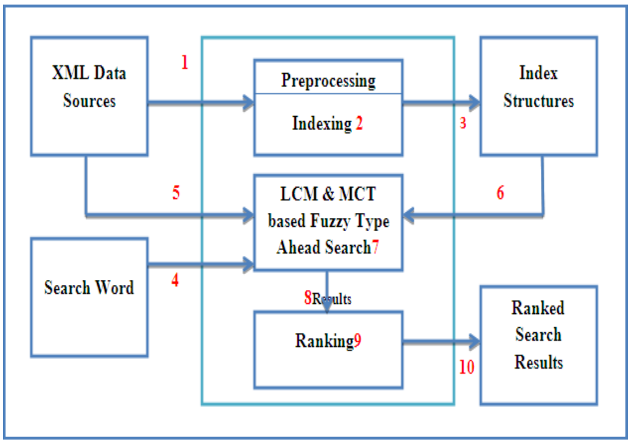


Fig. 2 – Proposed framework

The proposed framework has indexing concept that takes care of creating index structures through pre-processing. The index structures can help improve the speed of query processing. The LCA & MCT based fuzzy type ahead search component takes input from user in the form of search word. It also takes XML data sources, index structures for processing the query efficiently. After obtaining search results they are ranked to provide end users with top-n results that help them to choose based on ranking. Lowest Common Ancestor (LCA) and Minimal Cost Trees are the two methods that have been combined with fuzzy type ahead search to improve the performance. Once minimal cost tree is built successfully, it can be ranked for improved search results. The similarity between words is found as follows.

$$sim(k_i, w_i) = \gamma * \frac{1}{1 + \mathrm{ed}(k_i, a_i)^2} + (1 - \gamma) * \frac{|a_i|}{|w_i|},$$

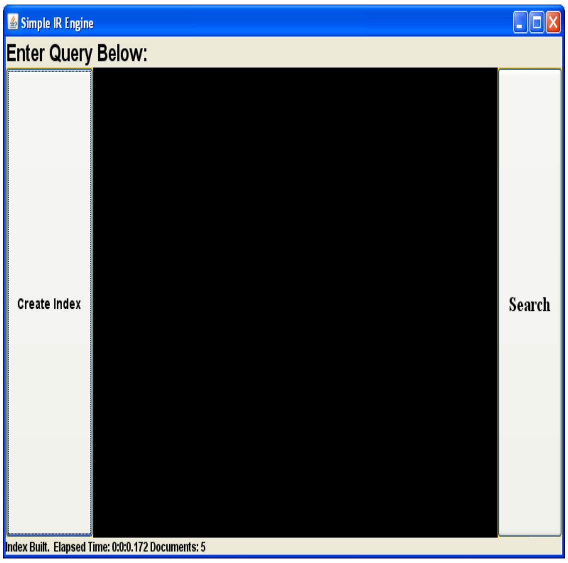
The final ranking is computed as follows.

$$\mathrm{SCORE}(n, Q) = \sum_{i=1}^{\ell} sim(k_i, w_i) * \mathrm{SCORE}(n, w_i).$$

With ranking in place the search results are presented in very useful fashion. Users will get top –n kind of query results that will help them to get the most important ones at the top. This makes the search results more user-friendly. The query processing is improved with the combination of LCA and MCT. More technical details can be found in [14].

PROTOTYPE APPLICATION

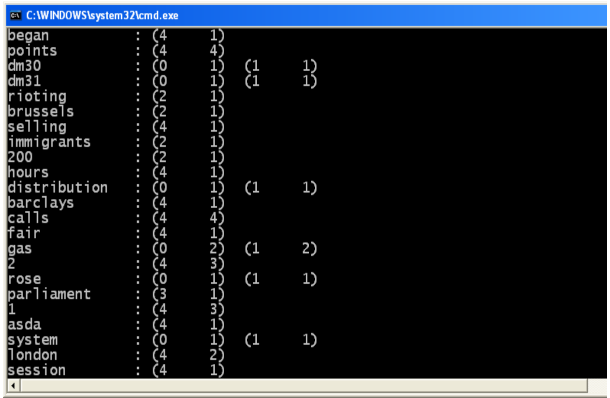
The prototype application is user-friendly with graphical user interface. It has been built in Java platform with SWING API and XML API besides other API for information retrieval. The application facilitates users to search for required contents without prior knowledge on the underlying data in XML data sources. The application is built in the environment where a PC with 4 GB RAM, core 2 dual processing running Windows 7 operating system is used. The main user interface is as shown in figure 3.



Index Built. Elapsed Time: 0:0:0.172 Documents: 5

Fig. 3 –the main UI of the application

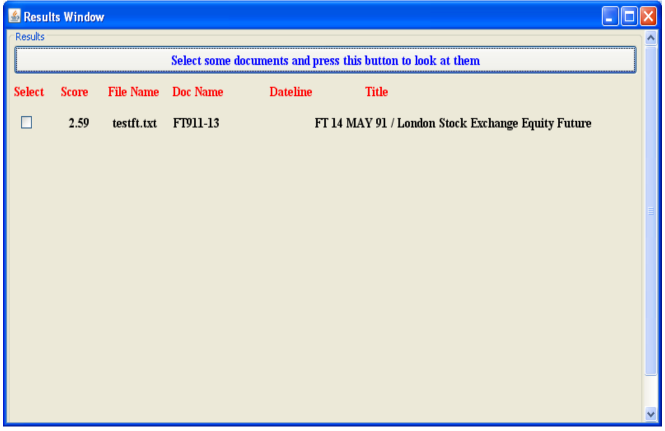
As can be seen in figure 3, the application provides user interface for two activities that are important for this application. They are creating index structures and performing search. When user clicks on “Create catalog” key, the index is performed and the catalog structures are generated as explained in the previous section. Once indexing is ready fuzzy type ahead search is possible that allows the user to choose a word and perform search. The fuzzy type ahead search is user friendly and makes use of techniques such as LCA and MCT in order to make the search process more robust. Then the results are ranked in order to show top k results to end users. Figure 4 shows the verbose output generated while creating index structures. These outputs allow ending users to get out the words type ahead fashion. This will eliminate the need for having prior knowledge in XML data sources.



```
began          : (4     1)
points         : (4     4)
dm30           : (0     1)  (1     1)
dm31           : (0     1)  (1     1)
rioting        : (2     1)
brussels       : (2     1)
selling        : (4     1)
immigrants     : (2     1)
200            : (2     1)
hours          : (4     1)
distribution   : (0     1)  (1     1)
barclays       : (4     1)
calls          : (4     4)
fair           : (4     1)
gas            : (0     2)  (1     2)
2              : (4     3)
rose           : (0     1)  (1     1)
parliament     : (3     1)
1              : (4     3)
asda           : (4     1)
system         : (0     1)  (1     1)
london         : (4     2)
session        : (4     1)
```

Fig. 4 – Verbose output generated while building index structures

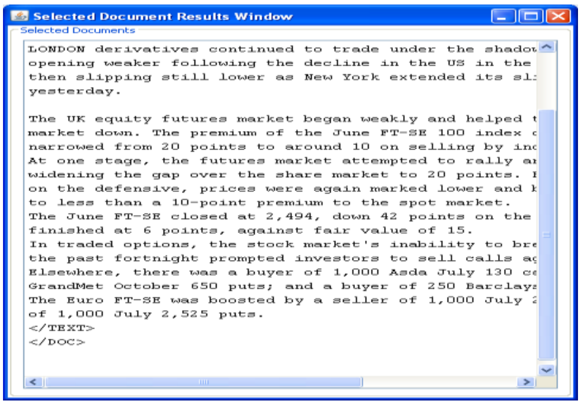
As can be seen in figure 4, it is evident that the application generates index structures that are used in search process when initiated by end user. On choosing search after ending search word, the system performs fuzzy type ahead search and produces output. The results of sample search are shown in figure 5.



Select	Score	File Name	Doc Name	Dateline	Title
□	2.59	testft.txt	FT911-13		FT 14 MAY 91 / London Stock Exchange Equity Future

Fig. 5 –Search results

As seen in figure 5, the search results are shown. After viewing search results in ranked order, the user can choose one of more search results to view actual XML documents that have been into the search results. For instance the figure 6 appears when the result row is selected.



```
LONDON derivatives continued to trade under the shadow
opening weaker following the decline in the US in the
then slipping still lower as New York extended its sl
yesterday.

The UK equity futures market began weakly and helped t
market down. The premium of the June FT-SE 100 index c
narrowed from 20 points to around 10 on selling by inc
At one stage, the futures market attempted to rally ar
widening the gap over the share market to 20 points. I
on the defensive, prices were again marked lower and k
to less than a 10-point premium to the spot market.
The June FT-SE closed at 2,494, down 42 points on the
finished at 6 points, against fair value of 15.
In traded options, the stock market's inability to bre
the past fortnight prompted investors to sell calls ag
Elsewhere, there was a buyer of 1,000 Asda July 130 ca
GrandMet October 650 puts; and a buyer of 250 Barclays
The Euro FT-SE was boosted by a seller of 1,000 July 2
of 1,000 July 2,525 puts.
</TEXT>
</DOC>
```

Fig. 6 –Search result in the form of an XML document

As seen in figure 6, the search result is shown. The XML document is the result of the search process. User can view one or more XML files that come in results. The results are ranked so as to allow end users to view best results.

EXPRIMENTAL RESULTS

Experiments are made in terms of queries, elapsed time, edit distance threshold, expected time, number of publications and index size, number of publications vs. search time, for LCA, MCT and fuzzy type ahead search that combines both of the techniques. The experimental results are presented below.

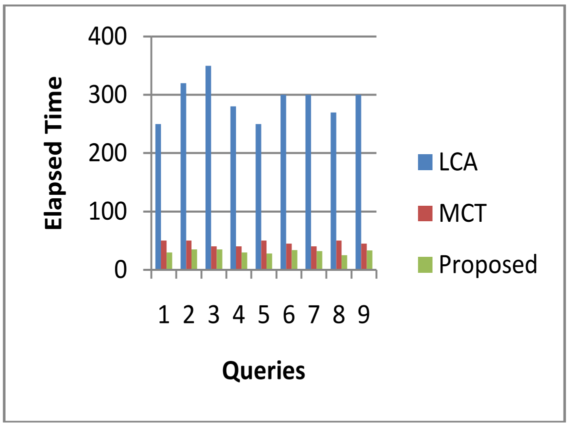


Fig. 7 -Search time (LCA versus MCT) Exact search.

As given away on peak of shape straight alignment represents queries while vertical axis represents elapsed time.

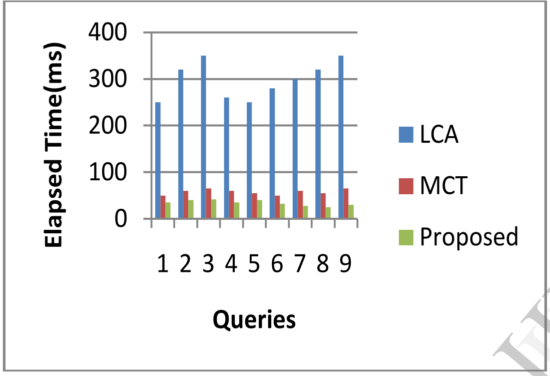


Fig. 8- Search time (LCA versus MCT) Fuzzy search

As shown in the above figure horizontal axis represents queries while vertical axis represents elapsed time.

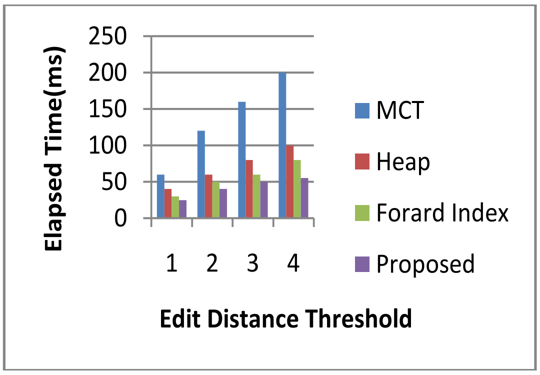


Fig. 9 - Search time (using Max heap and Forward Index) XMark data set.

As shown in the above figure horizontal axis represents edit distance threshold while vertical axis represents elapsed time.

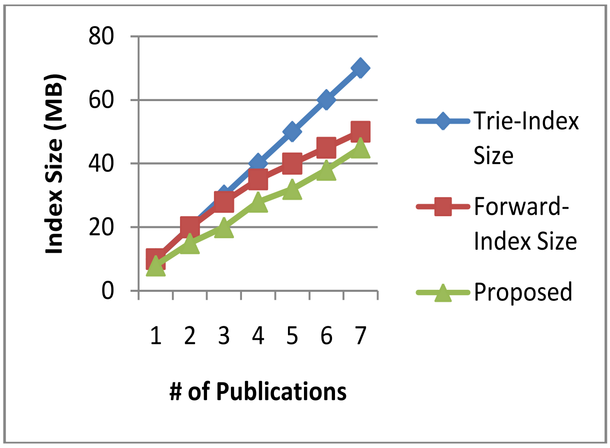


Fig. 10 - Scalability on DBLP data set by varying numbers of selected publications, 100K, 200K, . . 1M Index size

As shown in the above figure horizontal axis represents maximum of publications while vertical axis represents index size.

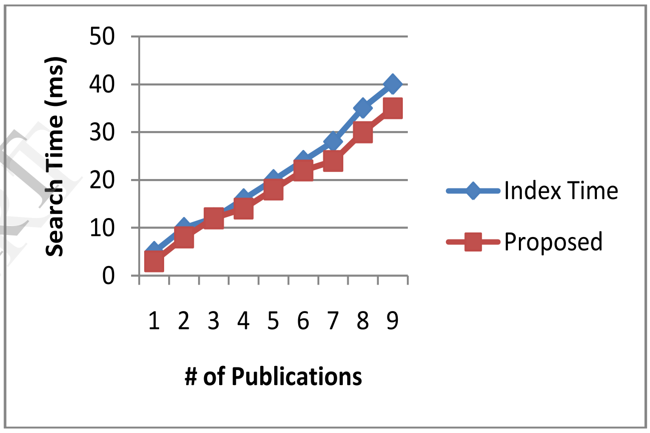


Fig. 11- Scalability on DBLP data set by varying numbers of selected publications, 100K, 200K, . . . , 1M. Index time.

As shown in the above figure horizontal axis represents maximum of publications while vertical axis represents search time.

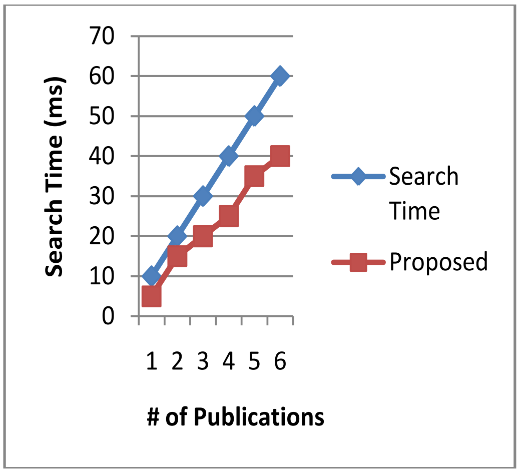


Fig. 12 - Scalability on DBLP data set by varying numbers of selected publications, 100K, 200K, . . . , 1M. Search time.

As shown in the above figure horizontal axis represents maximum of publications while vertical axis represents search time.

CONCLUSION

In this paper we are studying the problem of search for XML data. We have improved the fuzzy type ahead search mechanism proposed by Feng and Li[14]that has both efficient algorithms and index structures that allow users to search for XML data without having prior knowledge on data. In this paper we use their approach and also combine the LCA based and MCT based methods to make the search process more robust and user friendly. The indexing structure we used improves the performance of search process. We also built a prototype application that demonstrates the efficiency of the planned move toward. The experimental results exposed that the proposed framework is useful and feasible to be used with real world search systems that operate on XML data sources.

REFERENCES

[11] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "Xrank: Ranked Keyword Search over Xml Documents," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 16-27, 2003.

[2] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "Xsearch: A Semantic Search Engine for Xml," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 45-56, 2003.

[3] Y. Li, C. Yu, and H.V. Jagadish, "Schema-Free Xquery," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 72-83, 2004.

[4] Y. Xu and Y. Papakonstantinou, "Efficient Keyword Search for Smallest Lcas in XML Databases," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 537-538, 2005.

[5] C. Sun, C.Y. Chan, and A.K. Goenka, "MultiwaySlca-Based Keyword Search in Xml Data," Proc. Int'l Conf. World Wide Web (WWW), pp. 1043-1052, 2007.

[6] G. Li, J. Feng, J. Wang, and L. Zhou, "Effective Keyword Search for Valuable lcas over XML Documents," Proc. Conf. Information and Knowledge Management (CIKM), pp. 31-40, 2007.

[7] Z. Liu and Y. Chen, "Identifying Meaningful Return Information for Xml Keyword Search," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 329-340, 2007.

[8] Y. Xu and Y. Papakonstantinou, "Efficient LCA Based Keyword Search in XML Data," Proc. Int'l Conf. Extending Database Technology: Advances in Database Technology (EDBT), pp. 535-546, 2008.

[9] G. Li, C. Li, J. Feng, and L. Zhou, "Sail: Structure-Aware Indexing for Effective and Progressive Top-k Keyword Search over XML Documents," Information Sciences, vol. 179, no. 21, pp. 3745-3762, 2009.

[10] H. Bast and I. Weber, "The Completesearch Engine: Interactive, Efficient, and towards Ir&db Integration," Proc. Biennial Conf. Innovative Data Systems Research (CIDR), pp. 88-95, 2007.

[11] H. Bast and I. Weber, "Type Less, Find More: Fast AutocompletionSearch with a Succinct Index," Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 364-371, 2006.

[12] S. Ji, G. Li, C. Li, and J. Feng, "Efficient Interactive Fuzzy Keyword Search," Proc. Int'l Conf. World Wide Web (WWW), pp. 371-380, 2009.

[13] G. Li, S. Ji, C. Li, and J. Feng, "Efficient Type-Ahead Search on Relational Data: A Tastier Approach," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 695-706, 2009.

[14] Jianhua Feng and Guoliang Li, "Efficient Fuzzy Type-Ahead Search in XML Data". IEEE Transactions on Knowledge and Data Engineering, Vol. 24, NO. 5, MAY 2012.