ISSN: 2321-7782 (Online)
Volume 2, Issue 3, March 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Document Retrieval by Using Fuzzy Keyword Search in XML Data

M. Swapna[1]
Computer Science and Engineering
SVU College of Engineering
Tirupati – India

N. Usha Rani[2]
Computer Science and Engineering
SVU College of Engineering
Tirupati – India

Abstract: ***Now a day's most of the search engines store the data in XML documents. The stored data is retrieved by using query languages. The present work focuses on the retrieving of required results with the help of fuzzy logic. So, it is easy to identify the required data under uncertainty also. In this user just give keyword and searches the data by using relevancy value between the keywords in the XML data matching with the query keywords approximately. Then ranking algorithm is used to find top most results. The Present work provides user friendly interface while giving queries. The present work compares the efficiency in retrieving XML document by using TA and NRA algorithm and the experimental results shows that the execution time is reduced in NRA algorithm when compared to Threshold algorithm.***

Keywords: XML, XKeyword, Fuzzy Search, Relevancy.

I. INTRODUCTION

XML stands for Extensible Markup Language.XML is used to store and transport data.XML document shall have one root element and root element have child elements. Tags are used to store the data and the tags are user defined. After storing the data the tags must be closed. In general, XML data contains parent child relationship, and we need to id entify the relevant sub trees that capture such structural relationships from XML data to answer queries. In XML, data can be searched by using keyword search. It is widely accepted search paradigm for querying document systems and the World Wide Web. One important advantage of keyword search is that it enables users to search information without having prior knowledge about the structure of the data. In this user composes a keyword query, submits it to the system, and retrieves relevant answers by using query languages. In the case where the user has limited knowledge about the query language and the data, often the user feels difficulty when issuing queries. To improve the search efficiency Fuzzy keyword search is proposed.

Fuzzy Logic is a logic system for reasoning that are approximate rather than exact .The fundamental unit of a fuzzy logic is the fuzzy set. Fuzzy logic defines a membership function A: X– > [0, 1] that maps element x of X into real numbers in [0, 1]. 1 denotes that the element is completely in the set, 0 denotes that the element is not in the set and a number in between means partially in the set. Membership value denotes that how much it is related to the set. Fuzzy keyword search allows the users to search the data even in the presence minor errors in the query keywords. The data is retrieved by using relevancy value. It denotes how much the query keyword is related to the keywords in XML documents. It is calculated by using edit distance and best similar prefixes. Then NRA (No Random Access) algorithm is used to find the top most results. So, top most related answers are retrieved.

II. RELATED WORK

Extensive research efforts have been conducted in XML keyword search to find the XML data. XSearch [1] is one of the search engines; it has a simple query language, suitable for a naive user. It returns semantically related document fragments that satisfy the user's query. Query answers are ranked using extended information retrieval techniques and are generated in an order

similar to the ranking .Then the problem of efficiently producing ranked results for keyword search was considered in XRANK system [2].The notion of proximity among keywords is more complex for XML. In HTML, proximity among keywords translates directly to the distance between keywords in a document.

However, for XML, the distance between keywords is just one measure of proximity; the other measure of proximity is the distance between keywords and the result is XML element. XRANK system that is designed to handle these novel features of XML keyword search. An interesting feature of XRANK is that it naturally generalizes a hyperlink based HTML search engine such as Google. XRANK can thus be used to query a mix of HTML and XML documents. To provide efficient keyword proximity in XML, XKeyword [3] system is implemented. XKeyword is built on a relational database, it can also accommodate to very large graphs. Query evaluation is optimized by using the graph's schema. In particular, XKeyword consists of two stages. In the preprocessing stage a set of keyword indices are built along with indexed path relations that describe particular patterns of paths in the graph. In the query processing stage plans are developed that use a near optimal set of path relations to efficiently locate the keyword query results. The results are presented graphically using the idea of interactive result graphs, which are populated on-demand according to the user's navigation and allow efficient information discovery.

A framework for describing semantic [4] relationships among nodes in XML documents is presented. The XML documents may have ID references. A specific inter-connection semantics are defined explicitly or derived automatically. The main advantage of interconnection semantics is the ability to pose queries on XML data in the style of keyword search. The interconnection semantics can be efficiently applied to real-world XML documents. The problem of effective keyword search over XML documents can also be solved by using Valuable Lowest Common Ancestor (VLCA)[5].Then the concept of Compact VLCA (CVLCA)[6] and compute the meaningful compact connected trees rooted as CVLCAs as the answers of keyword queries. Then Interactive fuzzy search [7] has been implemented, in which the system searches the underlying data "on the fly" as the user types in query keywords. It extends auto complete interfaces by allowing keywords to appear in multiple attributes of the underlying data and finding relevant records that have keywords matching query keywords approximately. XReal [8] is one of the methods which provide keyword query results by using relevance oriented ranking. XML TF*IDF ranking strategy is used to rank the individual matches of all possible search intentions.

III. Document Retrieval by using Fuzzy Keyword Search

In this paper, XML document is retrieved by using Fuzzy Keyword Search [9]. It uses fuzzy logic to search the required data, so it is easy to identify the required data even in presence of errors in the keywords. In this User does not require the knowledge about the content, user just gives the keyword and retrieves the related data by using relevancy value between the keywords in XML Data matching with the query keywords approximately. Then NRA algorithm is used to find top most results to the query. So, it is easy to identify the required data without knowledge of the content.

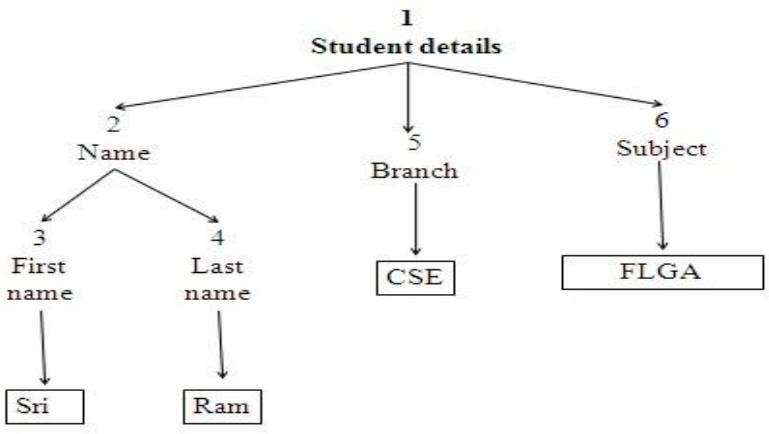

Fig 1: XML Tree

A. Index Structure

The present work uses the trie index structure to index the words in the XML data. In Trie index structure, each word corresponds to a unique path from the root of the trie node to a leaf node.Each node on the path has a label of character in the word. For each leaf node, the list of IDs of XML elements that contain the word of the leaf node are stored.

The trie structure for that XML tree will be as follows:

For example consider the keyword “sri”, In the trie index each node contains a label i.e. “s” is stored in one node, “r” is stored in next node and “i” is stored next node. And the value “3” in the box denotes that the keyword “sri” is stored in XML tree at node 3.

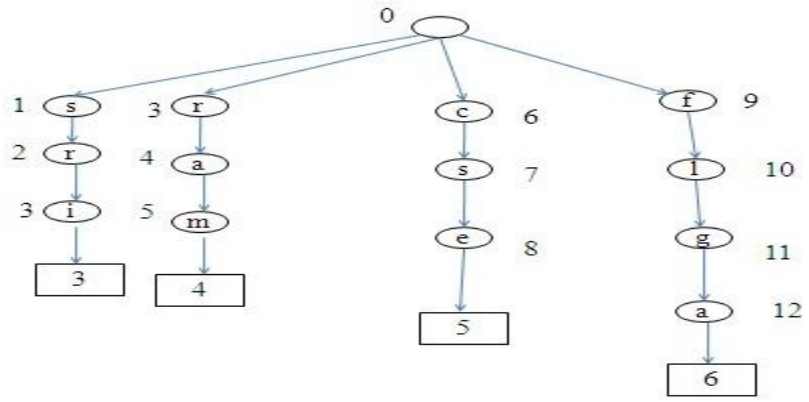

Fig 2 : Trie structure for the XML tree

B. Queries With Multiple Keywords

When user types a query, the query string is tokenized into keywords k_1, k_2 … k_l. For each keyword k_i corresponding active nodes will be computed, and for each active node, its leaf descendants and corresponding IDs lists are retrieved. Then union of active nodes is computed to answer the query.

C. Fuzzy Search

In exact search, the data if the tree contains the exact keywords. But in Fuzzy search, the Tree may not contain the exact input keywords, but contain the predicted words also data is retrieved. The system retrieves the related data by using relevancy [9] value between the keywords in XML Data matching with the query keywords approximately. Relevancy or Score [9] is calculated as follows.

$$Score_1(n, k_i) = \frac{\ln(1+tf(k_i,n))*\ln(idf(k_i))}{(1-s)+s*ntl(n)}$$

s is constant , it is set to 0.2.

tf (k_i , n) is the number of occurrences of keyword k_i in the sub tree rooted at n.

idf (k_i) is the inverse document frequency of k_i , i.e. ratio of the number of nodes in XML document to the number of nodes that that contain k_i .

$$ntl(n) = \frac{|n|}{|n_{max}|}$$

ntl is normalized term length

|n| is the number of terms contained in n .

$|n_{max}|$ denotes the node with the maximal number of terms.

ntl (n) is usually 1.

Score $_2(n, k_j) = \sum_{p \in P} \alpha^{\delta(n,p)} * \text{Score}_1(p, k_j)$

k_j is keyword,

p is the pivotal node for n and k_j .

α is the damping factor between 0 and 1.

δ(n, p) denotes the distance between n and p.

P is the set of pivotal nodes for n and k_j .

$Q = \{k_1, k_2, \ldots., k_l\}$

$$\text{Score}(n, Q) = \sum_{i=1}^{l} \text{Score}(n, k_i)$$

$$\text{Sim}(k_i, w_i) = \Upsilon * \frac{1}{1+ed(k_i, w_i)^2} + (1-\Upsilon) * \frac{|a_i|}{|w_i|}$$

k_i is partial keyword and users may type in more letters to complete the keyword .

w_i is similar prefix, a_i best similar prefix and Υ is the tuning parameter it is in between 0 and 1.

$$\text{Score}(n,Q) = \sum_{i=1}^{1} \text{Sim}(k_i, w_i) * \text{Score}(n, w_i)$$

D. Ranking

The present work uses NRA algorithm to find the top most results. In the trie index, for each leaf trie node, content nodes and quasi-content nodes in the XML document, corresponding relevancy values and pivotal paths for the keyword of the leaf node are maintained to answer the query. Given a keyword query Q, for each partial keyword k_i, first its predicted words are computed. Then, the union of inverted lists of k_i's predicted words U_{ki} is computed. Finally, NRA algorithm is used to compute the top most results of on top of every $U_{ki.}$.

NRA algorithm:

- Start
- Access all elements in parallel.
- At each access calculate the lower bound and upper bound of the all the elements.
- Compare the element lower bound values with the upper bound values of all the elements.
- Return the elements for which the lower bound is higher than the upper bound bound of other objects.
- The returned elements are the top most results.
- Stop

IV. EXPERIMENTAL RESULTS

The present work uses student detail in XML data format which contains first name, last name, branch, year and subject details. In this user give query as keyword and searches the related results to the query by using relevancy value. And then the results are ranked by using NRA algorithm. In the present work, NRA algorithm has implemented for reduction of execution time. It retrieves top most results quickly.

Table 1: Comparison of TA and NRA algorithm

Type of Query	Execution Time (milli seconds)	
	TA	NRA
Cse	300	140
Networks	250	190
Programming	400	240
Finite	150	90
Ram	300	220
Sri	180	120
Sandhya	320	250
Mavalluru	240	180

From the table it has been observed that, in threshold algorithm answers are retrieved within an average of 277ms where as in NRA algorithm answers are retrieved within an average of 182 ms. Comparison of execution time between Threshold algorithm and NRA algorithm is pictorially represented as follows:

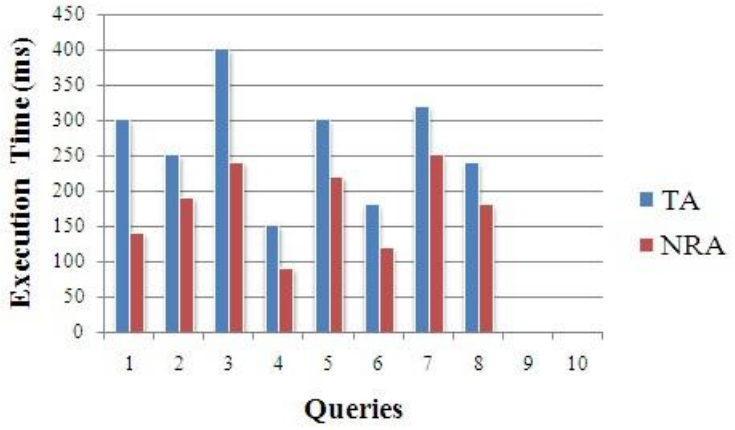

Fig 3: Comparison of TA and NRA

V. CONCLUSION

In the present work XML document is retrieved by using index structures and relevancy values and then NRA algorithm is used to rank the results. The present work observes that the reduction of time for executing the query when compared to Threshold algorithm for displaying top most results.

References

1. S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "XSearch: A Semantic Search Engine for Xml," Proceedings of International Conference on Very Large Data Bases, pp. 45-56, 2003.
2. L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "Xrank: Ranked Keyword Search over Xml Documents," Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 16-27, 2003.
3. V. Hristidis, Y. Papakonstantinou, and A. Balmin, "Keyword Proximity Search on XML Graphs," Proceedings of International Conference on Data Engineering (ICDE), pp. 367-378, 2003.
4. S. Cohen, Y. Kanza, B. Kimelfeld, and Y. Sagiv, "Interconnection Semantics for Keyword Search in Xml," Proc. Int'l Conf. Information and Knowledge Management (CIKM), pp. 389-396, 2005.
5. G. Li, J. Feng, J. Wang, and L. Zhou, "Effective Keyword Search for Valuable LCAs over XML Documents," Proceedings of Conference on Information and Knowledge Management (CIKM), pp. 31-40, 2007.
6. Y. Xu and Y. Papakonstantinou. Efficient Keyword Search for smallest LCAs in XML databases. In SIGMOD, pages 527–538, 2005.
7. S. Ji, G. Li, C. Li, and J. Feng, "Efficient Interactive Fuzzy Keyword Search," Proceeding of Conference on World Wide Web (WWW), pp. 371-380, 2009.
8. Z. Bao, T.W. Ling, B. Chen, and J. Lu, "Effective XML Keyword Search with Relevance Oriented Ranking," Proceedings of International Conference on Data Engineering (ICDE), 2009.
9. Jianhua Feng, Guoliang Li, "Efficient Fuzzy Type-Ahead Search in XML data," IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 5, May 2012

AUTHOR(S) PROFILE

M. Swapna received B.Tech degree in Computer Science and Engineering from Gokula Krishna College of Engineering, JNTUA, Anantapur, A.P, India in 2011 and currently pursuing M.Tech, Computer Science and Engineering, final semester, from Sri Venkateswara University College of Engineering, Tirupati, A.P, India. Her interested areas are Data Mining and Fuzzy Logic. She attended Two National Conferences during 2013 and 2014.

N. Usha Rani is working as Assistant Professor in the department of Computer Science and Engineering, SVU College of Engineering, Sri Venkateswara University, Tirupati. She did B.Tech in CSE, JNTU, Hyderabad. She did M.Tech in Artificial Intelligence, University of Hyderabad, Hyderabad. She is pursuing Ph.D in the University of Hyderabad, Hyderabad. She published papers in International Journals. She has presented papers in International and National Conferences. Her research areas are Speech Processing, Data Mining, Artificial Neural Networks, Fuzzy Logic, Genetic Algorithms and Machine Learning.