Interactive Search over XML Data to Obtain Top-K Results

Ch. Lavanya Susanna

Abstract- Internet search engines are much popularized keyword search paradigm. However the search engine that uses html based model does not capture more semantics. But the xml model captures more semantics and navigates into document and displays more relevant information. The keyword search is alternative method to search in xml data, which is user friendly, user no need to know about the knowledge of xml data and query languages. This paper focuses on the survey of techniques used to retrieve the top k results from the xml document more efficiently.

Keywords: xml, html, keyword search, xml data

I. INTRODUCTION

A keyword search looks for words anywhere in the record. It is emerged as most effective paradigm for discovering information on web. The advantage of keyword search is its simplicity-users do not have to learn complex query language and can issue query without any knowledge about structure of xml document. The most important requirement for the keyword search is to rank the results of query so that the most relevant results appear. Keyword search provides simple and user friendly query interface to access xml data in web. Keyword search over xml is not always the entire document but deeply nested xml. Xml was designed to transport and store data. It does not do anything, it is created to structure, store, and transport information.xml document contains text with some tags which is organized in hierarchy with open and close tag.xml model addresses the limitation of html search engine i.e. Google which returns full text document but the xml captures additional semantics such as in a full text titles, references and subsections are explicitly captured using xml tags. For querying xml data keyword search is proposed as an alternative method. In traditional approach to query over xml data it requires query languages which are very hard to comprehend for non database users. It can only understand by professionals. Recently database community has been studying challenges related to keyword search over xml data[1]. However the traditional approaches are not user friendly. To solve this problem many systems introduced various features. One method id Autocomplete which predicts the words the user had typed in. More and more websites support these features example Google, yahoo.

One limitation of this approach is it treats multiple key words as single key word and do not allow them to appear in different places. To address this problem other method is proposed complete search in textual documents which allows multiple keywords to appear in different places but it does not allow minor mistakes in query.

Recently fuzzy type ahead search [2] is studied which allows minor mistakes in query. Type ahead search is a user interface interaction method to progressively search for filter through text. As the user types text, one or possible matches for text are found and immediately present to user. The fuzzy type ahead search in xml data returns the approximate results. The best similar prefixes are matched and returned. For this edit distance is used. Edit distance is defined as number of operations (delete, insert, substitute) required to make the two words equal. For example user typed the query ”mices” but the mices is not in the xml document it contains miches ed(mices, miches) is 1 so therefore the best similar prefix is miches it is displayed.

II. METHODS FOR KEYWORD SEARCH OVER XML DATA

1. LCA based method

The **l**owest common ancestor (LCA) is a concept in graph theory and computer science. Let *T* be a rooted tree with *n* nodes. The lowest common ancestor between two nodes *v* and *w* is defined as the lowest node in *T* that has both *v* and *w* as descendants

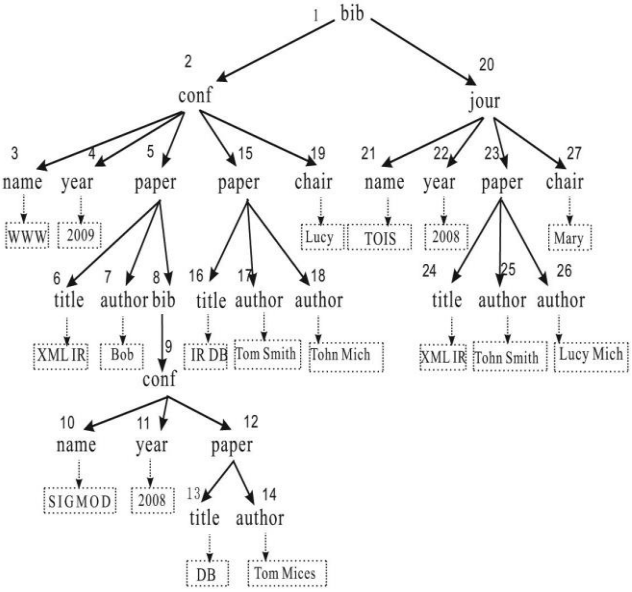

Fig 1: xml document

The LCA of *v* and *w* in *T* is the shared ancestor of *v* and *w* that is located farthest from the root. There are different ways to answer the query on an xml document; one commonly used method is LCA based method [3]. Many algorithms that use query over xml uses this method. Content nodes are the parent node of the keyword. For example consider keyword db in fig1 then content node of db is node 13 and node16.

Manuscript Received on July, 2013

Ch.LavanyaSusanna, Department Of Computer Science and Engineering, KLUniversity, India.

The server contains index structure of xml document which each node is letter in keyword and leaf node contain all nodes that contain the keyword this leaf node is called inverted list.

Procedure

- For keyword query the LCA based method retrieves content nodes in xml that are in inverted lists.
- Identify the LCAs of content nodes in inverted list
- Takes the sub tree rooted at LCAs as answer to the query

for example suppose the user typed the query "www db" then the content nodes of db are{13,16} and for www are3 ,the LCAs of these content nodes are nodes ,12,15,2,1.here the nodes 3,13,12,15 are more relevant answers but nodes 2 and 1 are not relevant answers.

Limitation

- It gives irrelevant answers
- The results are not of high quality

2. ELCA based method

To address the limitation of LCA based method exclusive LCA (ELCA)[4] is proposed. It states that an LCA is ELCA if it is still an LCA after excluding its LCA descendents. for example suppose the user typed the query "db tom" then the content nodes of db are{13,16} and for tom are{14.17} ,the LCAs of these content nodes are nodes2,12,15,1.here the ELCAs are 12,15.the subtree rooted with these nodes is displayed which are relevant answers Node 2 is not an ELCA as it is not an LCA after excluding nodes 12 and 15. Xu and Papakonstantinou [55] proposed a binary-search-based method to efficiently identify ELCAs.

III. PROGRESSIVELY COMPUTING TOP K ANSWERS USING FUZZY SEARCH OVER XML DATA

3.1 Architecture

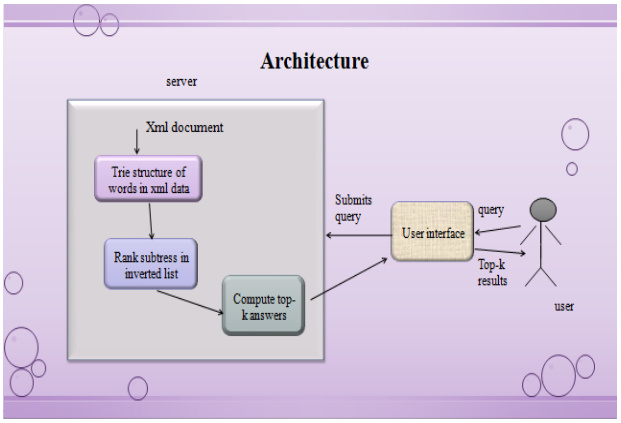

Fig 2 architecture for compute top k answers

The user composes query and submits. The browser sends the query to the server where the xml document and its index structure lies in it. The ranking module ranks the nodes that contain the keyword. The compute top k results uses effective algorithms to compute top k results and the server returns the results to user through browser.

3.2 Ranking the subtree

There are two ranking function to compute rank/score between node n and keyword ki[5]

1) The case that n contains ki.
2) The case that n does not contain ki but has a descendant containing ki.

Case 1: n contains keyword ki

The relevance/score of node n and keyword ki is computed by

$$\mathrm{SCORE}_1(n,k_i)=\frac{ln(1+tf(k_i,n))*ln(idf(k_i))}{(1-s)+s*ntl(n)}.$$

Where

tf(ki,n) - no:of occurences of ki in subtree rooted n

idf(ki) - ratio of no:of nodes in xml to no:of nodes that contain keyword ki

ntl(n) - length of n /nmax length, nmax=node with max terms

s - Constant set to 0.2

Assume user composed a query containing keyword "db"

score(13,db)= ln(1+1)*ln(27/2)

(1- 0.2)+(0.2*1)

= 1.52

Case 2: node n does not contain keyword ki but its descendant has ki

Second ranking function to compute the score between n and kj is

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

Where

P - Set of pivotal nodes

α - constant set to 0.8

$\delta(n,p)$ - Distance between n and p

Assume the user composed query "db"

Score2 (12, db) = (0.8)*score1 (13, db)

= 0.8 *1.52

=1.21

3.3 Ranking fuzzy search

Given a keyword query Q={k1,k2,…..kl} in terms of fuzzy search, a minimal-cost tree may not contain the exact input keywords, but contain predicted words for each keyword.Let predicted words be {w1,w2…..wl}the best similar prefix of wi could be considered to be most similar to ki.The function to quantify the similarity between ki and wi is

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

where ed – edit distance, ai – prefix, wi – predicted word,γ – constant

3.4 Progressively compute top k answers

The index structure is used to compute the answers. The leaf node inverted list contains the content nodes and quasi contend nodes, scores of the keyword. For computing top k results heap based method [6] is used which uses the partial virtual inverted lists which contain the higher score nodes so to avoid the union of lists which is expensive.

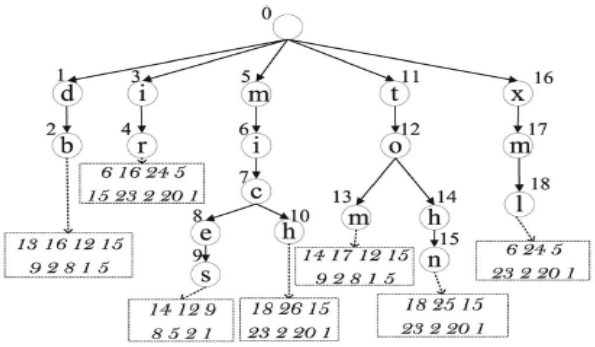

Fig 3 extended tier structure

Procedure

1. Sort the scores in the inverted lists
2. If the inverted list is long the partial virtual inverted list
3. Construct max heap, such that each node contain <node, score>
4. The top element of max heap is highest score node and is deleted, max heap is adjusted
5. Deleted node with score<=T (threshold) are taken into result set and return the result set if the top –k answers are retrieved

For example assume user composed the query "db". The inverted list of db contains the nodes 13,16,12,15,9, 2,8,1,5.The scores of these nodes computed by two ranking functions are 1.52,1. 52,1.21, 1.21,0. 9728,0, 495,0.77, 0,396,0.6225 respectively. These scores have to be sorted and max heap is constructed and a threshold is fixed be 10 so the top elements< (13, 1.52>, <16, 1.52>, <12, 1.21>, <15, 1.21> the top e results are retrieved. This technique is more efficient and effective.

IV. CONCLUSION

This paper presents the keyword search over the xml data which is user-friendly and there is no need for the user to study about the xml data .This paradigm gives the relevant results the user wants. Fuzzy search over xml data is studied which gives approximate results. Various methods for querying on xml data LCA based method, ELCA, heap based method are presented and of all these methods heap based method gives high quality results. To further improve the search performance, forward indexes can be used which eliminates the need to visit inverted list and gives the corresponding score of element.

V.REFERENCES

1. 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. 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.
3. 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.
4. 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.
5. 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.
6. Jianhua Feng, and Guoliang Li" Efficient Fuzzy Type-Ahead Searchin XML Data" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 5, MAY 2012

AUTHORS PROFILE

Ch. Lavanya Susanna, completed btech in Information Technology from RVRJC College of Engineering, Guntur, Andhra Pradesh, in 2012. pursuing Mtech in department of Computer Science and Engineering from K LUniversity, vaddeshwaram Andhra Pradesh. Her research interest mainly include data mining and image processing