RESEARCH PAPER | Engineering | Volume : 4 | Issue : 3 | Mar 2014 | ISSN - 2249-555X

Survey on Interactive Keyword Search Over XML Data to Obtain Top-K Results

KEYWORDS: XML, keyword search, type-ahead search, fuzzy search

Supriya C. Rathod
Student ME (CSE), Computer Science and Engineering, SYCET, Aurangabad, Maharashtra, India

Sonali M. Tidke
Head, Computer Science and Engineering, SYCET, Aurangabad, Maharashtra, India

ABSTRACT *In conventional keyword-search system on XML data, a user composes a query keyword, submits it to the system, and retrieves relevant information. In the case if the user doesn't know how to issue queries, he tries multiple queries and sees multiple times what the result is. In this paper, we study new keyword search system in which the system searches XML data as the user types in query keywords. It allows users to find data as they type, even if there is a presence of minor errors in query keywords. The proposed method consists of the following features: 1) It extends Auto complete which supports multiple keywords in XML data. 2) It can find high-quality answers that have keywords matching query keywords approximately. 3) Our effective index structures and searching algorithms can achieve a very high interactive speed. We propose effective index structures and top-k algorithms to achieve a high interactive speed. We propose effective ranking functions and early termination techniques to progressively identify the top-k relevant answers to achieve high search efficiency and result quality. This paper focuses on the survey of techniques used to retrieve the top k relevant results from the xml document more efficiently.*

I. INTRODUCTION

A keyword search looks for words anywhere in the record. It is emerged as most effective paradigm for finding 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 keyword search 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 is Autocomplete which predicts the words the user had typed in. More and more websites support these features example Google, yahoo. One limitation of autocomplete is it treats multiple key words as single key word and don't allow them to appear in different places. To solve this problem other method is proposed complete search in textual documents which allows multiple keywords to appear in different places but it doesn't 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 like delete, insert, substitute required to make the two words equal.

II. BACKGROUND

This section provides an overview on XML data models, query models and the definitions of query results. The XML documents represent hierarchically structured information and are generally modeled as Ordered Labeled Trees. XML document can be modeled as a tree. Each element, attribute and text value in the XML document is a node in the XML tree. Nodes represent XML elements and are labeled with corresponding element tag names, organized following their order of appearance in the document. Each edge in the XML tree represents the membership of the element corresponding to the child node, under the element corresponding to the parent node in the XML document. Graph model models an XML document as a graph. An example of XML document is shown in figure 1.

```
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Indian</title>
<author>G. D. Laurentis</author>
<year>2008</year>
<price>40.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J  Rowling</author>
<year>2008</year>
<price>25.00</price>
</book>
<book category="WEB">
<title lang="en">Learning XML & HTML</title>
<author>E. Ray</author>
<year>2009</year>
```

Figure 1 Example of an XML document

Volume : 4 | Issue : 3 | Mar 2014 | ISSN - 2249-555X

III. RELATED WORK

XRANK [1] allows for a graceful transition from HTML documents to XML documents (such as in the World Wide Web and Corporate Intranets) because it can handle both classes of documents using the same framework. In this paper used ElemRank algorithm, that ranks XML elements not XML documents, Newer Inverted List Structures to efficiently store keyword-element mappings for XML elements, Newer query evaluation algorithms to return top-m results X-Rank handle specifics of XML keyword search, such as removing spurious results and inferring position lists.X-Rank handles multi-way merges, corresponding to multiple keywords. It determines deepest common ancestors. The element ranking used in XRANK can be incorporated in XSEarch as well, but its utility is not clear. It returns semantic related fragments.

Wei Waet al in [3] presented a multidimensional search approach that allows users to perform fuzzy searches for structure and metadata conditions in addition to keyword conditions. Their techniques individually score each dimension and integrate the three dimension scores into a meaningful unified score. They also have designed indexes and algorithms to efficiently identify the most relevant files that match multidimensional queries. Experimental evaluation of their approach showed that their scoring framework for fuzzy query conditions in non-content dimensions can significantly improve ranking accuracy.

Chunxiao Liuetalin [4] presented a user-friendly Top-k keywords searching approach based on the relationship of keywords. The SLCA of a keyword search was first obtained by the LISA II algorithm. Then, the structure of SLCA was leveraged to speculate the relationship of keywords. Next, the relationship of keywords could be estimated by the structure of twig queries and these twig queries were ranked according to the relationships of keywords. Then all results of the ordered twig queries were obtained by using TJFast algorithm.

Yiqun Chen and Jinyin Cao in [5] have presented an approach to type-ahead keyword searched in XML data, call XIR. The IR-style approach basically utilized the statistics of underlying XML data to address challenges that identify the user search intention, i.e. identify the keywords to express user interests and identify nodes user wanted to search for and search via. and resolve keyword ambiguity problems: synonyms and polysemy exist in natural language, and a keyword could appear as the text values or tag value of different XML node and carry different meanings. They have modeled XML data as a graph, analyzed the identification of user search intention and result ranking in the presence of keyword ambiguities and used the related definition and formula to build a query prediction technique to improved search efficiency.

Zhifeng Baoetalin [6] have studied the problem of effective XML keyword search which included the identification of user search intention and result ranking in the presence of keyword ambiguities. They utilized statistics to infer user search intention and rank the query results. In particular, they have defined XML TF and DF, based on which have been designed formulae to computed the confidence level of each candidate node type to be a search for/search via node, and further proposed XML TF*IDF similarity ranking scheme to captured the hierarchical structure of XML data. Finally, the popularity of a query result was considered to handle the case that multiple results have comparable relevance scores.

Jiang Li and Junhu Wang [7] have presented an XML keyword search provided a simple and user-friendly way of retrieve data from XML databases.XReal utilized the statistics of underlying data to resolved keyword ambiguity problems. However, they found their presented formula for inferring the search-for node type suffers from inconsistency and abnormality problems. Finally a dynamic reduction factor schemes as well as an algorithm Dynamic Infer to resolve these two problems. Experimental results are shown provided to verify the effectiveness.

Liang Jeff Chen and Y. Papakonstantinouin [8] have presented a series of algorithm that incorporated both the efficient semantic pruning and the top-K processing to support top-K keyword search. They presented a join-based algorithm that processes nodes bottom up and reduced keyword query evaluated into relational joins. Several optimizations were presented to further improve its efficiency. They incorporated the idea of the top-K join from relational databases and presented a join-based top-K algorithm to computed top K results. Extensive experimental results confirmed the advantages of algorithms over previous algorithms in both efficiency and top-K processing.

XSEarch [15] employs the semantics of meaningful relation between XML nodes to answer query keyword. It incorporates a straightforward command language, appropriate for a naive user. It can easily accommodate different types of relationships between nodes. Extended information retrieval techniques are used to rank query answers. For efficient implementation, advanced indexing techniques are developed. It includes full-text search features and ranking to XQuery. In XSEarch [15], proximity is included in terms of the size of relationship tree in the ranking formula and it is not affected by the order of children. XSEarch focuses on the semantics and the ranking of the results.

Interactive search can also save user typing efforts[9].In this a new algorithm called "interactive, fuzzy search" which has two factors one in Interactive and the other one is fuzzy. To make search extremely interactive, for every keystroke on the shopper browser, from the time the user presses the key to the time the results computed from the server are displayed on the browser, the delay ought to be as tiny as attainable. Fuzzy search is used to find records with keywords that match question keywords some. On checking out relevant records, the system conjointly tries to seek out those records that embody words almost like the keywords within the query.

IV. DIFFERENT METHODS FOR KEYWORD SEARCH OVER XML DATA

A. LCA based method

The lowest common ancestor (LCA) is a concept in graph theory and computer science. There are different ways to answer the query on xml document; one commonly used method is LCA based method [12]. Many algorithms that use query over xml uses this method. Content nodes are the parent node of the keyword. For keyword query the LCA based method retrieves content nodes in xml that are in inverted lists and Identify the LCAs of content nodes in inverted list Takes the sub tree rooted at LCAs as answer to the query .There are some limitations of LCA that it gives irrelevant answers and results are not of high quality.

B. ELCA based method

To address the limitation of LCA based method exclusive LCA (ELCA)[12] is proposed. It states that an LCA is ELCA if it is still an LCA after excluding its LCA descendants. Xu and Papakonstantinouin [14] proposed a binary-search-based method to efficiently identify ELCAs.

C. MLCA based method

MLCA [10] concept of MLCA was proposed with schema-free XQuery which allow users to mix keyword search and structured query as it is beneficial to find relevant matches.. Li et al. [10] uses a stack based algorithm to compute MLCA nodes. First it retrieves list of all the matches to each keyword. Then it visits all the keyword matches in the document order and maintains a stack in which each node is a descendent of the node below it. If node contains all the keywords in its sub tree, it is identified as a potential MLCA. To determine the pattern matches, it examines meaningfully relatedness of keyword matches.

Volume : 4 | Issue : 3 | Mar 2014 | ISSN - 2249-555X

D. SLCA based method

*I*n SLCA [11], there are two different algorithms proposed for keyword search in XML documents according to the SLCA semantics are Indexed Lookup Eager and Scan Eager. These algorithms work quickly and produce parts of answers so that, user may not wait for long to see first few answers. The Indexed Lookup Eager algorithm is important in practice since the frequencies of keywords typically vary significantly. The Scan Eager is tuned for the case where the occurrences of the query's keywords do not vary significantly.

V. RESULT RANKING

There are various ranking strategies has been proposed for keyword search on XML data. The main types of ranking factor are TF IDF and vector space model.

A. Term frequency and inverse document frequency

The term frequency–inverse document frequency is a weight used in information retrieval, it is a statistical measure used to evaluate how important a word is to a document .The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the collection. It is used in many search engines for ranking a document's relevance for a user query.

The term frequency of term in document is computed as shown in (1).

$$\mathrm{tf}_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}}(1)$$

The inverse document frequency [13] of term is computed as shown in (2).

$$\mathrm{idf}(t)=\log\frac{|D|}{|\{d:t\in d\}|}(2)$$

B. Vector space model ranking factors

Vector space model[13] is an algebraic model for representing text documents. Word in document is the important component of vector. The document vector and the query vector are used to measure the relevance of a document with respect to a query. XSEarch[15] is implemented same as vector model, it represents each node in the result and each term in the query as a vector, and use vector similarity to determine the relevance of the result. The vector space model has the advantages over the Standard Boolean model are Simple model based on linear algebra, Term weights not binary, allows computing a continuous degree of similarity between queries and documents, allows ranking documents according to their possible relevance and allows partial matching.

VI. CONCLUSION

This paper focuses on the survey of different approaches used for keyword search to obtain the top k results from the xml document more efficiently which is LCA based method, ELCA, SLCA, MLCA heap based methods. All these heap based method gives high quality results. In ranking, top-k result computation methods for semantics and ranking functions are still unknown. To find the structured relationship between keywords, users feedbacks is used in existing work. We propose effective indexing methods and efficient algorithms to progressively and efficiently identify the top-k answers.

REFERENCE [1] L. Guo, F. Shao, C. Botev, and J. hanmugasundaram, "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] Wei Wang, Christopher Peery, Ame´lie Marian, and Thu D. Nguyen, "Efficient Multidimensional Fuzzy Search for Personal Information Management Systems",IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.24, NO. 9, SEPTEMBER 2012. | [4] Chunxiao Liu, XiangfuMeng and Ke Wei, "A Top-k Keywords Searching Approach based on the Relationship of Keywords", IEEE International Conference on Systems,Man, and Cybernetics, October 2012. | [5] Yiqun Chen and Jinyin Cao, "TakeXIR: a Type-Ahead Keyword Search XML Information Retrieval System", I.J. Education and Management Engineering, vol.8,pp: 1-5, 2012. | [6] ZhifengBao, Jiaheng Lu, Tok Wang Ling and Bo Chen, "Towards an Effective XML Keyword Search", Knowledge and Data Engineering, Vol. 22, no. 8, pp: 1077- 1092,2010. | [7] Jiang Li and Junhu Wang, "Effectively Inferring the Search-for Node Type in XML Keyword Search", Database Systems for Advanced Applications, p p.110-124, 2010. | [8] Liang Jeff Chen and YannisPapakonstantinou, "Supporting Top-K Keyword Search in XML Databases", Data Mining Workshops (ICDMW), p p. 805- 812, 2012. | [9] 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. | [10]. Schmidt, A., Kersten, M.L., Windhouwer, M.: Querying XML documents made easy: nearestconcept queries. In: ICDE, pp. 321–329 (2001). | [11]. Xu, Y., Papakonstantinou, Y.: Efficient keyword search for smallest LCAs in XML databases.In: SIGMOD Conference, pp. 537–538 (2005). | [12] 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. | | [13]. Ziyang Liu • Yi Chen, "Processing keyword search on XML: a survey", Springer Science+Business Media, LLC 2011. | | [14]. Chen, L.J., Papakonstantinou, Y.: Supporting top-K keyword search in XML databases. In:ICDE, pp. 689–700 (2010). | [15]. Cohen, S., Mamou, J., Kanza, Y., Sagiv, Y.: XSEarch: a semantic search engine for XML.In: VLDB, pp. 45–56 (2003). |