Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Create XML Document and Efficient Interactive Keyword Search Technique over XML Data

Harshal R Aher
Department of Computer Engineering,
Dr.D.Y.Patil Collage of Engineering, Ambi, Pune, India

Anupkumar Bongale
Department of Computer Engineering,
Dr.D.Y.Patil Collage of Engineering, Ambi, Pune,india

Abstract—: in this paper we create XML document to store data, in XML formats for Security purpose. We consider the problem of efficiently producing ranked result for keyword search queries over XML document. We study query process, keyword process. In traditional method there are Xlink, Xpath and Xquery are query method to submit data in XML file of XML database. In this method new user can't understand syntax of query when issuing the query, in this process first take query, submit to the system and retrieve relevant answer. In keyword search there are fuzzy type ahead search in XML data that user type a keyword search on fly and access a new information paradigm, this method are alternative to traditional method the user no need to know knowledge of XML query language and syntax. We also present a user study confirming the keyword-based search over SQL for a range of database retrieval tasks. At query time, the text index supports keyword-based searches with interactive response. Effective keyword search valuable top-k over XML document, these are user easily handle, semantic and navigate into document. Effective XML keyword search with relevance ranking is an approach that contains ambiguities, because a keyword can appear in a name tag or a text value of XML node. Top-k queries on large multi-attribute data sets are fundamental operation in information retrieval and ranking application. We used top-k it can be identify approximate answer in best ranking system in XML document more effectively and efficiently.

Keywords— Fuzzy search, Keyword Search, LCA and MCT, Type-ahead search, XML.

I. Introduction

In traditional keyword-search system over XML data, a user composes a keyword query; submit it to system and retrieves information. Actually particular person know about language what is Xpath and Xquery, what are their syntax, notation etc because without syntax, no one can retrieve data, Xquery and this paper, we study effective search in XML data, system search XML data on the user type in query keywords. It allow user to explore data as they type, even in presence of minor error of their keyword. We propose effective index structures and top-k algorithm to achieve a high interactive speed. We examine effective ranking function and early termination techniques to progressively identify the top-k relevant answer [1], [2].

Nowadays most of the transactions on the internet XML are used for storing and retrieving purpose. Most of the leading product developed companies use XML metadata framework. This paper started with a goal to manage XML data. It helps in storing, relevant answer. In this case user has limited knowledge about the data, often the user feels left in dark when issuing queries, and has to use a try and see approach for finding, managing, publishing, retrieving data from database in XML format and updating storing data in XML Document. There are different modules of this paper. One of the modules is a SQL manager, which helped to retrieve and manage data from XML data in XML database and we implement keyword search XML data in XML database, user management and security are another modules. Database server is Client-Server based database. It is more user-friendly, easy to retrieve and easy to access the database for both programmer and the client. It is used to create database, table, query, the report [3].

II. Literature Survey

XML stands for Extensible Markup language. The word "Extensible" implies that a developer can extend his ability to describe a document, and define meaningful tags for his application XML is used to generate dynamic content. Databases are study of SQL-SERVER, ORACLE, My SQL , XML are done in the aspect of manipulating the stored data by their respective query language. XML database helps professionals and the corporate to record and maintain the data into the database. For using the above specified database corporate has to pay respected amount as per the company rules and regulations for getting the registration from the authorized database companies. Installation cost, maintenance cost and the implementation cost can affect the company's production cost. The XML database is a platform independent server database and can be used with free of cost provided by the Sun Microsystems [11].

In XML There are two types Xpath and Xquery. Xpath is declarative language for XML that provide a simple syntax for addressing part of on Xml document. Xpath collection of element can be retrieved by specifying a directory like path with zero or more condition place on the path. Xpath treat an a XML document as a logical tree with nodes for each element, attribute text, processing instruction, comment, namespace and root [17],[1]. The basic of the addressing mechanism is the context node (*start node*) and location path which describe a path from one point in an XML document

© 2014, IJARCSSE All Rights Reserved

to another. Xpointer can be used specify on absolute location or relative location. Location of path is composed of a series of step joined with "/" each move down the preceding step. Xquery is incorporate feature from query language for relational system (*SQL*) and Object oriented system (*OQL*)[11]. Xquery support operation on document order and can negative, extract and restructure document. W3c query working group has proposed a query language for XML called Xquery. Values always express a sequence node can be a document, element, attribute, text, namespace. Top level path express are ordered according to their position in the original hierarchy, top-down, left-right order [14]. The important parts are Data-Centric document and Document-Centric document. Data-centric document Xpath are complex for understand. It can originate both in the database and outside the database. These documents are used for communicating data between companies. These are primarily processing by machine; they have fairly regular structure, fine-gained data and no mix content. Document- Centric are document usually designed for human consumption, they are usually composed directly in XML or some other format(*RTF*, *PDF*, *SGML*) which is then converted to XML. Document-Centric need not have regular structure, larger gained data and lots of mixed content [13], [3]. In this paper to analysis of previous technology that they are working LCA (lowest common ancestor) [10], ELCA (Exclusive Lowest common Ancestor) [10], MCT (minimum cost tree)[14] and introduce new technology Top-k algorithm[16], [1] identify approximate best ranking answer in system in XML document more effectively and efficiently.

Pros:

1. Keyword search provides user friendly interface rather than Xpath and Xquery.
2. XML is used to store data in XML document format rather than table format.
3. XML provide security to data, user not easily recognize the XML data rather than traditional Table format.
4. User enter a keyword (i.e Attribute, key, identifier), but traditional method require syntax of Xpath and Xquery.
5. Top-k provides most approximate answer than MCT and LCT.
6. Administrator has assign authority to each user for to do work effectively and efficiently.
7. It is client-server module that's why it is very easy to handle relations between users and database.
8. When each user enter into his own session, data store into temp folder as per his database document name .

Cons:

1. To required proper keyword for correct answer otherwise it not provide good answer.
2. It required most time to iteration, to retrieve the answer.
3. User knows SQL Query language, when inserting data and creating record.
4. Limited users have permission to changes in database.

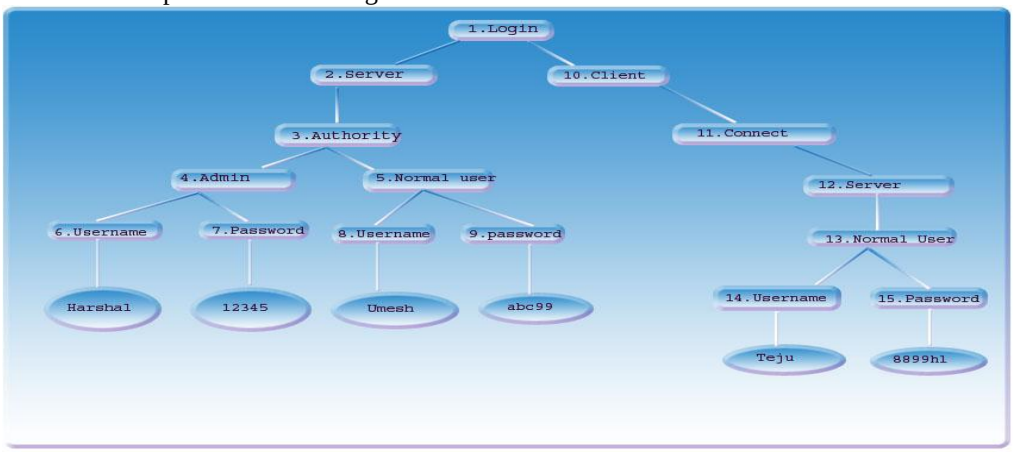

Fig 2.1 XML Document

III. Xml Query Techniques Based Fuzzy Methods

Database server is a client-server based database. It is more user-friendly, easy to retrieve and easy to access the database for both the programmer and the client or end user. It is used to create database, table, query and the reports. User can view the database, create table and analyze the query and a after all he can make report on the basis of tables and with respect to their queries.

For creating, accessing and maintaining the database. User should have permission from the server. Server granted the permission and after that client (user) can do what he wants to do. Client can view only the encrypted from of data a because of all the data are maintained in the XML database in decrypted from what a client can never perceive it. For the security point of view it has particular user with their passwords who are the authorized persons who can access the database. This is query analyzer database to which multiple users can access the database at the same time with no restrictions. It is a platform independent database and more economical than any other database. We propose the index to improve search performance. We can utilize "random access" based on the index to do an early termination in the algorithms. That is, given an XML element and an input keyword, we can get the corresponding score of the keyword and the element using the index, without accessing invert lists. Fagin et algorithm have proved that the threshold-based algorithm using random access is optimal over all algorithm that correctly find the top-k answer.

Notice that it is very expensive to construct the union lists of every input keyword as there may be multiple predicted words and many inverted lists. Instead, we can generate a partial virtual list on the fly. We only use the element in the

© 2014, IJARCSSE All Rights Reserved

partial in the partial virtual list to compute the top-k answers. The partial virtual list can avoid accessing all the element of inverted lists of predicted words. It only needs to access those with higher scores, and if we have computed the top-k answer using the partial accessed element, we can do an early termination and do not need to visit other element on the inverted lists.

In this system large number of security options provide to data and user. Administrator has most of responsibility to create user, allow permission maintain database. When user enter username and password into the system to login, he perform work on data to store data into document, retrieve data from document present in database, simultaneously he goes to temp folder where document is store only on his data see into temp folder not other person data because when user log out his file delete from temp folder maintain database security of each user. Another one is data traveling form one user to document database or other person, data must be encrypted and maintain reliability into network.

3.1 Database Design

XML server consists of various modules. GUI Client establishes the connection with server using respective username and password. SQL manager to establish the connection with respective database, parse the query in proper syntax. (It contains "syntax.xml" which is used for syntax checking), Display result Grid or in File format. All syntax's for DDL as well as DML queries are stored in "syntax.xml" file. Example *syntax:*

Data Storage with the popularity of XML of the server need to work with and store XML data. Example .xml files format.

```
<article>
        <author>Gerhard</author>
        <title>The web in 10 years</title>
</article>
```

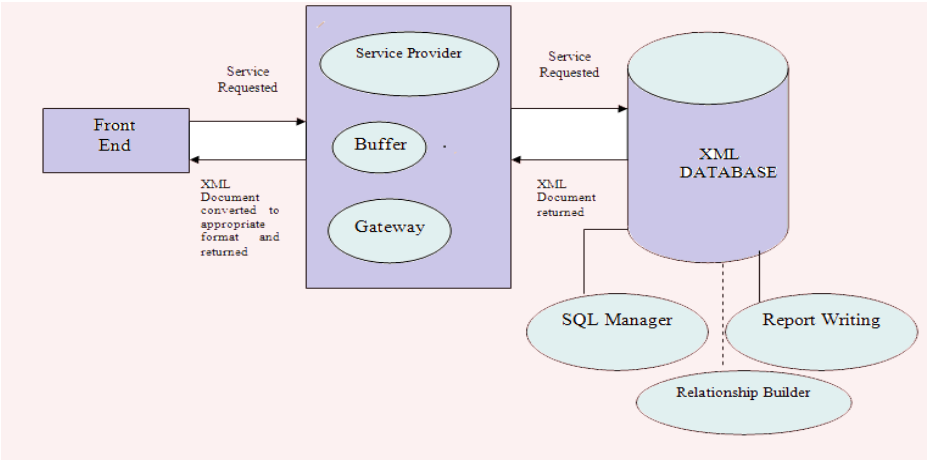

Fig 3.1 XML server

Relationship builder of XML server builds relation between two XML documents, implementing the concept of primary keys. User management helps in managing users by Creating users, editing users, Deleting users, Assigning/Editing their passwords. Server will provide its own authentication features to authenticate users via facility of XML Encryption/Decryption, which ensure that, unauthorized individuals or intruders cannot access important document.

In this section three important XML query and keyword search methodologies are explain. Major problem associated to Xpath and Xquery are their complexity involved in the syntax for query. Compared to Xpath and Xquery, LCA-based interaction search [7] and minimum cost tree [14] are better and efficient. Following subsection give detailed information on the above said methods.

3.1.1 Minimum cost tree

To find relevant answer, to a keyword query over an XML document. For each node, we define its corresponding answer to the query as its sub tree with paths to nodes that include the query keyword. This sub tree called the "minimal cost tree" for this node. Different node corresponding to different answer to the query, and we will study how to quantify the relevance of each answer to the query for ranking. Given an XML document D, a node n in D, and a keyword query Q={k1,k2,k3,…,kl}, a minimal cost tree of query Q and node n is the sub tree rooted at n, and for each keyword ki € Q, if node n is a qussi-content node of ki, the sub tree include the pivotal path for ki and node n. we first identify the predicated word for each input keyword. Then, we construct the minimal cost tree for every node in the XML tree based on the predicated word, and return the best ones with the highest score. The main advantage of that, even if a node does not have descendent nodes that include all the keyword in the query, this node could still be considered as a potential answer [4].

3.1.2 LCA-Based interactive search

We propose a lowest common ancestor (LCA) based interactive search method. We use the semantics of exclusive LCA to identify relevant answer for predicated words. We use trie to index the tokenized words in XML data. First for a single keyword, find corresponding tree node. Then we locate the leaf descendents of this node, and retrieve. The corresponding predicated words and the predicted word and the predicated XML element on their inverted lists. For a query string into keyword k1, k2, k3……, kl. For each keyword ki (1< i< l), there are multiple predicated word [5].

Procedure

© 2014, IJARCSSE All Rights Reserved

- For keyword query the LCA based method retrieve 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 answer to the query for example suppose the user type the query "www db" then the content nodes of db are{13,16} and for www are 3, the LCAs of these content nodes are nodes.

Limitation

- It gives irrelevant answer
- The result are not of high quality

3.1.3 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 nodes 2, 12, 15, 1 here the ELCAs are 12,15. The sub tree rooted with these nodes is displayed which are relevant answer Node2 is not an ELCA as it is not an LCA after excluding nodes 12 and 15. XU and papakonstantinou [9] proposed a binary-search based method to efficiently identify ELCAs.

3.2 Efficient and effective top-k algorithm for XML data search

In this paper we first check it out that how top-k search algorithm are come. Whenever ranking the answer of keyword it used LCA and MCT with their particular score [7],[14]. Our parameterized top-k algorithm proceeds in two stages. First one is a structure algorithm that on a problem that on a problem that on a problem instance construct a structure of feasible size, and the second stage is an enumerating algorithm that produces the k best solutions to the instance based on the structure. We develop new techniques that support efficient enumerating algorithm. We investing the relation between fixed-parameter tractability and parameterized top-k algorithm [16],[1].

3.2.1 Ranking query answer

Now we discuss how to rank the MCT for a node n as answer to the query. Intuitively, we first evaluate the relevance between node n and each input keyword, and then combine these relevance score as the overall score of the MCT. We will focus on different method to quantity the relevance of node n to a query keyword, and combine relevance score [4], [5], [16].

a. Ranking the sub tree

There are two ranking function to compute rank/score between node n and keyword k_i.

Case 1: n contain keyword k_i.

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

$$\text{SCORE1 (n},k_i) = \frac{\ln\left(1+tf(k_i,n)\right)*\ln\left(idf(k_i)\right)}{(1-s)+s*ntl(n)} \quad \ldots\ldots\ldots\ldots(1)$$

Where, tf (k_i, n) – no: of occurrence of ki in sub tree rooted n

idf (k_i)- ratio of no: of node in XML to no: of nodes that contain keyword k_j

ntl (n)- length of $\frac{n}{nmax}$= node with max terms

s- Constant set to 0.2

Assume user composed a query containing keyword"db"

$$\text{SCORE (13, db)} = \frac{ln(1+1)ln\left(\frac{27}{2}\right)}{(1-0.2)+(0.2*1)} = 1.5$$

Case 2: node n does not contain keyword ki but its descendent has k_i. Ranking based on ancestor- descendent relationship.

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

$$SCORE2(n,k_j) = \sum_{p\in P}^{n} \propto^{(n,p)} * \text{SCORE1(p,k}_j) \ldots\ldots\ldots(2)$$

Where p- set of pivotal nodes

α – constant set to 0.8

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

b. Ranking Fuzzy search

Given a keyword query Q= {k_1, k_2,…., k_j} in term of fuzzy search, a minimal-cost tree may not contain predicated words for each keyword, but contain predicted words for each keyword. Let predicated word is {w_1, w_2,…., w_l} the best similar prefix of wi could be considered to be most similar to k_i. The function to quantify the similarity between ki and wi is

$$\text{Sim}(k_i,w_i) = \gamma * \frac{1}{1+ed(k_i,a_i)} + (1-\gamma) * \frac{|a_i|}{|w_i|} \ldots\ldots\ldots(3)$$

Where ed- edit distance

a_i –prefix

w_i – predicted word

© 2014, IJARCSSE All Rights Reserved

γ -constant

Where γ is turning parameter between 0 and 1, as the former is more important, γ is close to 1. Our experiment suggested that a good value for γ is 0.95. We extend the ranking function by incorporating this similarity function to support fuzzy search as below

SCORE (n, Q) = $\sum_{i=1}^{l} sim(k_i, w_i) * SCORE1(n, w_i)$

SERVER

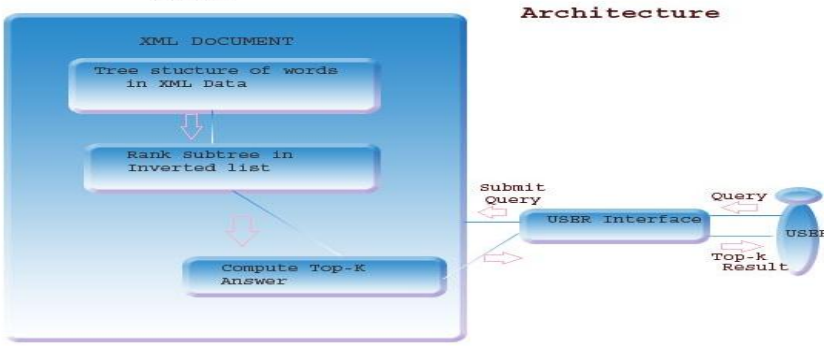

Fig 3.2 Architecture of top-k

IV. Result set and Data set

Format	Input	Output
Query	If we create table using "create" query, which contain data	.XML file created successfully. <article> <author>Gerhard</author> <title>The web in 10 years</title> </article>
	If he/she select database and fire the query as per his/her requirement 1) select Gerhard from testing ; 2) drop table head ;	1. List of required record is display and format is XML <article> <author>Gerhard</author> <title>The web in 10 years</title> </article> 2. List of record not display if record not present in database.
Keyword search	Appropriate keyword, id (primary key), Attribute	List of row or column display as per keyword.

Fig.4.1. Data Set and Result set.

In this paper most of things clear about the keyword search algorithm and searching time of data from the .xml file. We observe that the MCT-based method and achieves much higher search performance in terms of both exact search and fuzzy search.

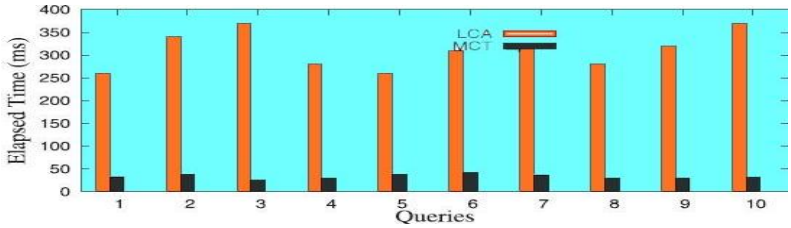

Fig 4.2.Keyword query time

In below given graph indicate that the we develop novel ranking techniques and efficient search algorithms. In our approach

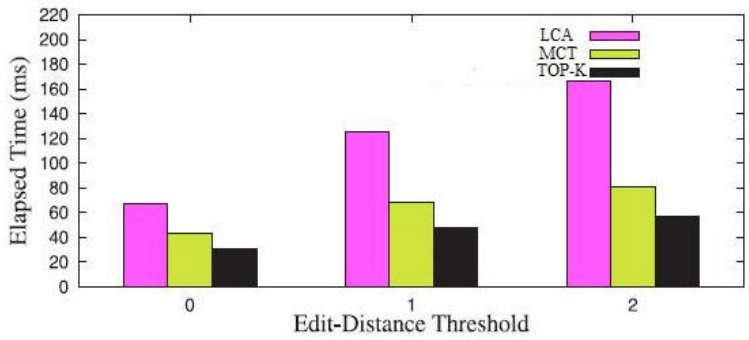

Fig 4.3.Keyword Search Time

© 2014, IJARCSSE All Rights Reserved

Each node on the xml tree could be potentially relevant to a keyword query. For each leaf node in tree, we index not only the content nodes for the keyword of the leaf node, but also those quasi-content nodes whose descendents contain keyword. We locate the leaf descendant of this node and rerieve the corresponding predicted words and the predicted XML element on the inverted lists. This is attributed to our effective index structures and threshold-based computing algorithm.

V. 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 result the user want fuzzy search over XML data is studied which gives approximate result. We studied the problem of fuzzy typ-ahead search in XML data. We proposed effective index structure efficiently identify the top-k answer. We examine the LCA-based method to interactively identify the predicated answer. We have developed a minimal-cost-tree based search method to efficiently and progressively identify the most relevant answer. We have implemented our method achieves high search efficiency and result quality.

Acknowledgement

I would like to express my gratitude to all those who gave me the possibility to complete this project. I want to thanks the Department of Computer Engineering for giving me permission to do the necessary work and to use department data. I deeply indebted to my project guide prof. Anupkumar Bongale whose help, stimulating suggestion and encouragement helped me in the all time. I have furthermore to thank to all staff member, ME, PG Coordinator, who gave and confirmed this permission and encouraged me to go ahead with my project.

REFERENCES

[1] J.Feng and Guoliang Li "Efficiently Fuzzy type-ahead searching XML data" IEEE transction on Knowledge and Data Engineering Vol.14,May 2012

[2] CH.Lavanya "Interactive search over XML Data to obtain Top-k result" International journal of Soft Computing and Engineering, ISSN: 2231-2307, Volume-3, Issue July 2013

[3] S.Agrawal, S. Chaudhri and G.Das "DBXplore: Asystem for Keyword Based Search over relational Database", proc. Int'l Conf. Data Eng(ICDE), pp.5-16-2002

[4] Z. Bao, T.W.Chen and J. Lu," Effective XML Keyword search with relevance oriented Ranking", proc Int'l conf Data Eng(ICDE)2009

[5] H. Bast and I.Weber,"Type less, find more:Fast Auto Completion search with a index", Proc. Ann Int'l ACM conf Research and Development in information Retrieval(SIGIR) 2006

[6] L.Li, H. wang, J. LI, H.Gao" Efficient algorithum for skyline top-k keyword queries on XML streams" Harbin Institude of Technology.

[7] Y.Xu and Y.Papakonstantiou, "Effiient keyword search for smallest LCA in XML data" proc Int's conf Extending Database Technology Advance in Database technology(EDBT) 2008

[8] G. Li, S.Ji,C.Li and J.Feng,"Efficient type-ahead search on Relational Data: A Tastier Approch" proc ACM SIGMOD Int't conf Management of data,2009

[9] S.Ji, G. Li, C. Li and J.Feng, "Efficient Interactive Fuzzy Keyword Search", Proc Int'l conf World Wide Web ,2009

[10] Yu. XU Teradat, Yannis Papakonstantion university of California", Efficient LCAbased keyword search in XML Data" ACM Copyright, 2003

[11] Andrew Eisenberg IBM,"Advancement in SQL/XML" Jim Meton oracle corp, 2002

[12] Ronald Bourret," XML and Database", Independent consultant, Felton, A 18 Woodwardia Ave. Felton CA 95018 USA SPRING 2005

[13] G.Li, Jian Hua Feng, Lizhu Zhou,"Interactive search in XML Data" Deparment of Computer Science and Technology, Tshinghua National Laboratory for Information Science and Technology, Tsinghua university, Beijing 100084,China

[14] Bolin Ding, Jeffrey Xu Yu, Shan Wang, Lu Qin, Xiao Zhang Xuemin Lin" Finding top-k Min-cost –connected Tree in Database", The Chinese university of Hong Kong China

[15] L.Chen, Lyad A kanj, Jie Meng, Ge Xia, Fenghui Zhange ," Parameterized top-k algorithm", communicated by D-Z DU, 2012

[16] Dolling Li, Chen Li, J. Feng, Lizhu Zhou, "SAIL: Structure-aware indexing for effective and progressive top-k keyword search over XML document", Deparment of Computer Science, university of California, Irvine, CA 92697-3435,USA

[17] H.Willimson,"The complete Reference of XML", The McGrew-Hill Companies, Inc, New York 2009

© 2014, IJARCSSE All Rights Reserved