International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072

Achieving effective keyword ranked search by using TF-IDF and cosine similarity

M.PRADEEPA[1], V.MOHANRAJ[2]
[1]PG scholar department of IT, Sona College of Technology,Tamilnadu,India.
pradeepa92.murugan@gmail.com
[2] Professor department of IT, Sona College of Technology,Tamilnadu,India.

vmohanraj06@gmail.com

ABSTRACT -Recent advancement in day to day life accumulates more data in database hence database grew larger and complex since the number of entities is more and searching through the database is also becoming complex. The users are interested in gathering most relevant information by querying the database initially, by using Structured Query Language (SQL), but this can be done only by the SQL experts. So keyword search is becoming emerging field in the research area whereon prior knowledge is not needed about the schema and also query language and also reduces the search time. Producing top answers that are highly relevant to the user's query is a great challenge. In this paper various related works of mining methodology using keyword are described. The survey also discuss about the preprocessing stage, query generation stage, recommendation stage for each techniques. Further, the pros and cons of each method is discussed in detailed manner. In this paper, a novel indexing and TF-IDF framework are used in preprocessing and recommendation stage respectively to get the answers related to the users query.

***Index Terms*-**database, SQL, keyword search,Top-k Query Processing

I. INTRODUCTION

Data mining[6] is a process of searching through large amount of data in order to find useful pattern. Once these useful patterns are found they can be further used to make certain decisions for development of their business. Since database is complex and larger, there are many mining techniques to retrieve the data. Different kinds of methods and techniques are needed to find different kinds of pattern.

Database is a collection of data. Some Examples of databases are airline reservation system, registration records etc. The data that are stored in the database does not have any meaningful information until it is retrieved. Information means that the data is structured and communicated in the meaningful
manner. Therefore the data that are relevant to the users query has to be retrieved and converted into meaningful information, finally that information is evaluated and organized as knowledge. Retrieval of relevant data from the database quickly and effectively is a challenging task.

Retrieval of data from the relational database is performed using SQL. For this one need
to have knowledge about the SQL. But all users will not be an SQL expert, so by considering the naïve users also to query the database[10], keyword search[1],[7] concept has emerged. Most of the users uses the keyword search[16] to get the answers related to their query in order to reduce their search time

The answers that are obtained from the keyword search[12],[13] includes the tuples that have the keyword appearing in the query. To make the searching easy various approaches that relay on building a graph or maintaining an indices priori for the database. It is not an easy task to produce the most relevant answers quickly and effectively without doing some processing work. So the big challenge is to provide top answers[9],[21] for the keyword query with minimum processing content. Current studies focus on different stages of keyword search: preprocessing, query generation and recommendation. At the preprocessing stage, the data in database is modelled as either graph or index. At the query generation stage, all the tuples that matches the users query are retrieved from the database. Among those the top answers that are relevant to the user interest is listed at the recommendation stage by means of some measures.

1.1.FUNDAMENTAL CONCEPTS

In this section some basic notions of the keyword search is described.

A.Relational Database:
The relational model[17] is widely used data model where data are organized into one or more tables of rows and columns with a unique key identifying each row and majority of the database uses this model. Relational Database Management System (RDBMS) provides an abstract view of the underlying data for the users and the searching is done mostly on the top of the RDBMS.

B.Keyword search:

The keyword search[14] tells that when the user gives the query which contains the finite amount of keyword, the resultant tuples are produced by searching through the database which contains that particular keyword.

C.Data graph:

Usually graph[15] is represented in nodes and edges. Here the nodes are the tuples of the tables and the edges represents the Foreign to Primary key relationships. Whenever the searching operation is performed, it is just enough to scan the data graph and no need for searching in database or datasets.

D.Schema graph:

The schema graph represents the graph structure where the nodes represents the entire schema. A schema can be described as the "layout" of a database that tells the way data is organized into tables.

E.Preprocessing Stage:

At the preprocessing stage, the data in database is modelled as either graph or index. The main advantage of maintaining this is to avoid searching the data in the database which will consume large amount of time in an efficient manner.

F.Query generation Stage:

At this stage, all the tuples[18] ,[19] that matches the users query are retrieved by either constructing queriesor by pointing to the positions of that keyword in the graph or indexwithout constructing an SQL queries.

G.Recommendation Stage:

This stage plays a more challenging part in all current researches of the keyword search which focus on retrieving the top answers[11],[20],[22] for the users query by various methods like calculating TF-IDF ranking methodology etc.

II.RELATED WORKS

In this section, different techniques used for keyword search in database are detailed.

Hristidis V and Papakonstantinou Y, 2006[5] proposed a work "DISCOVER" which performs keyword search over relational database.

At the preprocessing stage, the graph is maintained for the database. The graph used in DISCOVER is schema graph where the nodes represents the entire schema and the edges represents the Foreign to Primary key relationships. The schema graph is represented below

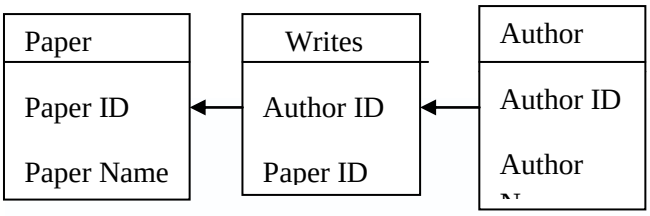

Fig. 1-Schema Graph

In the Fig. 1,at the query processing stage, when the user types the query which contains the finite set of keywords, a candidate network is generated from the schema graph. Candidate network is used in the filtration process that finds all the connections between tuples to the query keyword. This process is done by the breadth first traversal.

To produce the top k answers the number of joins to the query keyword is calculated. The score is incremented by one when the joins keep on increasing. The path that contains all the finite set of keywords that the user has mentioned with the least score is selected and this is used in calculating the ranking measures but these study is done in a theoretical way.

Maintaining a schema graph will have only attribute names. So in order to reach the keyword query another scanning in the database is needed this takes a long time. And top K answers is still in the theoretical phase which is great challenging nowadays in the keyword search over relational database.

Siva Kumar N and Goldman R, 2004[4] proposed a work "Proximity Search" in Database

Graph is maintained for the database at the preprocessing stage. Here the data in the database is modeled as data graph where the nodes are the tuples of the tables and the edges represents the Foreign to Primary key relationships. The data graph is represented in the below figure

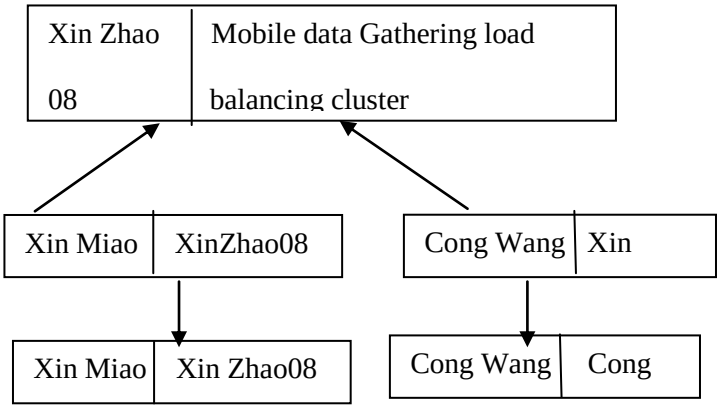

Fig. 2- Data Graph

In the Fig. 1, the database is represented as set of linked objects. For example consider movie database the objects here includes movie, actor, director etc. When the user types the Find query and near query, find query specifies the find set of objects. Ex "find movie". The near query specifies the near set of objects Ex "actor names". The main objective is to rank objects in the find set according to the distance to their near set. The distance information is provided by a distance module.

For effective distance computation, K-neighborhoods distance lookup tablewhich maintains distance for all the nodes to all other nodes. The hub is an indices that maintains all possible nodes for the database and also the shortest distance from a node to all other nodes by referring to the K-neighborhoods distance lookup table.

At the preprocessing stage maintaining graph and indices are done which requires more time. Whenever the indices table is been changed or added each time traversing has to be done which is more complex and also inefficient way.

Disha M and Bilimoria, 2015[2] proposed a work "Empirical Framework on Adaptive Keyword Query Searching in Linked Databases"

At the preprocessing stage keyword table, prefix table, inverted table is maintained. In the keyword table all the possible keywords along with unique key id is maintained for the database. In the inverted table the position of the keyword in the database is stored. The prefix table maintains the range of the prefix keyword that is to be searched.

When the user types the query keyword at the query processing stage, SQL query is constructed by the developers to retrieve the range of the keyword that is to be searched by using the above tables that is created at the preprocessing stage. And from that all the possible tuples that have the prefix equals to the keyword is retrieved.

It does not focus on the Top k answers. So the top answers are not viewed to the users instead all the tuples that contain the finite set of keyword is listed from the database. This stage plays more vital role in the recent search.

Though the creation of indices is easy but the top answers are not recommended to the users which makes this a great defect. And any insertion or deletion of rows in the database will initiate changes in indices and number of indices is also too high.

Silva AS and Mesquita F,2007 [8] proposed a work "LABRADOR"

It models the query and find the attribute that matches the query keyword and performs the searching as normal one. There is no preprocessing is performed.

When the user types a pair of query keyword, the search operation is performed on the domain of the attribute to find the corresponding attribute name by filtering the search on certain attributes that does not satisfy the basic conditions. For ex:if the keyword given is in terms of numbers (i:e: 23) search is performed only on the age attribute and not on other attributes that does not accept the input as the number. This validity is checked by the Labrador engine.

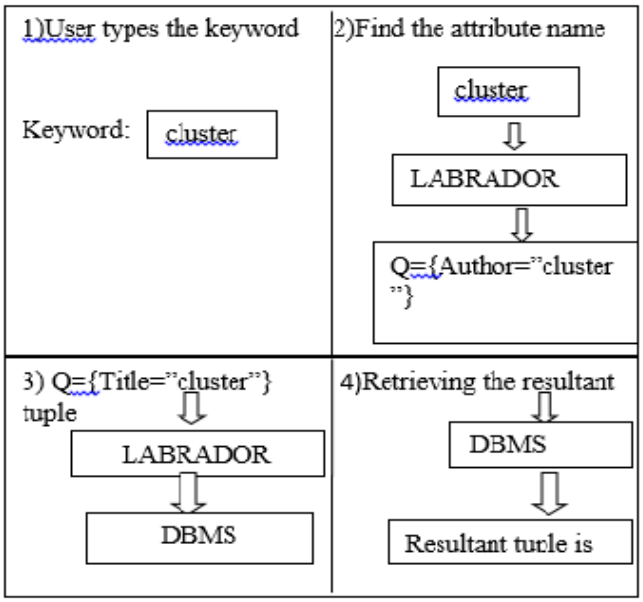

Fig. 3- LABRADOR

In the figure 3,in the worst case, if the answer generated at last found to be correct. Then definitely before arriving at the answer many test cases has been performed which is a waste of time that is generated is at last, is correct.If the matching goes wrong then all the process that are done till the last goes wrong and also entire result produced will be incorrect.

Baeza-Yates and Frakes W.B R,1992[3] proposed a work"IR-Style" on keyword search over relational database. :

It aims at producing top answers for the user's query. There is no preprocessing stage (ie no graph or indices)is maintained for the database. All values are referred from the database directly each time.

When user types their query, each time from the database it is referred and the results are displayed. At recommendation methodology in producing top results, TF-IDF ranking methodology is used. Since no graph or inverted file is maintained at the preprocessing stage. This would

make the searching process to refer the databases at each and every time of the user's query which is time consuming.

At recommendation methodology in producing top results,a novel based TF-IDF ranking methodology is used.

No preprocessing is maintained and searching is done from the database each time.So , therefore time consuming is more in retrieving the relevant information.

III. PROPOSED WORK

Our proposed work aims at finding the answer for keyword search which are more relevant to the user interest. In our proposed work similar interest users are grouped together using a clustering approach. For this at the preprocessing phase, index is built and maintained on the user interests. A term is entered into index if it has been used by more number of users.In the Fig. 4 when the user submits the user query(i.e. keyword), the keyword is been forwarded to the preprocessing phase where index and datagraph are been maintained for the dataset.

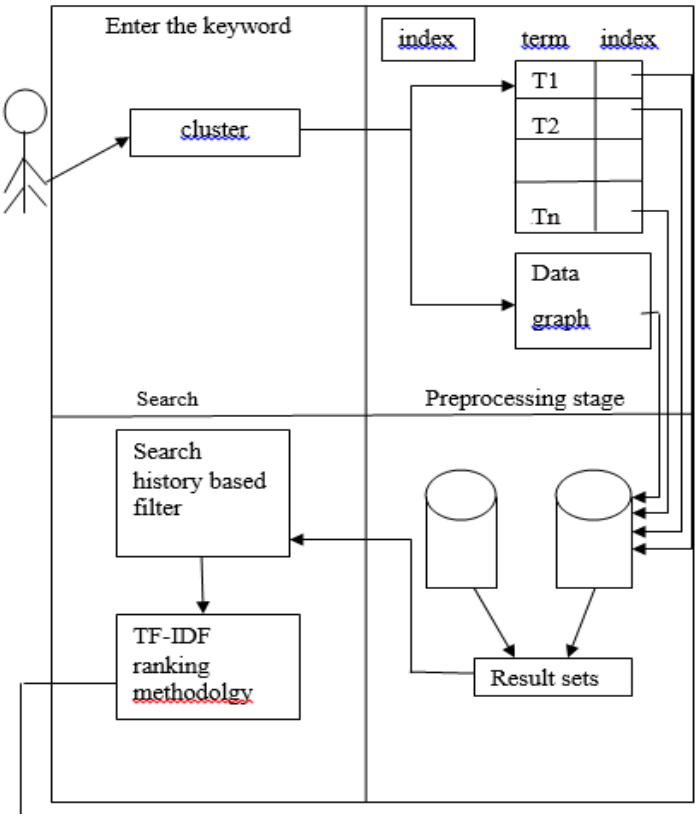

Fig. 4 -Architecture of our proposed work

If the keyword search term matches any entry in index, then relevant information is directly retrieved. Otherwise, keyword is searched on the data graph constructed from the dataset. All the matched results are collected from the query generation phase. During the recommendation phase, more relevant results are filtered from derived resultset using TF-IDF and user interest profile.

Initially, user is classified to search history of similar interest users. Based on the search hisinitial filtering is carried out for resultset. All filtered results are ranked based on TF-IDF methodology.

TF-IDF weighting:

Term frequency is a measure of how frequently a term occurs in a document.

$$tf_{t,d} = \frac{\text{Number of times term t appers in a document}}{\text{Total number of terms in the document}}$$ □ □ □ □ □

As in Eq. 1, how to calculate the term frequency for a particular term t for the document d is given

Inverse Document Frequency is a measure of how important a term is.

$$idf_{t,d} = \frac{\log_e(\text{Total number of documents})}{\text{umber of documents with term t in it}} \quad (2)$$

As in Eq. 2, how to calculate inverse document frequency for a particular term t for the document d is given

We now combine both to produce a composite weight for each term in each document.

$$tf\_idf_{t,d} = tf_{t,d} * idf_t \quad (3)$$

As in Eq. 3, the TF-IDF is calculated by combining the Eq. 1 and Eq. 2

IV. IllUSTRATION OF OUR PROPOSED WORK

When user enters a keyword "cluster", the consequences of the search in the proposed work is illustrated in this section.

Initially the result set of the query generation phase are produced either from the index or from the data graph at the preprocessing phase for the given keyword.The result set si forwarded to the recommendation phase. At the recommendation phase user is initially classified based on the similar interest users using the search history in the first step of that phase in which the keyword that is not present in the document is filtered out. All the filtered results of the search history are then ranked based on the weighting.

Let us explain the TF-IDF weighting for the users query "cluster" from the set of below given documents.

Doc 1:The game of cluster is a game of everlasting learning

Doc 2:The unexamined cluster is not worth living

Doc 3:Never stop learning

Calculating TF:

TF for cluster in Doc 1 =0.1 (ie 1/10)

TF for cluster in Doc 2 =0.1428 (ie 1/7)

Calculating IDF:

IDF for cluster =1+log(3/2)

=1+0.4055

IDF for cluster =1.4055

TF-IDF

cluster in Doc 1 =(0.1)*(1.4055)

=0.140550715

Similarly for Doc 2 and Doc 3

TABLE 1: Calculating TF-IDF

	Doc 1	Doc 2	Doc 3
Cluster	0.140550715	0.200786736	0

Similarly for each users query we can calculate TF-IDF and do the ranking as above,

CONCLUSION:

This paper discusses about different techniques related to improve the accuracy of keyword search. Also, the pros and cons of each method is detailed. Our survey reveals that many researchers are contributing to different phases of keyword search such as preprocessing, query generation & recommendation for achieving improved accuracy. Most of the methodology failed in aspects like more time to build index ,no relevant results are produced and also prior knowledge of SQL is needed. Our proposed work combines search history based index or TF-IDF methodology for faster retrieval and filtering out result respectively to keyword search.

RESULTS:

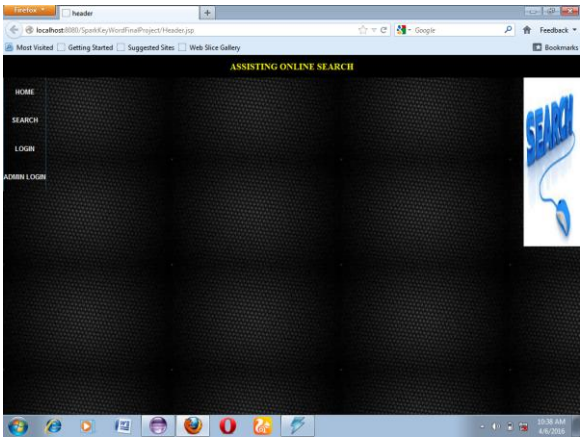

Fig 5-Home page

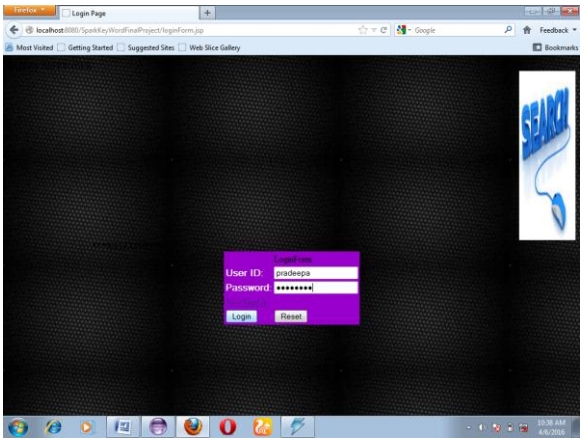

Fig 6-Login page

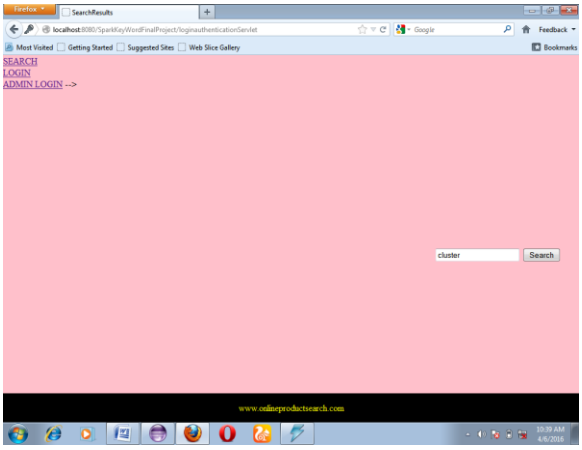

Fig 7-Keyword search

Fig 8-Candidate network

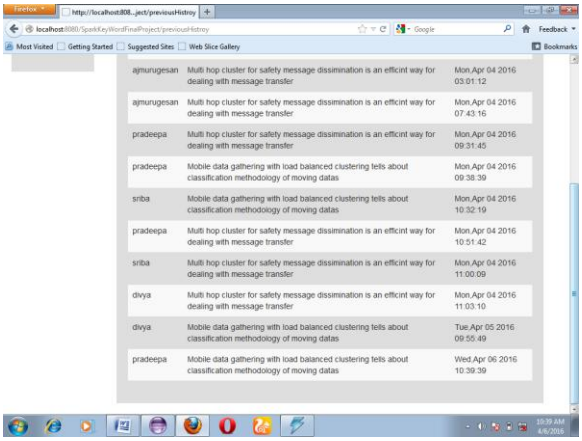

Fig 9-Previos history

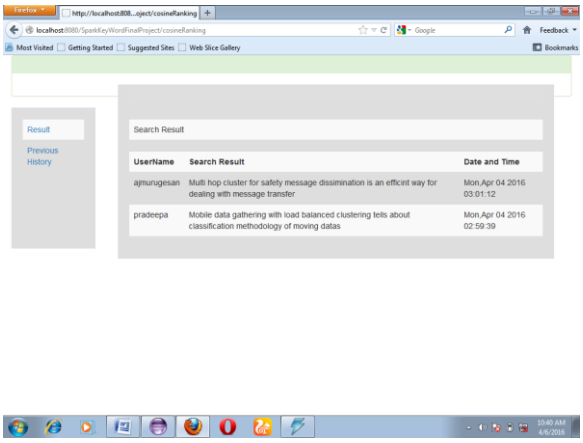

Fig 10-Filtered result

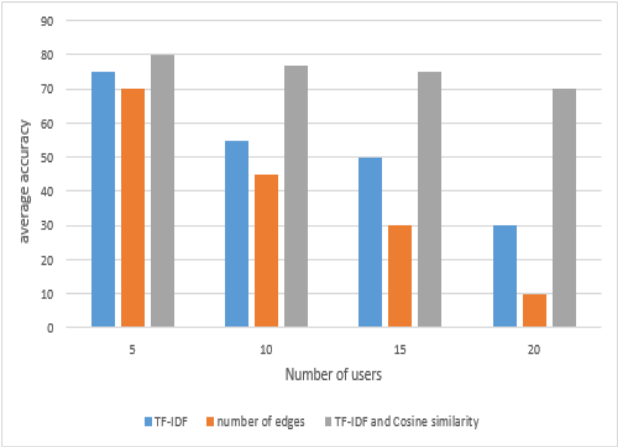

Fig 11-Compare accuracy in accordance with number of users

REFERENCES

1) Chavan Aparna1, Suvarna Bangar2 "Review on keyword search over relational database",In:Proceedings 4th International Conferenece on Emerging Technology and Advanced Engineering , pp 45-49,2014.

2)Disha M, Bilimoria,"An Empirical Framework on Adaptive Keyword Query Searching in Linked Databases", In:Proceedings 2nd International Conference on Computing for Sustainable Global Development, pp 35-41,2015.

3) Frakes W. B and Baeza-Yates R. A.," Information Retrieval: Data Structures and Algorithms" , 1992.

4) Goldman R, Shivakumar N, Venkatasubramanian S, Garcia-Molina H," Proximity search in databases" In: Proceedings of the 24th international conference on very large data bases, pp 26–37, 2004.

5) Hristidis V, Papakonstantinou Y ,"DISCOVER: keyword search in relational databases" In: Proceedings of the 28th international conference on very large data bases", pp 670–681, 2006.

6)Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", In Elseiver Inc, 2006.

7)L. Qin, J.X. Yu, and L. Chang, ―Keyword Search in Databases: The Power of RDBMS,‖ Proc. SIGMOD Int'l Conf. Management of Data, pp. 681-694, 2009.

8) Mesquita F, da Silva AS, de Moura ES, Calado P, Laender AHF." LABRADOR: efficiently publishing relational databases on the web by using keyword-based query interfaces. Inform Process Management, pp 983–100,2007.

9)M. Hua, J. Pei, A.W.C. Fu, X. Lin, and H.-F. Leung, ―Efficiently Answering Top-k Typicality Queries on Large Databases,‖ Proc. Int'l Conf. Very Large Data Bases (VLDB), 2007.

10) M. Jayapandian and H. V. Jagadish, "Automating the design and construction of query forms," IEEE Trans. Knowl. Data Eng.,vol. 21, no. 10, pp. 1389–1402, Oct. 2009.

11)Natsev A, Chang Y-C, Smith JR, Li C-S, Vitter JS (2001) Supporting incremental join queries on ranked inputs. In: Proceedings of the 27th international conference on very large data bases, pp 281–290, September 11–14, 2001.

12) S. Chakrabarti, and S. Sudarshan, Bhalotia, A. Hulgeri, C. Nakhe ―Keyword Searching and Browsing in Databases Using Banks,‖ Proc. Int'l Conf. Data Eng. (ICDE), pp. 431-440, 2002.

13)Tok Wang Ling ,Thuy Ngoc Le and Zhong Zeng "Towards an intelligent keyword search over XML and relational databases",In:Proceedings 2nd International conference on Big Data and Smart Computing, pp 1-6 ,2014.

14) Tong H, Faloutsos C, Pan J-Y Random walk with restart: fast solutions and applications. Knowl Inform Syst 14(3):327–346,2008.

15) V. Kacholia et al., ―Bidirectional Expansion for Keyword Search on Graph Databases,‖ Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 505-516, 2005.

16) Wang, and L. Zhou, ―Efficient Keyword Search for Valuable Lcas over XML Documents,‖ Proc. ACM Conf. Information and Knowledge Management (CIKM), 2007.

17) Wang S, Peng Z, Zhang J, Qin L, Wang S, Yu JX, Ding B NUITS: a novel user interface for efficient keyword search over databases. In: Proceedings of the 32th international conference on very large data bases, pp 1143–1146, September 12–15, 2006, Seoul, Korea,2006.

18) Wang S, Zhang K-L Searching databases with keywords. J Comput Sci Technol 20(1):55–62,2005.

19) Wang Z, Wang Q, Wang D-W Bayesian network based business information retrieval model. Knowl Inform Syst 20(1):63–79,2009.

20) Wan X Beyond topical similarity: a structural similarity measure for retrieving highly similar documents. Knowl Inform Syst 15(1):55–73,2008.

21) Zhang J, Peng Z-H, Wang S, Nie H-J CLASCN: candidate network selection for efficient top-k keyword queries over databases. J Comput Sci Technol 22(2):197–207,2007.

22) Zhu X,Chen C,Shen p,Zomaya A Y "An Efficient Ranked keyword search Method", IEEE Trans,Parallel and Distributed System.,vol. 18,no. 10,pp. 1045-9219, Apr 2015.