Keyword Query Routing At Advance Level

Pruthika Patel, Prof Chetna Chand

[1] *Student, Computer Engineering, Kalol Institute of Technology and Research Center, Gujarat, India*
[2] *Professor, Computer Engineering, Kalol Institute of Technology and Research Center, Gujarat, India*

ABSTRACT

Keyword search is a familiar and potentially effective way to find information of interest that is "locked" inside databases. To route keywords only to the relevant sources to reduce high cost of processing keyword search queries over all sources. In the proposed system first Pre-process Text, Build a network of keyword, Rank the Node, Remove common ranked node, Store them in a table. Then finally display the result. The proposed system uses routing keyword search for queries having many keywords. This improves the performance of keyword search. This way can greatly reduce time costs.

Keyword: - *Keyword Search, Linked Data, Keyword Query Routing, and Ranking*.

1. INTRODUCTION

Web has evolved from a global information space of linked documents to one where both documents and data are linked. Linked data [1][3][4][10][12] is an approach to publishing and sharing data on the web. Data from different domains i.e. companies, people, books, films, scientific publications, television ,music, and radio programs, proteins, genes, clinical trials and drugs, online communities, scientific data and statistical, and reviews. In database research, solutions have been proposed, which given a keyword query, retrieve the most relevant structured results or simply, select the single most relevant databases .However, these approaches are single-source solutions. They are not directly applicable to the web of Linked Data, where results are not bounded by a single source but might encompass several Linked Data sources. Existing work uses keyword relationships [1][2][5] collected individually for single databases.

2. LINKED DATA

The web is collection of textual documents and also interlinked data sources. Linked data is about using the web to connect related data that wasn't previously linked, or using the web to lower the barriers to linking data currently linked using other methods. Linked data compromise hundreds of sources containing millions of links. Same-as-link denotes two RDF resources represents same real world. Linked data is given in fig. 1.

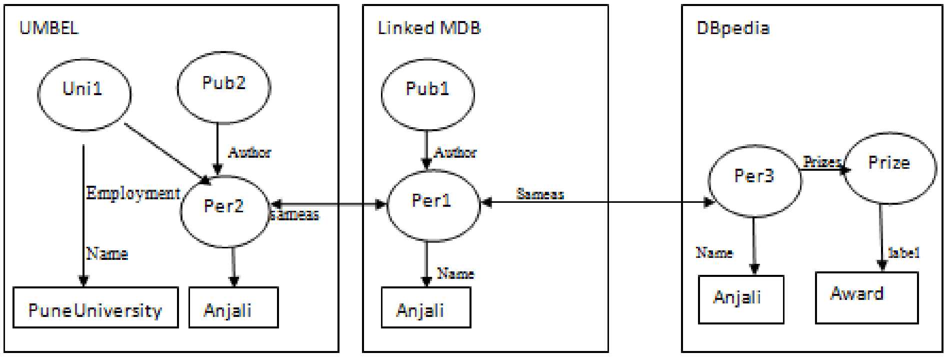


Figure 1: Extract of the web data graph

It is difficult for any non technical web users to get the data. Technical users have knowledge of SQL language, so that he can easily exploit the web data. Every user can get data using keyword search. Keyword search do not required any knowledge of structured queries i.e. query languages, the schema, or the underlying data. There are generic Linked Data browsers which allow users to start browsing in one data source and then navigate along links into related data sources. There are Linked Data search engines that crawl the Web of Data by following links between data sources and provide expressive query capabilities over aggregated data, similar to how a local database is queried today. The Web of Data also opens up new possibilities for domain-specific applications. Unlike Web 2.0 mashups which work against a fixed set of data sources, Linked Data applications operate on top of an unbound, global data space. This enables them to deliver more complete answers as new data sources appear on the Web.

3. APPROCHES OF KEYWORD SEARCH

There are some approaches for keyword search as per below.

3.1 Schema Based Approaches

There are schema-based approaches [1][2][3][4][5][9][10][11] implemented on top of off-the-shelf databases. A keyword query is processed by mapping keywords to elements of the database (called keyword elements). Then, using the schema, valid join sequences are derived, which are then employed to join (“connect”) the computed keyword elements to form so-called candidate networks representing possible results to the keyword query.

A system called Kite extends schema-based techniques to find candidate networks in the multisource setting. Many practical settings, however, require that they combine tuples from multiple databases to obtain the desired answers. Such databases are often autonomous and heterogeneous in their schemas and data. Kite gives a solution to the keyword-search problem over heterogeneous relational databases. It employs schema matching techniques to discover links between sources and uses structure discovery techniques to find foreign key joins across heterogeneous sources. Such joins are critical for producing query results that span multiple databases and relations. Kite operates in two phases: offline preprocessing and online querying. In the offline preprocessing *phase* (Fig. 2), the index builder constructs standard inverted IR indexes on the text attributes of the databases. Then, the FK join finder leverages data-based join discovery and schema matching methods to identify FK joins across the databases. In the online querying phase, given a top-k keyword query Q, the condensed candidate network (CN) generator employs the FK joins and the IR indexes to quickly identify a space of possible answers to Q.

3.2 Schema -Agnostic Approaches

Systems for Schema-agnostic keyword search on databases, such as DBXplorer , BANKS and Discover , model a response as a tree connecting nodes (tuples) that contain the different keywords in a query (or more generally, nodes that satisfy specified conditions). Here “schema-agnostic” [1][2][3][4][5][9][10][11] means that the queries need not use any schema information (although the evaluation system can exploit schema information). For example, the query “Gray transaction” on a graph derived from DBLP may find Gray matching an author node, transaction matching a paper node, and an answer would be the connecting path; with more than two keywords, the answer would be a, connecting tree. The tree model has also been used to find connected Web pages, that together contain the keywords in a query.

Schema-agnostic approaches operate directly on the data. Structured results are computed by exploring the underlying data graph. The goal is to find structures in the data called Steiner trees (Steiner graphs[1][2] in general), which connect keyword elements.

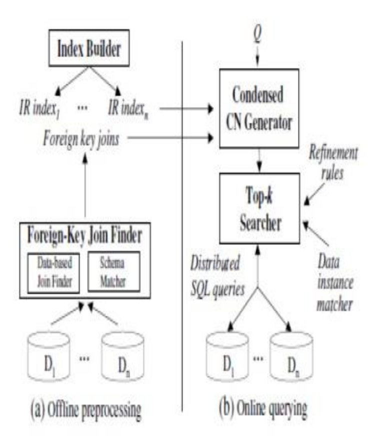


Figure-2 Kite architecture

3.3 Database Selection

A database is relevant if its keyword relationship model covers all pairs of query keywords. MKS [1][2][3][4][5] captures relationships using a matrix. Since M-KS considers only binary relationships between keywords, it incurs a large number of false positives for queries with more than two keywords. This is the case when all query keywords are pairwise related but there is no combined join sequence which connects all of them.G-KS [1][2][3][4][5] addresses this problem by considering more complex relationships between keywords using a keyword relationship graph (KRG)[1]. Each node in the graph corresponds to a keyword. Each edge between two nodes corresponding to the keywords <ki, kj> indicates that there exists at least two connected tuples ti →tj that match ki and kj. Moreover, the distance between ti and tj are marked on the edges.

4. THE PROPOSED SYSTEM

4.1 Proposed Algorithm

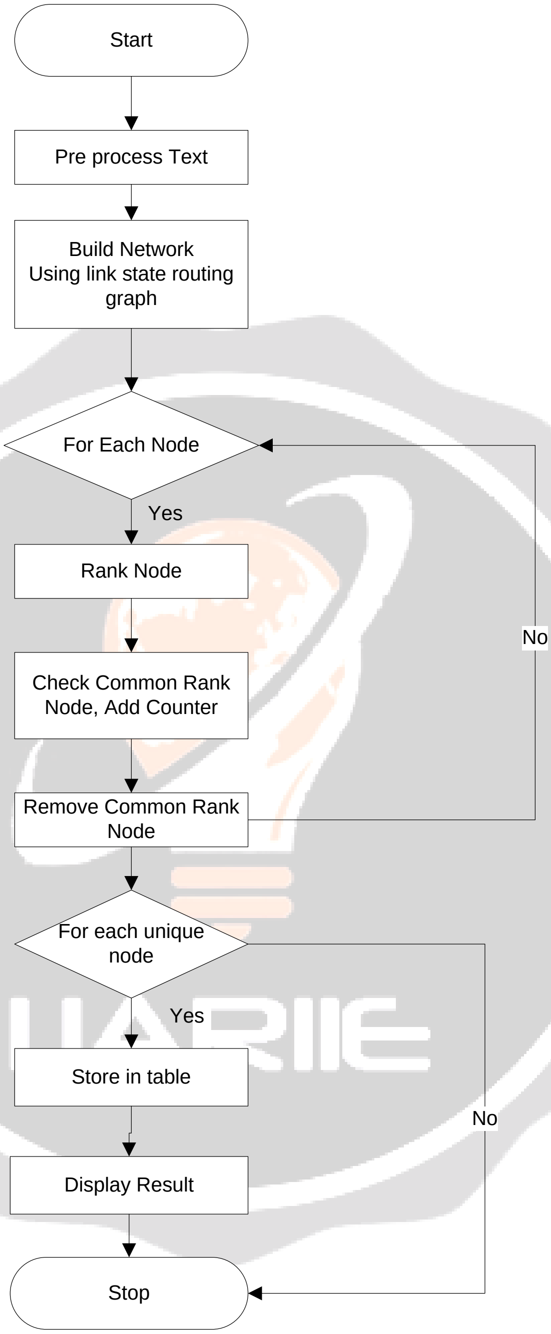
To route keywords only to the relevant sources to reduce high cost of processing keyword search queries over all sources. Aim is to improve the performance of keyword search. In the proposed system first Pre-process Text, Build a network of keyword, Rank the Node, Remove common ranked node, Store them in a table. Then finally display the result.

Proposed algorithm and flowchart are as following.

1) Remove all stop words from the text(eg for, the, are, is , and etc.)
   2) create an array of candidate keywords which are set of words separated by stop words
   3) find the frequency of the words – using stemming algorithm
2) Build a network of keywords (repeat until all keywords and docs are being referred)
   a. Create a G-KS approach based graph as per the degree of the each word with the keyword as node and its link as edge. (Degree of a word is the number of how many times a word is used by other candidate keywords)
   b. For every candidate keyword find the total frequency and degree by summing all word's scores.
   c. Finally degree/frequency gives the score for being keyword.
3) Create a single database with attributes based on graph

4) Display the results

4.2 Flow chart of Proposed Method



5. EXPIREMENTAL RESULT

The number of experiments done to evaluate the performance of Keyword Search.

For the searched keyword, find the related keyword by online.
So it results number of related keywords.

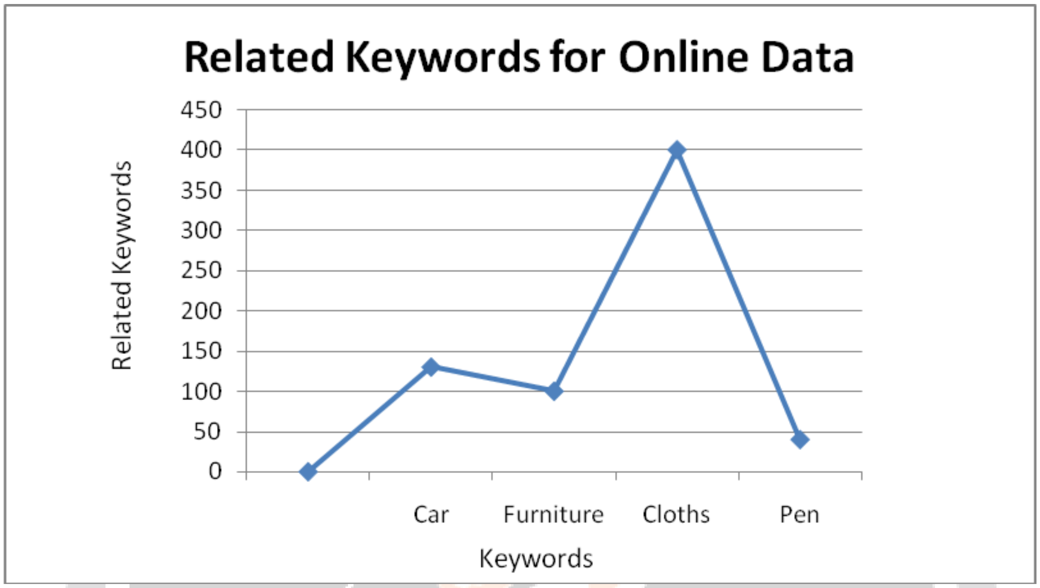


Fig 3 Keywords vs Related Keywords

Also for the searched keyword, find the related keyword by offline. So it results number of related links which consist searched keyword.

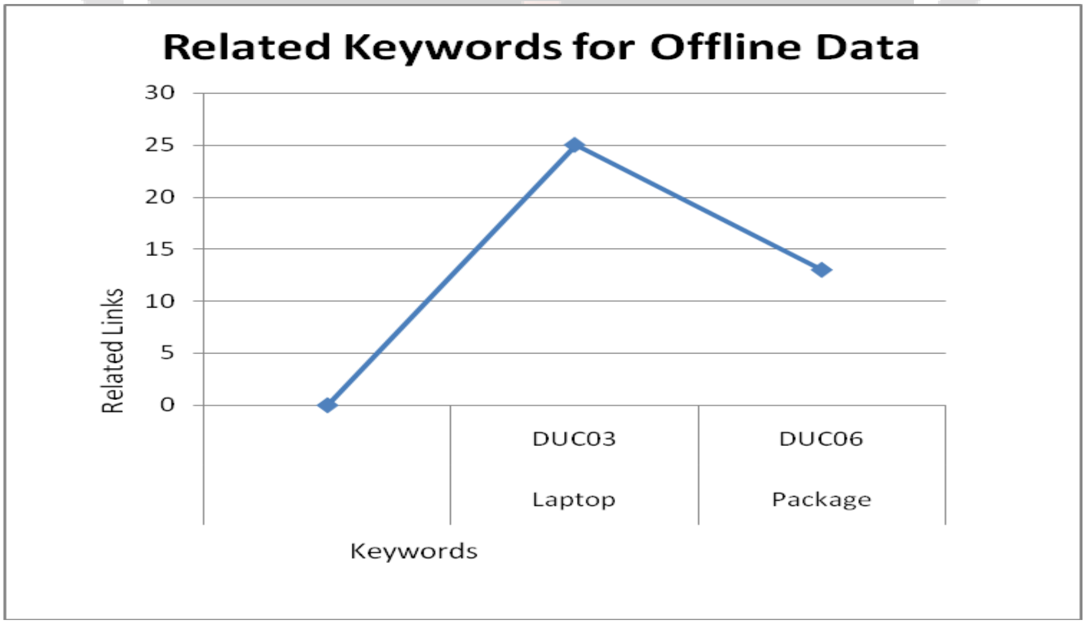


Fig 4 Keywords vs Related Links

Finally the time needed to search keywords are displayed as shown below.

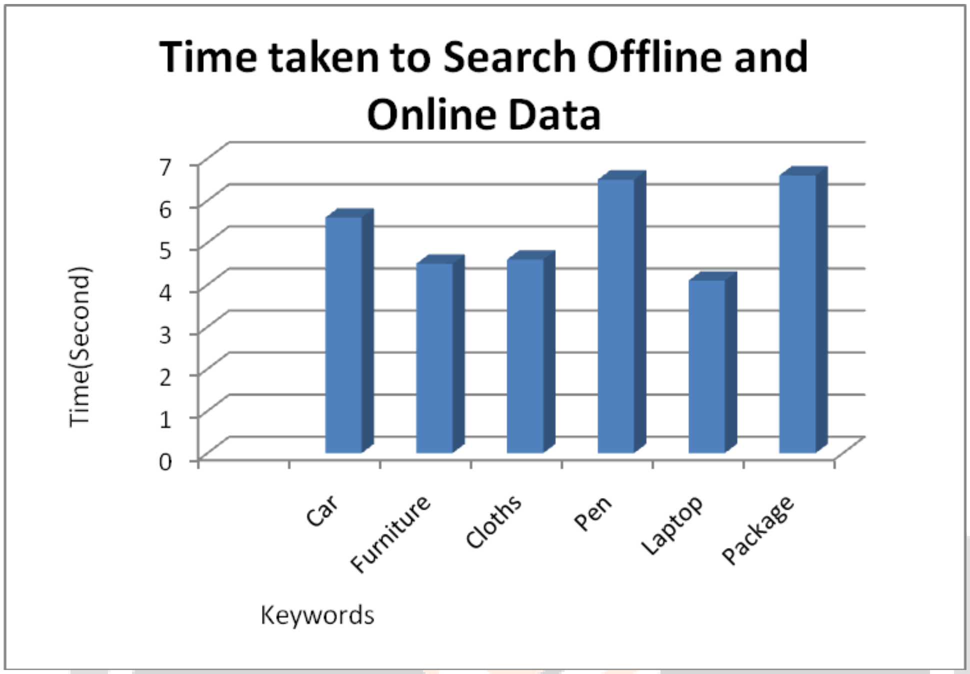


Fig 5 Keywords vs Time(Second)

6. CONCLUSION

Existing keyword query routing methods focus on to generate only routing plans based on keyword and element relationships at the level of sets. In the proposed system first Pre-process Text, Build a network of keyword, Rank the Node, Remove common ranked node, Store them in a table. Then finally display the result. The experiments are done on DUC datasets for offline mode and some website links for online mode. Here, comparative study between proposed system and existing system is studied. Results shows that it requires average 5 second to search keywords.

7. REFERENCES

[1] Thanh Tran and Lei Zhang, "Keyword Query Routing", IEEE Transactions, VOL.26, NO.2, February 2014.

[2] Mrs. Suwarna Gothane, Srujana.P, "Approaches for Keyword Query Routing**",** *Int. Journal of Engineering Research and Applications ISSN: 2248-9622, Vol. 4, Issue 10(Part -1), October 2014, pp.16-22*

[3] Pawar Prajakta Bhagwat et al, "Efficient Keyword Query Routing for Search Engines", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (1) , 2015, 434-437

[4] Prachi M. Karale1**," A** Survey on Keyword Query Routing ", *International Journal of Advance Research in Computer Science and Management Studies Volume 2, Issue 11, November 2014 pg. 312-320*

[5] Pawar Prajakta Bhagwat **,"** Query Expansion for Efficient Keyword Query Routing System **"** *International Journal of Advanced Research in Computer Science and Software Engineering 5(7), July- 2015, pp. 1018-1024*

[6] B. Yu, G. Li, K.R. Sollins, and A.K.H. Tung, Effective Keyword-Based Selection of Relational Databases”, Proc. ACM SIGMOD Conf., pp. 139-150, 2007.

[7] Q.H. Vu, B.C. Ooi, D. Papadias, and A.K.H. Tung, “A Graph Method for Keyword-Based Selection of the Top-K Databases,” Proc. ACM SIGMOD Conf., pp. 915-926, 2008.

[8]. Vikram Singh and Balwinder Saini, ” AN EFFECTIVE TOKENIZATION ALGORITHM FOR INFORMATION RETRIEVAL SYSTEMS”, pp. 109–119, 2014.

[9] Junjie Yao, Bin Cui, Liansheng Hua, Yuxin Huang, “Keyword Query Reformulation on Structured Data”, 2012 IEEE DOI 10.1109/ICDE.2012.76[10] Ms.Tejashree R. Shinde,” An Efficient Schema Free Keyword Search Approach Over Linked Data using Lucene Indexing Algorithm” , International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4496-4500

[11] Ms. J. Laveena Grasy, Ms R. Soundharya and Ms. Dhanalakshmi, “ An Effective Mining Approach Using Query Routing ” INTERNATIONAL JOURNAL FOR RESEARCH IN EMERGING SCIENCE AND TECHNOLOGY, VOLUME-2, ISSUE-5, MAY-2015 [12] Efficient optimization and processing of queries over Text rich graph structured data by Gunter ladwig, Scientific publishing