ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 6, June 2016

Graph Based Keyword Search Techniques over Database

Patil Sweta
Dept. of Information Technology
SGGS IE &T, Nanded.

Mrs. J. V. L. Megha
Head.Dept.of Information Technology
SGGS IE &T, Nanded.

***Abstract* - Now a day's World Wide Web is huge source of information as well as widely used by all the people where internet is large network of websites. All of these websites are administered by database system. Therefore it becomes necessary to search databases efficiently while working with web. In this paper we present a survey work on keyword search which uses graph for navigation. The techniques like BANKS, DataSpot, Proximity Search, are studied.**

***Index Terms* - Keyword Search, Graph, Database, Keyword Query Interface**

I. INTRODUCTION

Structured Query Language is basic approach for retrieving data from database. The end user, who wants data form database, must have to know SQL and scheme of database. Though all the information on the web is stored in database, there must be some easier way to interact with web and to accomplish this need Keyword Query Interface has been developed. Adapting this technique keyword search over relational database produces some immediate and efficient result.

Therefore, keyword search over database has become important research area over last few years. Several prototype techniques like DBXplorer, BANKS, Dataspot, DISCOVER have initiated the problem of keyword search very efficiently[1][2][3]. Those techniques answers keyword queries effectively and return tuple which contain all or most of the query keywords.

Many Vertical Search Engines also called as Topical Search Engines such as Amazon.com, Flipkart or any shopping site are driven by keyword search[9]. In those applications databases are typically entity databases where each tuple correspond to real world entity. This paper is organized as follows section 2 describes what graph based keyword search is and short summary of techniques based on graph based keyword search. Section 3 explains all the techniques in detail. Section 4 will give the conclusion that among these techniques which is good technique for particular application. Section 5 contains references.

II. GRAPH BASED KEYWORD SEARCH

The Information Retrieval technique provides facility of querying unstructured or structured data using keywords where as relational database system uses structured query language to interact with database therefore integration of these two system provides flexible ways for users to query the information[7].

In this section, I will explain the basics of graph based techniques. There are two types of approach for processing keyword query, first one is Schema Based Approach and second one is Graph Based Approach. A Schema Based Approach supports keyword search in RDBMS using SQL. Two main steps in this approach how to generate set of SQL queries that can find all the structures among tuple and how to evaluate generated set of SQL queries efficiently.

A data graph G_D can be considered as manifestation of Relational Database. Now it is explained that how to answer keyword queries using graph algorithms. We consider weighted directed graph that is $G_D(V, E)$. Weights are assigned to edges to show the proximity of the corresponding tuples, denoted as $W_e\{(u,v)\}$. For a foreign key of tuple t_u to t_v the weight for directed edge (u,v) is given in following equation Eq. 1 and weight for backward edge is given in Eq.2

$$w_e\{(u,v)\}=1 \quad (1)$$

$$w_e\{(u,v)\}=\log_2(1+N\_in(v)) \quad (2)$$

Where $N_{in}(v)$ is the number of tuples that refer to t_v, which is the tuple corresponding to node v. There are different structures of tuples to be returned that is tree-based semantics and subgraph-based semantics. In tree based semantics an answer to Q i.e. Q-SUBTREE is defined as any sub-tree T of G_D that is reduced with respect to Q. The Q-SUBTREE is popularly used to describe answers to keyword queries. To rank Q-SUBTREE two different weight functions are used in increasing weight order. Two semantics are proposed based on two weight functions, namely steiner tree-based semantics and distinct root based semantics.

III. GRAPH BASED KEYWORD SEARCH TECHNIQUES

a) BANKS

BANKS is an acronym for **Browsing and Keyword Searching.** BANKS is a system for keyword search on relational database which allows data and schema browsing together[1][2].

Model Description: In BANKS system database is modeled as directed graph, tuple is modeled as node. The edge of graph is foreign key primary key link between the corresponding tuples. This process makes graph model for BANKS technique. The answer for query is subgraph which is formed by connecting nodes matching query keyword. Practically it is not the final answer as just by looking at subgraph we cannot say that this subgraph contains all the information what we requested.

We have to indentify central node in a graph that connects all the keyword nodes and relationship between them. This central node can be called as root node and answer can be considered as rooted directed tree which contains directed path from root to each keyword node. The root node is information node and tree is connection tree.

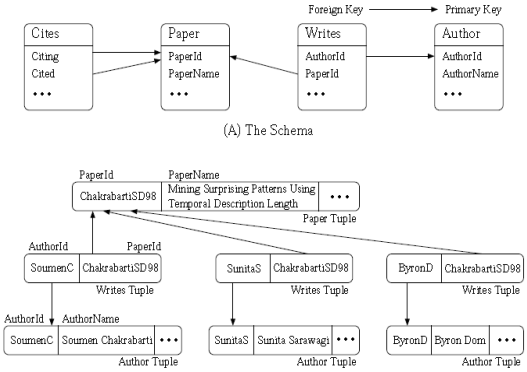

Figure 1:The DBLP Bibliography Database(A Fragment of Database.)

BANKS used two datasets one is DBLP dataset as shown in Figure 1, they converted DBLP into structured relational format. There were 124,612 nodes and 319232 edges on the graph. A second is small thesis database.

A formal database model of BANKS consists of vertices, edges, edge weights, node weights. For each tuple T in database the graph has corresponding node μ_T. let's say there are two tuples *T1*& *T2* such that they have foreign key relationship then graph contains an edge from μ_{T1} to μ_{T2} and back edge from μ_{T2} to μ_{T1} as well. The weight for forward link along foreign key relationship reflects proximity relationship between the tuples and it is set to 1 by default. In the current implementation of BANKS the forward edge weight is set to $s(R(u),R(v))$ and the reverse edge weight is set to $[s(R(v),R(u))*IN_v(u)]$ and the actual weight is the minimum of the two as follows:

$$b(u,v)=min(s(R(u),R(v)), s(R(v),R(u))*IN_v(u))$$

BANKS present model for answering queries as let the query consist of n terms, t_1, t_2, t_3,...t_n then the query is conceptually answered as for each term t_i in the query we find set of nodes that are relevant to search term. Let the set is S_i and $S=\{S_1,S_2.....S_n\}$. A node is relevant to search term if it has a search term as its attribute value, node may also be relevant through metadata. A metadata may be column, table or view name. Keyword searching in BANKS is done using proximity based ranking, where foreign key and other key links are used. For browsing, BANKS provide rich interface for browsing through relational database and automatically generates hyperlinks corresponding foreign key and other keys link on displayed result. BANKS is used to publish organizational data, bibliographic data and electronic catalogs.

b) DataSpot

DataSpot is a database publishing tool which helps end user to explore large database without using any query language. DataSpot uses a schema-less semi-structured graph called a hyperbase. Search Server in of DataSpot performs searches within the hyperbase, and return answers to the user either in HTML pages or through an object API[5]. The DataSpot has been successfully deployed in diverse application areas including electronic catalogs, yellow pages, classified ads, help desks and finance.

Model Description:

A graph structure used in dataspot for data representation is called hyperbase. A hyperbase is made up of nodes, edges, and node labels. Nodes are connected via directed edges. There are two types of directed edges are used simple edge and identification edge. The simple edge is used to connect parent node to child node and also represent inclusion. These set of children nodes are ordered and parent nodes are not ordered. Labels should be used for leaf nodes and non-leaf nodes doesn't require label. The simple edges cannot make a cycle.

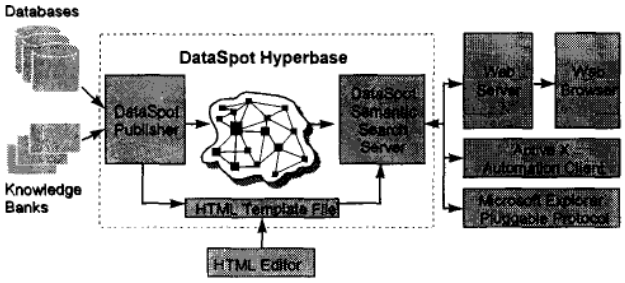

Figure 2: The Dataspot Architecture

The second type of edge used is identification edge used indicate reference and subject node relationship. The reference node uniquely identifies subject node and subject node can have maximum single reference node. Initially nodes in hyperbase represent data objects and edges represent association between them. An

equivalence relation is defined on nodes in hyperbase recursively as two nodes A & B in H hyperbase are equivalent if one of the three conditions is satisfied, A & B are leaf nodes and have identical node labels, if there is equivalent reference between A & B, if A & B have no reference and have equivalent list of ordered set of children. The hyperbase H is normalized if only leaf nodes have labels and if two distinct nodes in H are equivalent.

The result of dataspot query is a list of answers, where an answer is connected hyperbase. The answer to query is ordered according to their weights. The score of an answer is based on the size of hyperbase and strength of associations. There is also a provision of continuous query where keyword of this query will be form the answer of previous query.

c) Proximity search

Proximity search enables find out queries answers based on general relationships among objects, this techniques helpful for interactive query sessions. The motivation behind proximity search is database of Internet Movie Site Database. It consist of 140,000 movies over 500,000 film industry workers. The idea took place that this database can be viewed as set of linked objects, where objects represent movies, actors, directors and so on. Here we can define distance function based on links separating objects[3].

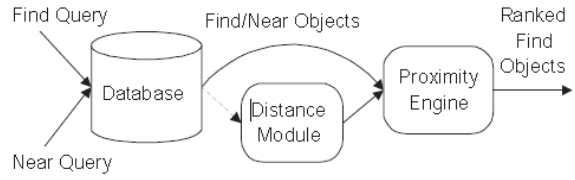

Figure 3: Proximity Search Architecture

Model Description:

The database is viewed as a graph where data (objects) is represented as vertices and relationship as edges this does not mean that underlying database system must manage its data as graph. Proximity depends on shortest distance between objects. For proximity searching database is viewed as collection of objects where objects are related by distance function.

The figure 3 shows Proximity Search Model. In the above model database system stores a set of objects, find and near queries are provided through application then find and near set queries are evaluated by database to generate find and near object result set in order to forward it to proximity engine. The proximity engine deals with object identifiers only. Distance module provides distance information to proximity engine based in this engine will re-rank the find and near set object result set. The proximity search used IMDB and DBG group dataset.

The object may be tuples, records or actual objects. The distance function is provided by the system or an administrator. The basic idea is to rank the objects in one given set (find set) based on the proximity to objects in another set (near set) in assumption with objects are connected by given numerical distances.

IV. EVALUATION

In this section we will compare all these techniques with each other. All of these techniques are used for retrieving data from database but by using little different functionalities. The performance, functioning, structural evaluation is done here to understand these three different techniques from each other.

BANKS and DataSpot are closely related to each other. DataSpot and BANKS computes relevance scores for trees and returns trees of maximum relevance but the graph formation is different. BANKS takes references only form equivalence edges where as it is explicitly stated in DataSpot. In BANKS system edges can have attributes such as weight or type therefore it can model a separate system.

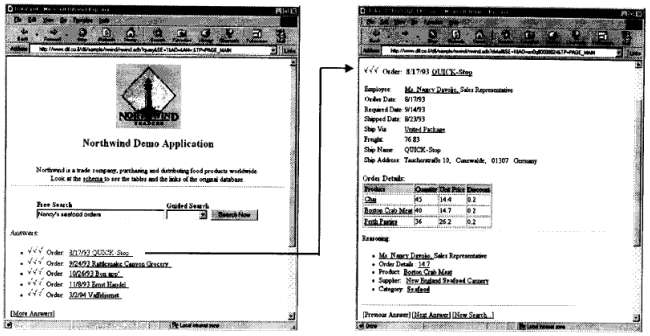

Figure 4: The Query Nancy Sea Food Orders & answer (DataSpot)

In DataSpot System Microsoft's Northwind Traders database is used, a demo database consisting of 8 tables, the query as shown in figure will return records from order table. The first record as shown in figure represents an order form QUICK-STOP that was processed by Nancy Davolio. The order contains sea food in this way the answer is related to the query which is ranked first in the results.

Find picture Near China	Photos of 6 Chinese students, followed by Prof. Widom, who advises 3 of them, and Prof. Ullman, who advises 2
Find publication Near Garcia	All of Prof. Garcia-Molina's publications, followed by publications of his students
Find publication Near Garcia Widom	The top publications are co-authored by Profs. Garcia-Molina and Widom, followed by their individual papers
Find group_member Near September	The top results are members born in September
Find publication Near OEM	The top pub. has "OEM" in its title, followed by a pub. stored in "oem.ps," followed by one with keyword "oem"

Figure 5: Result of Proximity search Stanford database Group Keyword Search

The proximity search result is as shown in above figure, several query results are summarized over a database describing the members, projects, & publications of Stanford database group. The database has been build form scratch in OEM, which contains 4200 objects, and 3600 edges.

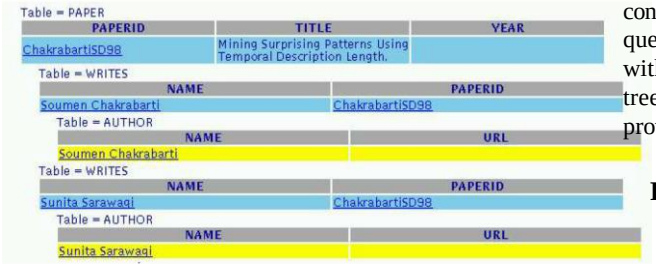

Figure 6: Result of query "soumen sunita".

BANKS examines few queries on bibliographic database. The query soumen sunita returned only one answer i.e a tree contains node corresponding to the paper as the information node. This paper has soumen and sunita co-auther and tree has corresponding author tuples as leaf nodes and two writes tuples connecting two author tuple as intermediate node.

In BANKS we assign weights to back edges based on in-degree as well as we can use the weight of nodes this is not specified in DataSpot. The use of node weights played vital role in effective web search as well as database search. BANKS also take in account the effect of metadata queries which is also not available in DataSpot.

Proximity Search also uses the idea of representing database as a graph like BANKS. The Proximity Search takes queries in the form of near object and find object. Proximity is the basic concern for building graph. The node weight concept is not used in proximity search which makes it less usable than BANKS. BANKS uses many ranking criteria for answer ranking default is indgree value. BANKS concentrate on keyword queries rather than focusing on graphical user interface. As shown in above figures BANKS has returned the closest tuple to query term for this complicated query. Therefore BANKS is one of the efficient methods to use for database retrieval.

BANKS is accessible over the web at URL: http://www.cse.iitb.ac.in/banks/ *[1]*

V. CONCLUSION

Though these several techniques available, we must address several issues related to keyword search over database. All of the keyword search technique listed above basically returns tuples whose attribute value contain all keyword present in query or most of the keywords form query. This approach might work satisfactorily in some cases, but still suffers from several limitations especially in the context of relational databases. These keyword searching approach may not return all matching result or may return irrelevant results. Now a day's user wants more satisfactory answers for their complex queries. Therefore we should develop techniques for keyword retrieval which does not suffer from inaccuracy and produce more and more relevant answers. As evaluated in section IV BANKS is most efficient and can be used in wide verity of applications which works directly with database not like like proximity model which need object model which results on some time consuming process. BANKS can search very complex queries such as shown in figure where direct relevance is with soumen according to keyword but its showing whole tree where user gets detailed information with respect to provided query.

REFERENCES

[1] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword searching and browsing in databases using BANKS," in *Proc. 18th ICDE*, San Jose, CA, USA, 2002, pp. 431–440.

[2] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. Technical report, Indian Institute of Technology, Bombay, November 2001.

[3] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H. Garcia-Molina. Proximity search in databases. In *Proc. of the Int'l Conf. on VLDB*, pages 26–37, 1998.

[4] N. L. Sarda and A. Jain. Mragyati: A system for keyword based searching in databases. Report No. cs.DB/011052 on CORR (http://xxx.lanl.gov/archive/cs), 2001.

[5] S. Dar, G. Entin, S. Geva, and E. Palmon. DTL's DataSpot: Database exploration using plain language. In *Proc. of the Int'l Conf. on VLDB*, pages 645–649, 1998.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 1998.

[7] H. Schutize, C. D. Manning, P. Raghavan. An Introduction to Information Retrieval

[8] Kopytoff, V. (2005) New search engines narrowing their focus, San Francisco Chronicle, Monday, April 4, 2005,

[9] Kevin Curran, Jude Mc Glinchey Vertical Search Engines, ITB Journal, December 2007.

[10] U. Masermann, G. Vossen SISQL: Schema Independent Database Querying, 0-7695.0789-1100 @2000 IEEE

[11] A Thesis submitted by Vagelis Hristidis, Keyword Search in Structured and Semistructured Database to the university of California.