Relational Keyword Search System

Pradeep M. Ghige[#1], Prof. Ruhi R. Kabra[*2]

[#]*Student , Department Of Computer Engineering, University of Pune, GHRCOEM, Ahmednagar, Maharashtra, India.*

[*]*Asst. Professor, Department Of Computer Engineering, University of Pune, GHRCEM, Pune, Maharashtra, India.*

[1]pradip.ghige@gmail.com

[2]ruhi.kabra@raisoni.net

Abstract— Now a day's for extending the keyword search to relational data set has been an area of research within the database and Information Retrieval. There is no standardization in the process if information retrieval, which will not clearly show the actual result also it displays keyword search without ranking and Execution time is more in existing system. We propose a system for; performance evaluation of relational keyword search systems. In the propose system combine schema-based and graph-based approaches and propose a Relational Keyword Search System to overcome the mentioned disadvantages of existing systems and manage the information and user access the information very efficiently.

The objective of this technique is to manage Information, Database and Information Retrieval systems involved independently and developed their own unique systems to allow users to access information. We also explore the relationship between execution time and factors. The proposed search technique will overcome the poor performance for datasets exceeding tens of thousands of vertices..

Keywords— keyword search; information retrieval; ranking; relational databases; data mining; database queries; search engine

INTRODUCTION

With the growing use of internet more and more people search the data on internet. Advents of Internet, it became possible to store a large amount of information. Several techniques are used to Information Retrieval (IR). Keyword search is one of the techniques use for the same. Keyword search is possible on both structure and semi-structure databases, also it possible on graph structure which combines relational, HTML and XML data. In relational databases the keyword search is used to find the tuples in by giving queries. Keyword search use number of techniques and algorithm for storing and retrieving data, less accuracy, does not giving a correct answer, require large time for searching and large amount of storage space for data storage.

We propose a system to overcome the disadvantages which discussed for efficient keyword search. Data mining or information retrieval is the process to retrieve data from dataset and transform it to user in understandable form, so user easily gets that information. One important advantages of keyword search is user does not require a proper knowledge of database queries. User easily inserts a keyword for searching and gets a result related to that keyword. Keyword search on relational databases find the answer of the tuples which are connected to database keys like primary key and foreign keys.

So we also present which comparative techniques used for keyword search like DISCOVER, BANKS, BLINKS, EASE, and SPARK. One important thing is that any existing techniques for information retrieval on real world databases and also experimental result indicate that existing search techniques are not capable of real world information retrieval and data mining task.

I. RELATED WORKS

Relational Keyword search are change for different applications and retrieval systems are different for that purposes. Requirement of applications change as per its use and also change algorithm and techniques, also vary as per requirement. One technique is not fulfilling the requirement of other dataset. In this section we will discuss all the research and techniques which are available in existing approaches.

A. Schema based approaches

Schema based approaches support keyword search over relational databases using execution of SQL commands [1]. These techniques are combination of vertices and edges including tuples and keys (primary and foreign key). There are some techniques are existed for schema based approaches.

DISCOVER

DISCOVER is the techniques that multiple Information Retrieval approaches follow. DISCOVER allows it's user to issue keyword queries without any knowledge of the databases schema or SQL [2]. DISCOVER returns qualified joining network of tuples, which is set of tuples that are associated because they join on their primary and foreign keys, collectively contain all the keywords of the query.

DISCOVER proceeds in two steps- 1. The candidate's network generator generates all candidate networks of relations.2.Plan generators builds plats for the efficient evaluation of the set candidate's networks.

In DISCOVER use a greedy algorithm that produces a near optimal execution plan with respect to the actual cost. Keyword search enables information DISCOVERY without requiring from the user to know the schema of the database. It proceeds in three steps. 1. It generates the smallest set of candidate networks. 2. Then greedy algorithm creates a non-optimal execution plan to evaluate the set of candidate networks. 3. The execution plan is executed by the DBMS.

DISCOVER uses static optimization. In future, it applies on dynamic optimization. DISCOVER returns a monotonic score aggregation function for ranking a result.

SPARK

With the increasing of the text data stored in relational databases, there are increase a demand for RDBMS to support keyword query search on text data. For the same existing keyword search method can't fulfill the requirement of text data search. This techniques focus on effectiveness and efficiency of keyword query search [16]. They propose a new ranking formula using existing information retrieval techniques. Major importance of this technique is works on large scale real databases (Eg. Commercial application which is Customer Relationship Management) using two popular RDBMS Effectiveness and Efficiency.

It uses a Top-k Join algorithm which includes two efficient query processing algorithms for ranking function. 1. Dealing with Non-monotonic scoring function. 2. Skyline Sweeping Algorithm. This proposed system a new ranking method that adapt the state of art IR ranking function also two algorithm were proposed that aggressively minimize database probes. The result confirms that our ranking method could achieve high precision with high efficiency.

B. Graph Based Approaches

Graph based approaches assume that the database is modeled as a weighted graph where the weight of the edges indicate the importance of relationships. This weighted tree with edges is related to steiner tree problem [5]. Graph base search techniques is more general than schema based techniques including XML, relational databases and internet.[1]

BANKS

A system which enables keyword based searched on relational databases together with data and schema browsing. BANKS enables user to exact information in a simple manner without any knowledge of schema [7]. A user can get information by typing a few keyword, following hyperlinks and interacting with controls on the displayed results. BANKS algorithm is an efficient heuristics algorithm for finding and ranking query results.

BANKS is focus on browsing and keyword searching. Keyword searching in BANKS is done using proximity based ranking on foreign key links. Model database is a graph with the tuple as nodes and cross references between edges. BANKS reduces the efforts involved in publishing relational data on the web and making it searchable.

BLINKS

In query processing over graph-structured is a top-k keyword search query on a graph finds the top k answered according to some ranking criteria. Before the implementation of graph existing system have some drawbacks like poor worst case performance, not taking full advantages of indexes and high memory requirements. To address this problem BLINKS (Bi-level indexing and query processing) scheme for top k-keyword search in graph algorithm will be implemented [4] . To reduce index space BLINKS partition a data graph into blocks. The bi-level index stores summery information at the block level. Main contribution of BLINKS is better search strategy; combining indexing with search and partitioning based indexing. BLINKS algorithm concludes that its major focus on efficiently implementing ranked keyword searches on graph structure data. It is difficult to directly build indexes for general schema-less graph to address this problem introduce BLINKS.

Result shows that BINKS improve the query performance by more than an order of magnitude. In future work of BLINKS includes two aspects for index implementation

1) When graph is updated need to maintain the indexes.
2) By monitoring performance at run time, dynamically change graph partitions and indexes in order to changing data and workloads.

II. COMPARATIVE STUDY

This section includes a study on some algorithm that searches the information from the database but not fulfill all the requirements.

Keyword proximity search in complex data graph

In keyword search over data graph, is a non-redundant sub-tree that includes the given keywords [3]. Algorithms for enumerating answer is presented in architecture have two parts 1.An engine that generate a set of candidate answer. 2. Ranker that evaluate their score. For effectiveness engine must have three fundamental properties: It should not miss relevant answer, must generate the answer in an order that i highly co-related with the desired ranking and it has to efficient.

Keyword search in databases is performed over graph in which nodes are associated with keywords and edges describe semantic relationship. If the databases are an XML document then it can be represented as a graph. The goal is to discover occurrences of the keywords as well as semantic relationship between them. The conclusion of proximity search is the architecture of a generator and a ranker is essential for overcoming the obstacle that arises when applying keyword search to complex data graph. The ranker presented in this search is aimed at eliminating repeated information by incorporating global measure.

B. Steiner-tree based search

A relational database can be modeled as a database graph *G=(V,E).* In this case there is a one to one mapping between a tuple in the database and a node in *V. G* can be consider as a directed graph with two edges: a forward edge (*u*,*v*) *E* if there is a foreign key from *u* to *v*, and a back edge (*u*,*v*)if (*u*,*v*)is a forward edge in *E*. An edge (*u*,*v*)indicate a close relationship between tuples *u* and *v* and the introduction of the two edge allows differentiating the importance of *u* to *v* and vice versa. When such separation is not necessary for some application *G* becomes an undirected graph.

Most existing method of keyword search over relational databases find the steiner tree composed of relevant tuples as the answer [5]. They identify the steiner trees by discovering by the rich structural relationship between tuples, and neglect the fact that such structural relationships can be recomputed and indexed. Existing method identify a single tuple unit to answer keyword queries. To overcome this problem this technique studies how to integrate multiple related tuple units to effectively answer keyword queries. It has implemented method in real database systems, and the experimental results show that this approach achieves high search efficiency and accuracy.

C. Efficient IR-Style Keyword Search over Relational Databases

Keyword search is the dominant information discovery method in documents. Increasing amount of data is stored in databases; lain text coexists with structure data. Till now information discovery in databases required knowledge of scheme, knowledge of a query language and knowledge of the role of the keywords. The goal of this research is enable IR style keyword search over DBMS without the above knowledge [11].

Advantages of IR-Style keyword search are: IR keyword search well developed for document search, modern DBMS offer IT-Style keyword search over individual text attributes. The result keyword query in each edge correspond to each edge corresponds to a primary-foreign key relationship. Some algorithm are used for IR-style keyword search including OR-semantics use DBMS estimator, AND-semantics probabilistically adjust DBMS estimator, if at most a few query results expected then use sparse algorithm, if many query results expected then use global pipelined algorithm.

The conclusion of this method is that: Extend IR-style ranking to databases, Exploit text-search capabilities of modern DBMS to generate result of higher quality, Support both “AND” and “OR “semantics and finally achieve speedup over prior work using pipelined top-k query processing algorithm.

D.Bidirectional Expansion for Keyword Search on Graph Databases

In relational XML and HTML data can be represented as graph with entities as nodes and relationships as edges. Test is associated with nodes and possibly edges. A problem in this system is to efficiently extract from the data graph a small number of the best answer trees. But it can perform poorly if some keywords match many nodes, or some node has very large degree.

In this techniques focus on new search algorithm that is *Bidirectional search* which improves on Backward Expanding search by allowing Forward Search from potential roots toward leaves [12]. Introduce bidirectional search algorithm to handle partial specification of schema and structure using tree pattern with approximate matching. In future improves look ahead techniques for bidirectional search which can reduce the number of nodes touched.

E. QUNITS: queried units for database search

Keyword search against structured databases has become a popular search, many users find structure queries too hard to express and use Google like query box into which search term can be entered. To overcome this problem this research focuses on to create a clear separation between Ranking and Database querying. The first task is to represent the database conceptually as a collection of independent “queried units”, each of which represents the desired result for some query against the database.

In QUNITS based search one high ranking segmentation is used which join between inserted query words [13]. The qunits base approach is a so cleaner approach to model database search. In current model of keyword search in databases, several are applied to construct a result. The benefits of maintaining a clear separation between ranking and database content is that structured information can be consider as one source of information amongst many others. This makes system easier to extend and enhance with additional IR methods for ranking, such as relevance feedback also it allows us to concentrate on making the database more efficient using indices and query optimization.

The conclusion is that IR techniques are not designed to deal with structured inter-linked data and database techniques are not designed to produce result for queries that are under specified. This research has bridge this gap through the concept of QUNITS. They presented an algorithm to evaluate keyword queries against such a database of qunits, based on typing the query. In future work expect to be able to substantially improve upon the qunit finding and utility assignment algorithm.

F. EASE: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-structured and Structured Data

Existing keyword search engines are restricted to a given data model and cannot easily adapt to unstructured, semi-structured and structured data. Current technique proposes an efficient keyword search method called EASE, for indexing and querying large collection of data [18]. To achieve efficiency in processing keyword queries first modeled all data as Graph.

Existing search engine cannot integrate information from multiple interrelated pages to answer to answered keyword queries meaningfully. Conclude that the efficiency of keyword search on structured and semi-structured data is a challenging problem because of the traditional approaches has the inverted index to process a keyword queries which is efficient for unstructured data but inefficient for semi-structured and structured data. For indexing and querying over large collection of unstructured, structured and semi-structure data, and ranking of the result propose a new techniques called EASE with integrating and databases and information retrieval techniques. EASE integrates efficient query evaluation and adaptive scoring for ranking result. EASE provides an efficient algorithm basis for top-k-style processing of large amounts of data for the discovery of rich structural relationship. EASE integrates an effective ranking mechanism to improve search effectiveness.

The contribution of this techniques is model structured, unstructured and semi-structured data as graph and propose an efficient keyword search method. Propose a novel ranking mechanism for effective keyword search also examine the issues of indexing and ranking and devise a simple and efficient indexing mechanism to index the structural relationships between the transformed data.

The conclusion is that EASE is the efficient and adaptive keyword search method to answer keyword queries over structured, unstructured and semi-structured data. The result shows that EASE achieves both high search efficiency and quality for keyword search.

III. *Proposed System:*

I proposed system techniques, empirical performance evaluation of relational keyword search system. The conclusion of literature survey is that many existing search techniques do not provide acceptable performance for realistic retrieval task. Existing search techniques required large memory size for dataset and process. Memory consumption is more in existing techniques. Also another important issue of existing system is that they require more time for query or keyword execution. In summery it conform that the result of existing system is unacceptable performance. Important disadvantages of that includes: 1) Keyword search without ranking. 2) Execution time is more.

In proposed techniques we try to avoid all the disadvantages of existing systems. In our techniques we combine number of algorithms and techniques from data structure and introduce new techniques that can satisfy number of expectation for keyword query search.

Algorithms:

1. *Mining Algorithm*

FPGROWTH Algorithm

The FPGrowth method indexes the database for fast support computation via the use of an augmented prefix tree called the *frequent pattern tree* (FP-tree). Each node in the tree is labeled with a single item, and each child node represents a different item. Each node also stores the support information for the item set comprising the items on the path from the root to that node. The FP-tree is constructed as follows. Initially the tree contains as root the null item. Next, for each tuple ht ,*X*i ∈**D**, where *X* = **i**(t), we insert the item set *X* into the FP-tree, incrementing the count of all nodes along the path that represents *X*. If *X* shares a prefix with some previously inserted transaction, then *X* will follow the same path until the common prefix. For the remaining items in *X*, new nodes are created under the common prefix, with counts initialized to 1. The FP-tree is complete when all transactions have been inserted. The FP-tree can be considered as a prefix compressed representation of **D**. Because we want the tree to be as compact as possible, we want the most frequent items to be at the top of the tree. FPGrowth therefore reorders the items in decreasing order of support, that is, from the initial database, it first computes the support of all single items i ∈I. Next, it discards the infrequent items, and sorts the frequent items by decreasing support. Finally, each tuple ht ,*X*i ∈**D** is inserted into the FP-tree after reordering *X* by increasing item support.

Tidset Intersection approach: Eclat Algorithm

The support counting step can be improved significantly if we can index the database in such a way that it allows fast frequency computations. Notice that in the level-wise approach, to count the support, we have to generate subsets of each transaction and check whether they exist in the prefix tree. This can be expensive because we may end up generating many subsets that do not exist in the prefix tree.

2. *Clustering Algorithm*

In this subsection, we describe the details of our clustering algorithm, the input parameters to our algorithm are the input data set S containing n points in d-dimensional space and the desired number of clusters k. As we mentioned earlier, starting with the individual points as individual clusters, at each step the closest pair of clusters is merged to form a new cluster. The process is repeated until there are only ***k*** remaining clusters.

Efficient search Engine

As we know about the Google search engine it deals with the most useful or most of time user use word. When user searches any key word in Google it replays only the most of time use meaning.

e.g suppose user search “ CAT ” Google give CAT-Exam as a result .but the fact is CAT is also an animal. So this is the main drawback.

So in this techniques we overcome this issues ,we define the category of the search word first then after user will select the appropriate word meaning that he going to search. Then after that we can do the several operations. We will statically add the database in our techniques.

Advantages:

1. Keyword search with ranking.
2. Execution time consumption is less.
3. Less memory require for storing a data and processing.

ACKNOWLEDGMENT

We would like to sincerely thank Mrs.Ruhi Kabra, our mentor (Asst. Professor, GHRCEM, Chas, Ahmednagar), for her support and encouragement.

CONCLUSION

Overall we will study all the existing techniques which is available in market. Each system has some advantages and some issues. We compare all the techniques and checked the performance. So finally conclude that any existing system cannot fulfill all the requirement of keyword query search. They require more space and time; also some techniques are limited for particular dataset.

The Proposed technique is satisfying number of requirement of keyword query search using different algorithms. The performance of keyword search is also the better to compare other and it shows the actual result rather than tentative. It also shows the ranking of keyword and not requires the knowledge of database queries. Compare to existing algorithm it is a fast process.

As a future work we can search the techniques which are useful for all the datasets, means only single technique can be used for MONDIAL, IMDb etc. Further research is necessary to investigate the experimental design decisions that have a significant impact on the evaluation of relational keyword search system.

REFERENCES:

[1] Joel Coffman, Alfred C. Weaver, “An Empirical Performance Evaluation for Relational Keyword Search Systems,” IEEE transaction on Knowledge and Data Engineering, 2014

[2] Vegelis Hristidis, Yannis Papakonstantinou, “DISCOVER : Keyword Search in Relational Database”,28thVLDB Conference, Hong Kong,China,2002

[3]Konstantin Golenberg, Benny Kimelfeld, “Keyword proximity Search in Complex Data Graphs.”, SIGMOD’08,Vancouver,BC,Canada,2008

[4] Hao He, Haixun Wang,”BLINKS: Ranked Keyword Searches On Graph”,SIGMOD’07,Beijing,China,2007

[5] M.L.Shore, L.R.Foulds,”An Algorithm for the Steiner Problem In Graph”,(National Chung-Cheng University,China,2004

[6] Sharmili C.,Rexie J.A.M.,”Efficient Keyword Search Methods in Relational Databases”,IJERA,2013

[7] Gaurav Bhalotia,Arvind Hulgeri,"Keyword Searching and Browsing in Database using BANKS",University of California, Berkeley

[8] E.W.Dijkstra, "Two Problems in Connexion with Graph"1959

[9] Bhavan Dalvi, Megha Kshirsagar,"Keyword Search on External Memory Data Graph",VLDB'08, Auckland, New Zealand,2008

[10] Amit Singhal, "Modern Information Retrieval: a brief overview",IEEE computer Society Technical Committee on Data Engineering,2001

[11] Vagelis Hristidis, Luis Gravano,"Efficient IR-Style Keyword Search Over Relational Databases"

[12] Varun Kocholia, Shashank Pandit,"Bidirectional Expansion For Keyword Search on Graph Databases",University of California,USA,2005

[13] Arnab Nandi,H.V.Jagdish,"Qunits:queried units for database search" ,4th Biennial Conference on Innovative data system research(CIDR) Asilomar,California,USA,2009

[14] Li Qin, Jeffrey Xu.Yu.,"Keyword search in Databases:The Power of RDBMS",SIGMOD'09 Rhode Island,USA,2009

[15] Fang Liu, Clemet Yu,"Effective Keyword Search in Relational Databases ",SIGMOD,Chicago,USA,2006

[16] Yi Luo, Xuemin Lin, "SPARK: Top-k Keyword Query in Relational Databases",SIGMOD'07 ,Chicago,China,2007

[17] Yi Chen, Wei Wang,"Keyword Search On Structured and Semi-Structured Data" ,SIGMOD'09 Rhode Island,USA,2009

[18] Guoling Li, Beng Chin Ooi,"EASE: An Effective 3-in-1Keyword Search Method For Unstructured, Structured and Semi- Structured Data",SIGMOD'08,Vancouver, BC,Canada,2008

[19] William Webber,"Evaluating the Effectiveness of Keyword Search",IEEE Computer Society Technical Committee on Data Engineering,2010