ISSN:0975-9646
Tejashree R. Shinde et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, 4496-4500

An Efficient Schema Free Keyword Search Approach Over Linked Data using Lucene Indexing Algorithm

Ms.Tejashree R. Shinde
Dept. of Computer Engineering
JSPM's BSIOTR, Wagholi - 412207
Pune, India

Prof. Sanchika A. Bajpai
Dept. of Computer Engineering
JSPM's BSIOTR, Wagholi - 412207
Pune, India

Abstract— **Now a days the huge amount of information is maintained, shared and stored on World Wide Web. Every individual need some kind of information which used to be extracted from internet through the various search engines. User expects to get relevant information results according to the query. The information should be relevant as well as valid. To fulfill the user requirements different techniques are used to provide the best and desirable results. A huge amount of research work focusing on the keyword searching, retrieval and query processing has been done in the relational database. In the Web, Linked Data renders data from various sources to be connected and queried. It defines a method of publishing structured and Linked data. Here, the user needs to express requirement in terms of simple keywords. On the basis of given keyword, the problem of finding the relevant sources are defined. Now the main concept of keyword query routing comes which reduces high cost of processing data. In this method with the help of Information Retrieval concept Top-K routing plans are computed using routing graph and which has high frequency count are selected. To overcome the problem of graph expansion the novel indexing method using lucene algorithm is proposed which represents relationship between keywords and data elements. This method reduces the processing time as well as the space required for expansion of graph. With the help of the proposed work the different tradeoffs of existing system are eliminated efficiently and effectively.**

Keywords—- Linked Data, Lucene Indexing Algorithm, Resource Description Framework [RDF], Routing Graph, Routing Plan, Schema-Based Keyword Searching, and Schema-Free Keyword Searching.

I. INTRODUCTION

Querying using keyword is simply the most popular form of query searching. Query searching is widely used to search related and valid documents on the web. Querying of databases is currently based on complex query languages which are not suitable for the nowise user, as they are complex and difficult to understand. To the traditional SQL (Structured Query Language) in querying relational databases with large, most probably the unknown schema and instances, the keyword queries offer an alternative. The challenge in answering such queries is to discover their semantics, construct the SQL queries and explore to retrieve the intended tuples. The discovered structure is the semantic interpretation of keyword query. Existing approaches typically depend on the database content. As the relational data complexity increases the user move towards the less technical skilled approach. The keyword search is popular due to its simplicity and user-friendly nature with the end user who may be less comfortable with the existing techniques. One key problem in web keyword search techniques to databases is that information related to a single answer to a query keyword may be split across multiple tuples in different relation [17]. Numerous studies and techniques have been found in the computer science literature. The existing work consider the database as a network of interconnected tuples, through the network they find the keywords in the query and connected components are derived based on association of tuples and return these connected tuples as an answer to the keyword query. For the purpose specialized indexing techniques are applied over it, which indexes the content of database. Using these indexing techniques, the tuples of interest may directly retrieve or they may instead construct the queries expressions. This is the basic idea followed by the modern commercial database management systems [17]. Linked Data consists of thousands of sources containing billions of RDF triples. It is difficult for the typical web users to exploit this web data by means of structured queries using languages like SQL or SPARQL (Sparkle Protocol and RDF Query Language). To this end, keyword search has proven to be intuitive, as opposed to structured queries, no knowledge of the query language, the schema or the underlying data are needed.

II. HISTORY

The amount of available structured data for ordinary users grows rapidly. Besides data types such as date, digits and time, structured databases probably contain a large amount of text data, such as names of organizations and their products, name of people, titles of books, country, river, songs and movies, street addresses, descriptions of products, contents of papers, and musical lyrics, etc. The need for ordinary user is to find information from text in these databases is dramatically increasing. The traditional search model in relational database requires users to have knowledge of the database schema and to use a structured [6] query language such as SQL or QBE (Query by

Example)-based interfaces. Even though most of the major RDBMS's (Relational Database Management System) have integrated full-text search capabilities with the help of relevance based ranking strategies developed in information retrieval.

Keyword search is the most popular information extraction method because the user does not need to know either a query language or the underlying structure of the data. With the help of web crawling the search engines available provide search results on top of sets of documents. When user enters a set of keywords, the search engine returns all documents that are associated with these keywords. Typically, keywords and a document are interrelated when the keywords are contained in the document and the degree of associativity of each keyword is the distance from both the keywords. Ranked keyword search over tree and graph- structured data has fascinated for two reasons. First, the simple, user- friendly query interface does not require users to have knowledge of complicated query language or understand the underlying data schema. Second, many graph structured data have no obvious, well-structured schema so many query languages are not applicable.

III. Literature Survey

The (RDBMS) relational database management system was first created in the 1970s. Then its popularity has sky touching and it has become a master data storage structure in both academic as well as in commercial fields. Relational databases range from small, individual databases like Excel-Sheet, Microsoft Access to large-scale database servers like Oracle, Microsoft SQL Server, and MySQL. In relational databases, information needed to answer a keyword query is often split across the tables/tuples due to normalization. The work is divided into two directions: A. To compute the most relevant structured results, B. Solutions for source selection compute the most relevant sources.

A. Keyword Search

Generally there are two basic approaches of keyword search classified on the basis of whether searching is based on fixed schema or schema free form;

1) Schema Based Keyword Search Approach

In this approach, keyword query is processed by mapping keywords to elements of the database. With the help of using the schema, verified valid join sequences are derived these joins are then computed keyword to form so called candidate networks representing possible results to the keyword query.

DISCOVER [1] formalizes on relational database. It provides facility of information discovery on the relational database by allowing its user to submit keyword queries without any knowledge of the database schema or of SQL. DISCOVER returns qualified joining networks of tuples which are associated as they join on their primary and foreign keys and contain all the keywords of the query. DISCOVER followed in two steps, first candidate network generation and second is candidate network evaluation.

In [4], text and structured data are often stored side by side within standard RDBMS. Commercial RDBMSs usually provide querying capabilities for text attributes that incorporate state-of-the art IR relevance ranking strategies. This search operation requires that the queries should mention the exact column format. The requirement that queries specify the exact columns to match can be unmanageable and inflexible from a user perspective: good answers to a keyword query might need to be in assembled form. The most important thing to notice is, this approach can handle queries with both AND and OR semantics and exploits the refined single-column text-search functionality often available in commercial RDBMSs.

2) Schema-Free Keyword Search Approach

This approach also called as graph based Graph based search techniques are more general than schema based approaches, for relational databases, XML and the internet are the best example of graph modelling. By understanding the underlying graphs, the structured results are computed. The connected keywords and elements are represented using Steiner trees [18]. The main goal of this approach is to find out structures in the Steiner trees. The algorithm evaluates additional results in approximate order. Various kinds of algorithms have been proposed for the efficient and effective exploration of keyword search results. The general examples are bidirectional search and dynamic programming.

B. Database Selection

The goal of this approach is to identify the most relevant databases. The main idea is based on modeling databases using keyword relationships [18]. A keyword relationship is a pair of keywords that can be connected via a sequence of join operations [17]. A database is relevant if its keyword relationship model covers all pairs of query keywords. M-KS [9] captures relationships using a matrix. It considers only binary relationships between keywords. G-KS addresses this problem by considering more complicated relationships between keywords using a KRG [13]. Each node in the graph corresponds to a keyword. Compared to M-KS, G-KS computes more relevant sources, G-KS uses IR-style ranking to calculate TF-IDF ratio for keywords and their keyword relationships. It provides an additional level of filtering, validating connections between different keywords based on complicated relationships and distance information in the KRG [17].

IV. Implementation Details

A. Introduction

- What is main concern?

A considerable number of search engines give repetitive results of keywords or the time required for retrieval is more or the contained results are from single source. It makes the keyword search system a less efficient and less effective process.

- Why is call for change?

Huge irrelevant, repetitive and complicated keyword search system infringements calls for the development of fast, valid and relevant keyword search system.

- How to retrieve fast result?

Lucene Indexing Algorithm with keyword routing plan, a potential mechanism for fast retrieval, valid and precise result.

B. Mathematical Model

- Let 'S' be the system implementing keyword search mechanism using Lucene Indexing Algorithm with relevant routing plans.
- Let 'I' be the set of inputs, I = $\{R_{db,} R, Q\}$

Elements of 'I' are, Element, Rdb represents set of original keyword (keyword repository),

Rdb = {Rj | R^j is the j^{th} keyword sequence and for all j, $1 \leq j \leq |R_{db}|$}

Where, |Rdb| = Cardinality of keywords in repository,

Where, Rdb = | R_{db} | = | $\{R^1, R^2, R^3, R^{4,}\}$ |

Element 'R' represents number of keywords in RDF form Element,

Element 'Q' represents query keywords by the user Q = {Q_i | Q_i is the i^{th} query keyword sequence and for all i, $1 \leq i \leq |Q|$}

Where |Q| = Cardinality query keyword Q, | Q | = | $\{Q_1, Q_2, Q_3,...\}$ |

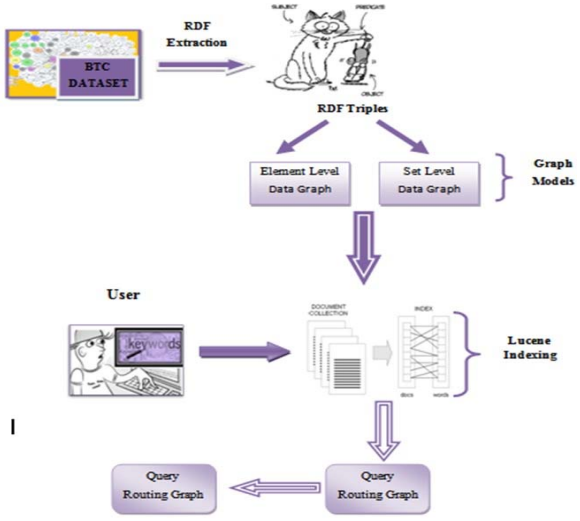
- Let 'O' be the set of outputs, O = $\{R_{plan}, R_{computation}\}$

Elements of 'O' are,

R_{plan} = This represents query keyword routing plans,

$T_{computation}$ = This represents time required for query keyword routing plan.

C. System Architecture



1) Data Extraction And RDF Generation

The data required for keyword searching system is taken from the Billion Triples Challenge 2009 Dataset. The major part of the dataset was crawled during February/March 2009 based on datasets provided by Falcon-S, Sindice, Swoogle, SWSE, and Watson using the Multi Crawler/SWSE framework. To ensure wide coverage, we also included a (bounded) breadth-first crawl of depth starting from http://www.w3.org/People/Berners-Lee/card. The data is encoded in NQuads format and split into chunks of 10m statements each [18]. The combined dataset (gzipped) is around 2.2GB. A smaller crawl useful for testing is available at btc-2009-small.nq.gz. This BTC dataset compromises of multiple sources. All the documents contained in the dataset are Linked Data. This Linked Data is converted into its basic RDF form means in the form of Subject, Predicate and Object with the help of various .jar files.

2) Data Graph Level Models

The graph level models are of two main types one is Element Level Graph Model and other is Set Level Graph Model. These are the basic graph forms of RDF. The element level data graph model is similar to RDF data where entities stand for some RDF resources, data values stand for RDF literals, relations and attributes correspond to RDF triples. In set level data graph, it captures the part of Linked Data schema which is described by RDFs. Here pseudo schema is used as the system uses schema free approach.

3) Lucene Indexing Algorithm

In the previous model graph models are generated on the basis of these graph models entity relationships are find out. These entity relationships are stored by indexing using Lucene indexing algorithm. The lucene indexing algorithm compromises of three main classes, the first is IndexWriter Class, the IndexWriter class takes two parameters, indexDir and config, which are Directory and indexWriter Config objects, respectively. The second is Analyzer class; we have used standard analyzer class for parsing. The third class is adding the object to the index.

4) Keyword Searching Using Lucene Indexing Concept

This is the last module of keyword search system. In this part actual keyword query is submitted by the user. This query is in simple text form the user does not need to know about any schema knowledge or query language. The query is further pre-processed using basic Information Retrieval (IR) concepts like stop word removal, tokenization and porter stemming algorithm. The pre-processed query is directly applied over the indexed lucene algorithm. Further query results are processed by routing graph over index tree and with the help of relevant routing plans are derived. These results are ranked for verification and its validity is further checked by the system.

D. Algorithm Strategy

1) Lucene Indexing Algorithm

Step 1:Accumulate an index,

Step 2:Create an IndexWriter object which is used to form an index and new index entities. It

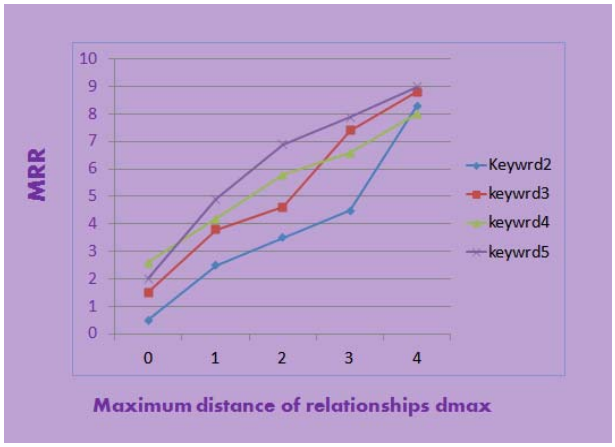
consists of two parameters like indexDir and config.
Step 3:Use the standard analyzer. These are of many types. Anlyzer is used to parse the each field of data.
Step 4:Add the objects to the index.
Step 5:Search the text using two classes QueryParser and IndexSearcher.
Step 6:Stop.

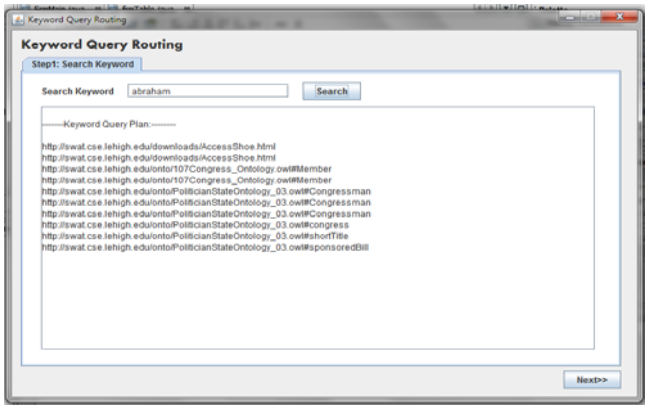
2) Computing Routing Plan Algorithm

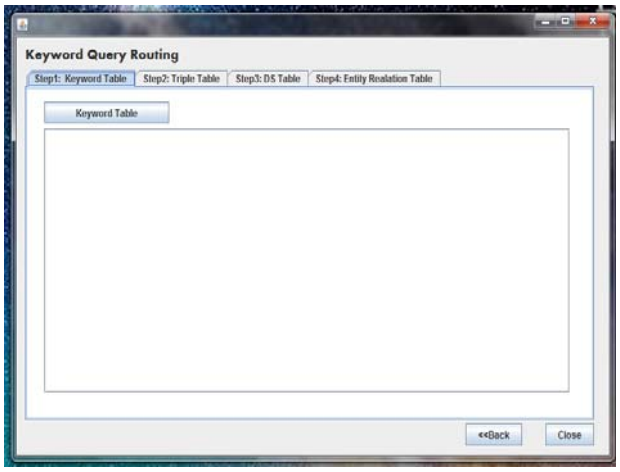
Input: Query and summary model
Output: Set of Routing Plans jp- a join plan that contains all (Ki, Kj) $\epsilon 2^k$
T- A table where every tuples captures a join sequence and combined the score of join sequence, it is initially empty.
Step 1: Start
Step 2: While - jp.empty () do
Step 3: (Ki, Kj) ←jp.pop ();
Step 4: E'(Ki, Kj) ← retrieve (E', (Ki, Kj))
Step 5: If T.empty () then
Step 6: T← E'(Ki, Kj);
Step 7: Else
Step 8: T← E'(Ki, Kj) ∞ T;
Step 9: Stop.

V. Experimental Results

The presented system has proposed the unique solution for keyword query routing by using indexing. We presented an novel indexing technique with lucene algorithm which maintains a stack of segment indices; create indices for each incoming object and push new indexes onto the stack. The lucene indexing algorithm can find among the biggest companies of the world like Comcast, LinkedIn, Twitter, Hi5. The proposed system efficiently carries the keyword search by routing graph to the relevant and valid keywords. Here we have used the routing plan mechanism which identify the valid routing plan result. By this method the quality of keyword query result is maintained . With the help of the proposed system the substantial performance can be achieved.







VI. Conclusion

A unique solution is provided to the problem of keyword query routing in this new approach. Based on multilevel inter-relationship graph concept, a novel indexing method is proposed for a summary model that merges keyword and element relationships at the different set levels and developed a multilevel ranking scheme to incorporate relevance at various dimensions. The experiments showed that the Lucene Indexing Algorithm compactly preserves relevant information. In combination with the proposed ranking, valid plans (precision@1 0:95) that are highly relevant (mean reciprocal rank 0:90) could be computed in 1s on average. Further, we show that when routing is applied to an existing keyword search system weed out the

unwanted sources, significant performance and effective manner which reduces the high cost of searching and within less response time give the valid and precise result.gain can be achieved. Keyword query search is very popular approach for retrieving Linked Data in an efficient

ACKNOWLEDGEMENT

Every work is source which requires support from many people and areas. It gives me proud privilege to publish my sincere work on the respective topic under the valuable guidance of Prof. Sanchika A. Bajpai. I would like to thank my organization for timely help and inspiration and also to all the unseen authors of various articles on the internet, helping me to become aware of the ongoing research in this field and all my colleagues for providing help and support in my work.

REFERENCES

[1] V. Hristidis and Y. Papakonstantinou, "Discover: Keyword earch in Relational Databases", Proc. 28th Int'l Conf. Very Large Data Bases (VLDB), pp. 670-681, 2002.

[2] G. Bhalotia, A. Hulgeri, C. Nakhey, S. Chakrabarti, and S. Sudarshan, "Keyword searching and browsing in databases using BANKS", ICDE , 2002.

[3] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram, "XRANK : Ranked keyword search over XML documents", SIGMOD 2003.

[4] V. Hristidis, L. Gravano, and Y. Papakonstantinou, "Efficient IR-Style Keyword Search over Relational Databases", Proc. 29th Int'l Conf. Very Large Data Bases (VLDB), pp. 850-861, 2003.

[5] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar, "Bidirectional Expansion for Keyword Search on Graph Databases", Proc. 31st Int'l Conf. Very Large Data Bases (VLDB), pp. 505-516, 2005.

[6] F. Liu, C.T. Yu, W. Meng, and A. Chowdhury, "Effective Keyword Search in Relational Databases," Proc. ACM SIGMOD Conf., pp. 563-574, 2006.

[7] B. Ding, J.X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin, "Finding Top-K Min-Cost Connected Trees in Databases", Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 836-845, 2007.

[8] 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.

[9] Y. Luo, X. Lin, W. Wang, and X. Zhou, "Spark: Top-K Keyword Query in Relational Databases," Proc. ACM SIGMOD Conf., pp. 115-126, 2007.

[10] H. He, H. Wang, J. Yang, and P.S. Yu, "Blinks: Ranked Keyword Searches on Graphs," Proc. ACM SIGMOD Conf., pp. 305-316, 2007.

[11] M. Sayyadian, H. LeKhac, A. Doan, and L. Gravano, "Efficient Keyword Search Across Heterogeneous Relational Databases," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), pp. 346-355, 2007.

[12] 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.

[13] G. Li, B.C. Ooi, J. Feng, J. Wang, and L. Zhou, "Ease: An Effective 3-in-1 Keyword Search Method for Unstructured, Semi-Structured and Structured Data," Proc. ACM SIGMOD Conf., pp. 903-914, 2008.

[14] L. Qin, J.X. Yu, and L. Chang, "Keyword Search in Databases: The Power of RDBMS," Proc. ACM SIGMOD Conf., pp. 681-694, 2009.

[15] T. Tran, H. Wang, and P. Haase, "Hermes: Data Web Search on a Pay-as-You-Go Integration Infrastructure," J. Web Semantics, vol. 7, no. 3, pp. 189-203, 2009.

[16] G. Ladwig and T. Tran, "Index Structures and Top-K Join Algorithms for Native Keyword Search Databases," Proc. 20th ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 1505-1514, 2011.

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

[18] "An Empirical Study of Effective and Versatile Keyword Query Search" is published in International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV4IS050573 Vol. 4 Issue 05, May-2015, www.ijert.org