A Novel Scalable Document Retrieval Approach Using Relational Keyword Search System

Neelam Pokale, ME IT Student
Department of Information Technology
SKNCOE
Pune, India
neelam.nsp30@gmail.com

Asst.Prof.Jyoti Yemul
Department of Information Technology
SKNCOE
Pune, India
jyotiyemul@gmail.com

***Abstract*—Keyword search pattern to the relational data is the important and wide area within search and information retrieval community. For the system evaluations, we can follow many methods that proposed but there is a lack of standardization or implementation. The result of lack of standardization degrades performance of the system. The previous search pattern system focus is on memory utilization. The number of queries for the system completed successfully in a query workload is performance wise not showing good results for relational keyword search system. The solution to above problem is to develop a technique that will manage utilization of memory, data swapping to and from hard disk with a help of document retrieval using relational keyword search. The new system will reuse datasets and query workloads to provide higher consistency of results depending on usage of dataset. The new system will explore the relationship between execution time and factors varied in previous are already implemented in contradictory results that are from different evaluations and number of variance confusion is resulted in different evaluations. Scalable document retrieval is useful to get documents with ranking and time consumption is less.The aim of this project is effectiveness in terms of quality to retrieve the most relevant set of documents for a query and efficiency in terms of speed to process queries from users as fast as possible. The average percentage of performance improved by proposed system is 15 to 20 percentage as compared to existing system.**

***Keywords*-Keyword search system, Databases**

I. Introduction

As the keyword search is the popular search that gives search result based on the user entered the keyword. There is an alternative to this system that gives the result when user searches by browsing classification hierarchies. Both searches are valuable and having success ratio as well.

Most amounts of data are present in a relational database. This data should be easily searchable and seamlessly accessible to the end users, allowing users to direct searches in a structured manner. Such search system will be helpful for the users, unlike the documents world there is little support for keyword search over the database that model can be considered extremely powerful in this scenario.

To enable keyword search in databases, that does not require knowledge of the schema. It is a difficult task. There is not direct way to apply techniques from documents world to databases. If there are keywords, a matching row may need to be obtained by joining several tables on the fly. Another thing is that the physical database design needs to be influenced by building compact data structures. In this paper, an efficient and scalable keyword search utility for relational databases is described. The main focus is on query and content based keyword search of documents from a relational database. This approach is helpful to get results in less amount of time.

There are some critical factors for document retrieval like query workload. It is to create own queries or create queries from terms selected randomly. The existing system performance is disappointing to overcome this problem the proposed system is used to get results in less amount of time [1].

The organization of this report is as follows: Section I Introduction.Section II Related work. Section III Proposed Framework. Section IV Implementation Details. Section V Experimental Evaluation and Section VI Conclusion and Future Work.

II. Related Work

Relational keyword search systems focus on three issues related to memory utilization for empirical performance evaluation. The first is developing techniques that are efficient to manage their memory utilization, swapping data to and from disk as required. The second is evaluations should reuse datasets and query workloads to provide greater consistency of results. A third is a researcher facing some evaluation mismatches. It has analyzed graph-based and schema-based systems where graph-based systems contain nodes associated with keywords and edges shows meaningful relationships and schema-based system for direct execution of SQL commands [1] [7].

The second paper shows that no existing system is good for search effectiveness. This paper is on the retrieval effectiveness given by available evaluations and strengthens the need for standardized evaluation. Relational keyword search system has strict splitting [2].

The goal of the third paper is that to use a suitable technique in such a way that the system can get reasonable response time using form based components. There is a time limit for answer generation using keyword search. Form based approach is appropriate to get answers in the particular time limit. It improves quality and performance of the system [3].

The fourth paper shows BANKS system that allows users to extract information in an easy way without schema knowledge or no need of writing complex queries. BANKS is used for large databases where results are direct and useful. It uses relational searches with schema and data browsing [4].

Fifth paper gives an overview of supporting query result definition, ranking functions, result generation, result clustering, top k query processing, etc. This paper described data models like graph - structured data, XML data, data stream, workflows and relational data [5].

Sixth paper explains how to find top k minimum group cost of Steiner trees, denoted GST k for I keyword queries in relational databases. They study detailed on largely directed and undirected graphs and obtained optimal GST 1 with high efficiency and high quality of GST-k [6].

Seventh paper has presented an answer generator that is the first one with all the essential properties of polynomial delay, weightcorrelated order and a no guarantee to miss answers where answer generator is effective. This is a scalable method of removing redundancy [7].

Eighth paper proposed a scalable approach for matchmaking of web services for composition of services of higher granularity from given atomic services. This approach is scalable to store many web services and retrieve a service as per the requirement of matching to a given service. This is implemented in jUDDI open source software [8].

Ninth paper presented a join- based algorithm that processes nodes bottom-up and reduces keyword query evaluation into relational joins. Top K keyword search in XML databases improves efficiency and the top K processing [9].

In tenth paper presented a novel framework for keyword search in relational databases. This will be used in databases on the web, information integration systems, where building and maintaining specialized indexes is not a feasible option [10].

From literature survey, it has been identified that existing system facing problems like severe lack of standardization for system evaluation, not efficient memory utilization and data swapping to and from disk and data redundancy problem as well. To overcome these problems scalable document retrieval using relational keyword search system is used.

The research components of this project are first gathering and recording statistical information about words, features, and documents, Second the index process: building data structures that enable searching, Third the query process: using those data structures to produce a ranked list of documents for a users query and Fourth is calculating index term weights using the document statistics.

III. Proposed Framework

A. System Architecture

The proposed system takes the content and a query as a system input from the user that further can be divided into the content part, and a query part objects. From the user provided input, a query part is extracted, and a content part is passed to the parser.

After applying parser on the content, system will calculate the weight age of the keyword that is passed to SQL generator through matcher. In SQL generator, path selector component sets the path from which server or disk data needs to be retrieved.

SQL Builder will create the query using the user input content that will be passed to the server for searching. Using the metadata extractor on server query is searched, on which two-phase algorithm is applied. After applying the algorithm, output is shown to the user. Figure1 shows an overview of the system architecture.

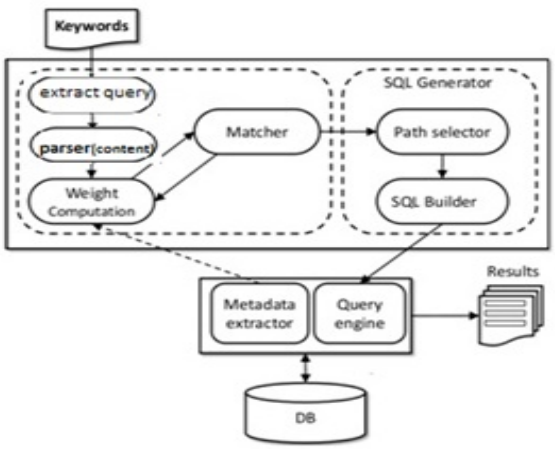

Figure 1. System Architecture

IV. Implementation Details

A. Mathematical Model

- I = $\{i1, i2, ..., im\}$ is a set of items.
- D = $\{T1, T2, ..., Tn\}$ be a transaction database where each transaction Ti, D ϵ I.
- o(ip, Tq), local transaction utility value, represents the quantity of item ip in transaction Tq.
- s(ip), external utility, is the value associated with item ip.
- u(ip, Tq), utility, the quantitative measure of utility for item ip in transaction Tq, is defined as o(ip,Tq) s(ip).
- u(X,Tq), utility of an itemset X in transaction Tq, is defined as Σ u(ip, Tq), where X = i1, i2,..., ik is a k-itemset, X $\subseteq$ Tq and 1 k m.

- $\Sigma u(X)$, represents utility of an itemset X, is Σ u(X,Tq). Tq ϵ DX $\subseteq$ Tq

B. Two-Phase Algorithm

This is a combination of the Iterative Range Selection (IRS) algorithm and the Single Pass Search (SPS) algorithm. This new algorithm is called two-phase algorithm (2PH). In the first phase, we execute a SRS with a tight similarity threshold, T. In the second phase, we run the SPS algorithm, but the initial bound on the number of common grams is computed based on the records retrieved in phase 1.

The algorithm is based on the following two observations. (1) Retrieving the records very similar to the query could be done efficiently using existing range-search algorithms. (2) The SPS algorithm is efficient since it can skip many elements. Still, a low initial frequency threshold makes the algorithm process a lot of elements at the beginning. The initial top-k candidates computed in phase 1 could give as a higher initial frequency threshold. Moreover, the traversal might stop earlier since the records very similar to the query have already been considered.

IRS algorithm is used to answer ranking query by answering range selection queries. Each selection query has a threshold of the similarity between the given string and a string in the collection. The advantage of IRS is easy to implement because it uses iterative range search algorithm. The disadvantage of this algorithm is that multiple search queries need to run so that search time is more.

Single Pass Search algorithm traverses the lists in a sorted order using a heap of the current top elements of the lists.

The advantages of this algorithm are no overhead of maintaining candidate set, and there are many chances of skipping elements.

V. Experimental Evaluation

Newswire dataset is used for searching documents. The Newswire has 20000 records available in dataset. The analysis of keyword search shows that number of documents retrieved from dataset with the help of execution time as follows

Table I
ANALYSIS OF KEYWORD SEARCH IN EXISTING SYSTEM

Result Documents	Execution Time(sec)
50	1.2
85	2.1
140	2.8
298	3.8

The average time required for keyword search is 1.8 seconds in existing system. It is calculated with the help of TABLE I. Figure 2 shows the graphical representation of keyword search in an existing system as follows

The analysis of query and keyword search shown in TABLE II represent that the number of documents retrieved from dataset with the help of execution time is as follows

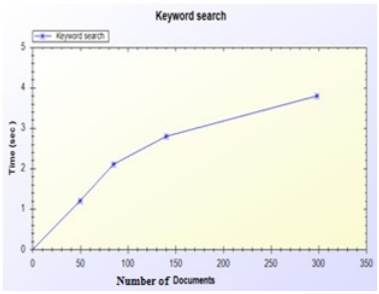

Figure 2. Graph of keyword search from analysis.

Table II
ANALYSIS OF KEYWORD SEARCH IN PROPOSED SYSTEM

Result Documents	Execution Time(sec)
50	0.8
85	1.5
140	2.2
298	2.9

Figure 3 represents a graphical representation of query and keyword search in proposed system is as follows

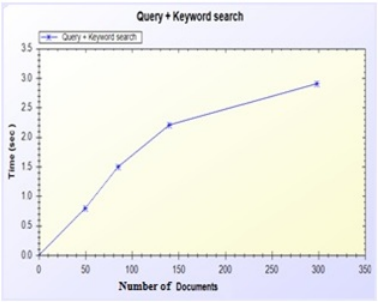

Figure 3. Graph of query and keyword search from analysis.

The average time required searching query and keyword is 1.4 seconds in proposed system. After the comparison of both the approaches query and keyword search takes less time for searching in the document, and that is an efficient method for document retrieval. In query and keyword search exact position is provided for searching that gives an exact result.

VI. Conclusion And Future Work

A Scalable Document Retrieval Scheme for Keyword Search Using Two-Phase Algorithm has an effective index structure and efficient algorithms to support keyword search. It is a fast, scalable method for Document Retrieval Scheme. Two-Phase Algorithm enhances the search speed by reordering documents in the datasets. The proposed system is useful to save time.

In a future, we can use scalable document retrieval using relational keyword search for retrieving images. This paper provides a work that is related to the text data only, so here we also add the concept of image as an input.

ACKNOWLEDGMENT

I am extremely thankful to my project guide Asst. Prof. J. R. Yemul for suggesting the dissertation topic and providing all the assistance needed to complete the work. She helped me greatly to work in this area.

REFERENCES

[1] J. Coffman and A. C. Weaver, *An Empirical Performance Evaluation of Relational Keyword Search Systems*, IEEE Transactions on Knowledge and Data Engineering,Volume: 26,Issue: 1,2014.

[2] A. Baid, I. Rae, J. Li, A. Doan, and J. Naughton, *Toward Scalable Keyword Search over Relational Data*, Proceedings of the VLDB Endowment, vol.3, no.1,2010.

[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan,*Keyword Searching and Browsing in Databases using BANKS*, in Proceedings of the 18th International Conference on Data Engineering,ser. ICDE 02, 2002.

[4] S. Chaudhuri and G. Das, *Keyword Querying and Ranking in Databases*,Proceedings of the VLDB Endowment, vol. 2,2009.

[5] Y. Chen, W. Wang, Z. Liu, and X. Lin, *Keyword Search on Structured and Semi-Structured Data*, in Proceedings of the 35th SIGMOD International Conference on Management of Data, ser. SIGMOD 09,2009.

[6] J. Coffman and A.C. Weaver, *A Framework for Evaluating Database Keyword Search Strategies*, in Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ser. CIKM 10,2010.

[7] B. B. Dalvi, M. Kshirsagar, and S. Sudarshan, *Keyword Search on External Memory Data Graphs*, Proceedings of the VLDB Endowment, vol. 1, no. 1,2008.

[8] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin,*Finding Topk Min-Cost Connected Trees in Databases*, in ICDE 07: Proceedings of the 23rd International Conference on Data Engineering,2007.

[9] K. Golenberg, B. Kimelfeld, and Y. Sagiv, *Keyword Proximity Search in Complex Data Graphs*, in Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD 08,2008.

[10] J. X. Yu, L. Qin, and L. Chang, *Keyword Search in Databases*, 1st ed. Morgan and Claypool Publishers, 2010.