Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 7, Issue, 07, pp.17815-17818, July, 2015

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

A NOVEL DOCUMENT RETRIEVAL SCHEME USING RELATIONAL KEYWORD SEARCH SYSTEM

***Neelam S. Pokale and Jyoti R. Yemul**

Student in Department of Information Technology, Smt. Kashibai Navale College of Engineering, Pune, India

ARTICLE INFO

Article History:
Received 25th April, 2015
Received in revised form
31st May, 2015
Accepted 28th June, 2015
Published online 28th July, 2015

Key words:
Keyword Search System,
Databases.

ABSTRACT

Keyword search pattern to relational data is the most important and highlighted area within search and information retrieval community. For the system evaluations, we can follow many approaches that proposed but there is a lack of standardization. The result of lack of standardization affects performance of the system. The previous system focus is on memory utilization. The number of queries 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 system. Scalable document retrieval improves search performance in terms of execution time, and cost efficiency. The aim of this project is to improve search effectiveness in terms of total execution time, and cost efficiency in terms of data retrieved size. The average percentage of performance improved by proposed system is 10% - 15% as compared to existing system.

***Citation*:** Neelam S. Pokale and Jyoti R. Yemul, 2015. "A novel document retrieval scheme using relational keyword search system", *International Journal of Current Research*, 7, (7), 17815-17818.

INTRODUCTION

Everywhere search text box has changed the way users interact with information. There is a lots of users use a search engine daily for searches. Internet search engine is popularized because it does not require knowledge of schema or query language. It only needs to know or enter contents for search. When user enters content a ranked list of documents returned to the user. Keyword search interface has more demand in the market for information access and therefore it is extended to relational data. An alternative to keyword search is structured search where users direct their search by browsing classification hierarchies. Both models are valuable – success of both keyword search and the classification hierarchy are evident today. 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. In this paper, an efficient and scalable keyword search utility for relational databases is described.

****Corresponding author: Neelam S. Pokale***
Student in Department of Information Technology, Smt. Kashibai Navale, College of Engineering, Pune, India

The main focus is on query and content based keyword search of documents from a relational database. This approach is useful to search performance and cost efficiency of the system. 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 (Coffman and Weaver, 2014). 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.

Related work

This section, presents keyword search techniques in brief as follows:

Relational Keyword Search System

The keyword search paradigm to relational data has been an active area of research within the database and information retrieval (IR) community. A discrepancy exists between the data's physical storage and a logical view of the information. Relational databases are used to eliminate redundancy, and foreign keys searches related information.

Schema Based Systems

This approach supports keyword search over relational databases via direct execution of SQL commands. The schema separates logically connected information, and foreign keys identify related rows. In schema based system search queries cross relationships, the data must be mapped back to a logical view to provide meaningful search results (Coffman and Weaver, 2010). The relation-based approaches aim at processing a keyword query with SQL, use the schema information in RDBMS (Ding *et al*., 2007).

Graph Based Systems

Keyword search in databases is performed over a graph in which nodes are associated with keywords and edges describe semantic relationships (Golenberg *et al*., 2008). We model the database as a directed graph and each tuple in the database as a node in the graph. Each foreign-key–primary key link is modeled as a directed edge between the corresponding tuples (Bhalotia *et al*., 2002). Graph based systems are not schema – aware. Examples of graph based systems are BANKS, BLINKS and DBPF (Baid *et al*., 2010).

Candidate Network Based Systems

Candidate Network is generated with the help of text indices over the data and the users Keywords. Answers to the user's keyword query can be produced by encoding each candidate network. After this candidate networks translated into SQL queries and the respective queries are executed to get result tuples. Candidate Network based system examples are DISCOVER and DBXplorer (Baid *et al*., 2010).

Proposed Framework

System Architecture

The scalable document retrieval system takes content and query based input from the user that again divided into a content part and a query part objects. From this entered input a content part is passed to the parser, and a query part is extracted. After this step, parser is applied on content, so that system calculates total weight-age of the keyword that is passed to SQL generator through matcher.

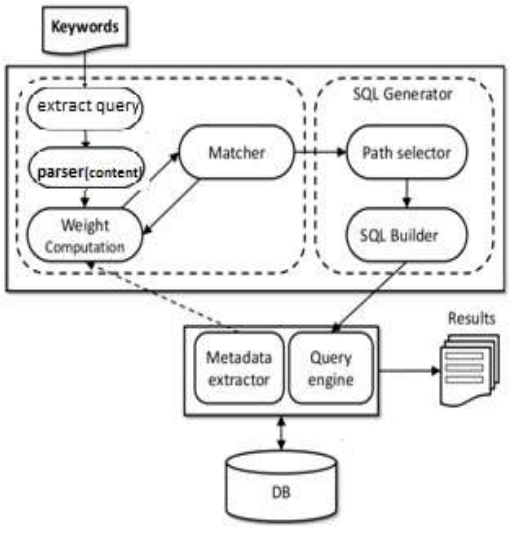

Fig. 1. Scalable Document Retrieval System

Path selector component is used in SQL generator to set path from server, so that server or disk data is retrieved. Query is build by SQL builder with the help of user entered input. This will be passed to the server for searching. Server query is searched with the help of metadata extractor. On which advanced two phase algorithm is applied. While using this algorithm output is displayed to the user. Figure1. shows an overview of the proposed system architecture.

Implementation Details

Mathematical Model

- I = {i1, i2, …, im} is a set of items.
- D = {T1, T2, …, Tn} be a transaction database where each transaction Ti ∈ D is a subset of 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 item set X in transaction Tq, is defined as $\Sigma u(i_p, T_q)$, where
- X = {i1, i2, ik} is a k-item set, X ⊆ Tq and 1≤ k≤ m.
- U(X), represents utility of an item set X, is
- Σ u(X, Tq).
- Tq∈D∧X ⊆Tq

Advanced Two Phase Algorithm

Advanced two phase algorithm is a combination of Iterative Range Selection (IRS) and Single Pass Search (SPS) algorithm. In the first phase, SRS is executed with tight similarity threshold. In the second phase, number of queries are computed depending on the records retrieved in phase1. Advanced two phase algorithm is based on the retrieving records very similar to query efficiently using existing range search algorithm. The SPS algorithm is an efficient, it skips many elements.IRS is used to get ranking queries where as SPS is used to traverse a list in sorted order.

Let k be the number of results requested;
Let wmax be the maximum weight of a string in the dataset;
Let f 1 is a multiplication factor;
Let R be the range-search-result set;
Let be the initial similarity threshold;
Let T be the top element on H;
Insert the top element on each list to a heap, H;
Let p be the number of popped elements;
Pop from H those elements equal to T;

Step 1: Computing initial candidates:
while size(R) < f k do
R ApproxRangeSearch( );
if size(R) < f k
then Decrease ;
end while
Step 2: Finalizing results: Compute scores for elements in R and keep the first k;

Let 1 be the minimum similarity for which Score(1,wmax) > Score(R(k));

if l < then
Topk- while H is not empty do,
if p in R then
if Score(T) > Score(kth in Topk) then
R ApproxRangeSearch(1);
Compute scores for elements in R and keep the first k;
Insert T into Topk and pop the last one;
Recompute threshold;
if R > n then break;
end if
Push next element (if any) of each popped list to H;
Else
Pop additional R p 1 elements from H;
Let T be the current top element on H;
for each of the R 1 popped lists do
Locate its smallest element E T (if any);
Push E to H;
end for
end if
end while
end if
Return R(1..k);

Experimental Evaluation

A scalable document retrieval system uses Newswire dataset and Resume dataset. This dataset contains 20500 thousands of records. The analysis of keyword search system is done with the help of execution time in seconds and data retrieval size in kilobytes. A scalable document retrieval system is designed in such a way that user is able to enter content plus query based input. Due to this input user will get results within a less amount of time. The system is useful to improve search performance and cost efficiency so that execution time and memory space is reduced. This is basically achieving memory and data uses efficiently. The graphical user interface of the system is shown in Figure2 as follows

Fig. 2. Home Page of Scalable Document Retrieval System

Table 1. Analysis of document retrieval system with the help of total execution time

Input	Existing System Output(second)	Proposed System Output(second)
Query1	0.25926	0.231709
Query2	0.235405	0.210387
Query3	0.416809	0.403139
Query4	0.233832	0.213024

The average time required for keyword search is 0.2863265 seconds in existing system and for current system it is 0.26456475seconds. It is calculated with the help of TABLE I. Figure 3 shows graphical representation of existing system and proposed document retrieval system is compared with the help of total execution time in seconds. As shown in Figure3, it proves that the total time required to search a query is less than the existing system. The graph1 represents on X axis numbers of queries searched and on Y axis total execution time as shown below:

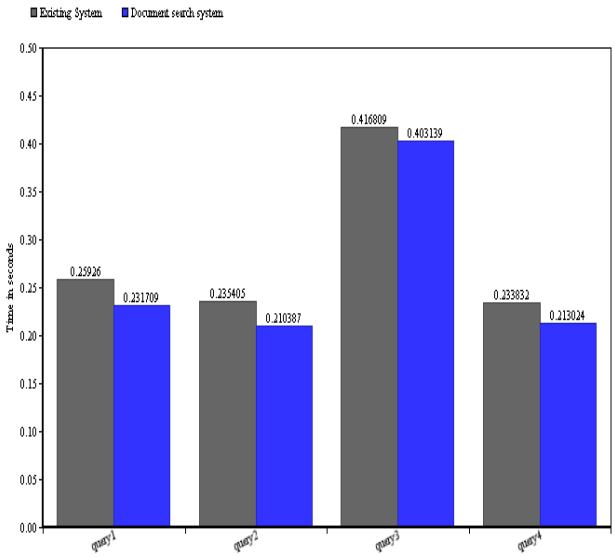

Fig. 3. Comparison of existing and proposed system in terms of Total Execution Time

Table 2. Analysis of document retrieval system with the help of data retrieval size

Input	Existing System Output(kb)	Proposed System Output(kb)
Query1	47	25
Query2	52	26
Query3	80	31
Query4	20	11

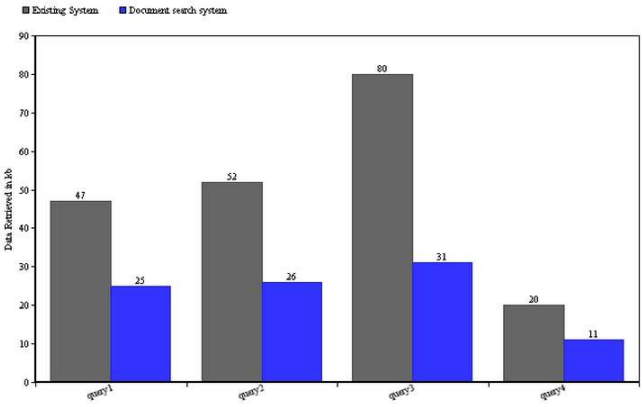

Fig. 4. Comparison of existing system and proposed system in terms of data retrieval size

The average data retrieval size for document retrieval system is 49.75 kilobytes and for current system it is 23.25kilobytes. It is

calculated with the help of TABLE I. Figure 4 shows graphical representation of existing system and proposed document retrieval system is compared with the help of data retrieved size in kilobytes. As shown in Figure 4, it proves that the data retrieved size of the proposed system is less than the existing system. The graph2 clearly shows that cost efficiency of the proposed system is better compared to the existing system. Data retrieved size of an existing system is more than the proposed system.. The graph2 represents on X axis input query and on Y axis data retrieved size as shown below:

Conclusion

From the above discussion on keyword search on relational databases it has been identified that the overall performance of relational keyword search system is somewhat disappointing particularly with regard to the number of queries completed successfully in query workload. Therefore we will describe an efficient and scalable keyword search utility for relational databases. A scalable document retrieval system is designed in such a way that user is able to enter content plus query based input. Due to this input user will get results within a less amount of time. The system is useful to improve search performance and cost efficiency so that execution time and memory space is reduced. This is basically achieving memory and data uses efficiently. The overall efficiency of the system is improved by 10 to 15 % compared to an existing system. In a future, we can use this scheme to retrieve images. This paper supports text based input, we can give add image as a input concept.

Acknowledgment

I am extremely thankful to my Project guide Prof. J. R. Yemul for suggesting the topic for literature survey and providing all the assistance needed to complete the work. She inspired me greatly to work in this area.

REFERENCES

Baid, A. I. Rae, J. Li, A. Doan, and J. Naughton, "Toward Scalable Keyword Search over Relational Data," Proceedings of the VLDB Endowment, vol. 3, 2010, pp. 140–149.

Bergamaschi, S., E. Domnori, R. Emilia, F. Guerra, R. T. Lado, and Y Velegrakis, "Keyword Search over Relational Databases: A Metadata Approach", SIGMOD'11, 2011.

Bhalotia, G. 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, pp. 431–440.

Chen, L. J., Y. Papakonstantinou, "Supporting Top-K Keyword Search in XML Databases", research was supported by NSF IIS award 0713672.

Chen, Y. 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, pp. 1005–1010.

Chenthati, D. H. Mohanty, A. Damodaram, "A Scalable Relational Database Approach for WebService Matchmaking", DOI 10.5013/IJSSST, 2003.

Coffman, J. and A. C. Weaver ,"An Empirical Performance Evaluation of Relational Keyword Search Systems", *IEEE Transactions on Knowledge and Data Engineering*,Vol.26 , 2014.

Coffman, J. 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, pp. 729–738. (Online). Available: http://doi.acm.org/10.1145/1871437.1871531.

Ding, B. 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,pp. 836–845.

Golenberg, K. 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, pp. 927–940.

****