A Novel Document Retrieval Scheme using Relational Keyword Search System

Neelam S. Pokale
Student in Department of Information technology, Smt.Kashibai Navale College of Engineering Pune, India

Jyoti R. Yemul
Assistant Professor in Department of Information Technology, Smt.Kashibai Navale College of Engineering, Pune, India

ABSTRACT
Keyword search pattern to relational data is the most important and the highlighted area within a search and information retrieval community. For the system evaluations, there are many approaches followed but there is a lack of standardization. The result of lack of standardization affects performance of the system. 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 novel technique for efficient document retrieval using relational keyword search system. The new system is developed to manage uploading and downloading of data from disk to improve performance and reuse dataset and query workload to provide greater consistency of results. A scalable document retrieval improves the search performance in terms of execution time, cost efficiency and apply ranking to the document depends on query weight. The new system gives 30% to 40% reduction in the search execution time compared to the existing system.

1. 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. 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 [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.

2. RELATED WORK
This section, presents a keyword search techniques in brief as follows:

2.1 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.

2.2 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 [2]. The relation-based approaches aim at processing a keyword query with SQL, use the schema information in RDBMS [6].

2.3 Graph Based Systems
Keyword search in databases is performed over a graph in which nodes are associated with keywords and edges describe semantic relationships [7]. 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 [4]. Graph based systems are not schema – aware. Examples of graph based systems are BANKS, BLINKS and DBPF [3].

2.4 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 [3].

3. PROPOSED FRAMEWORK

3.1 System Architecture

The scalable document retrieval system takes a content and a 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, the parser is applied on a content, so that system calculates total weight-age of the keyword that is passed to SQL generator through the matcher. Path selector component is used in SQL generator to set path from server, so that the server or a disk data is retrieved.

Query is build by a 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. Fig.1. shows an overview of the proposed system architecture.

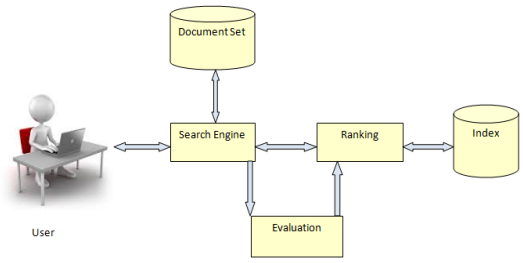

Fig.1: Scalable Document Retrieval System.

3.2 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 $\Sigma u(X, Tq)$. $Tq \in D \wedge X \subseteq Tq$

3.3 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 is 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 1 < 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];

4. 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 ,number of files found 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 performance is improved. This is basically achieving memory and data uses efficiently. The graphical user interface of the system is shown in Fig.2 as follows

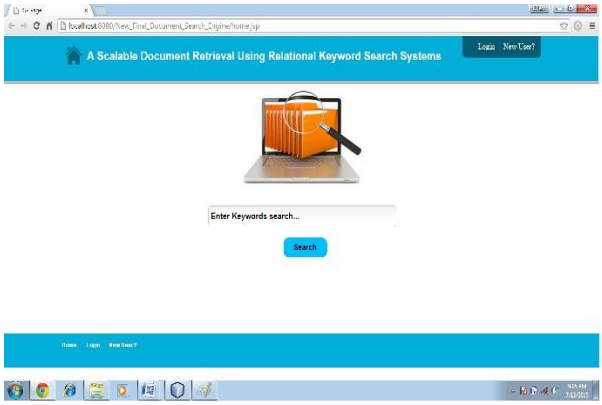

Fig.2: Home Page of Scalable Document Retrieval System

The average time, number of files found and data retrieved size for existing keyword search system is 4.1513125 seconds, 1793.5 and 4936.75 respectively. It is calculated with the help of TABLE 1.

Table1. Analysis of Document Retrieval System for Existing System

Input Keywords	Total Execution Time in Seconds	Number of Files Found	Data Retrieved Size in kb
Memorial university, organization: memorial university	6.999245	2015	6257
Us, organization: us	3.899654	4345	11785
Information, organization:information	3.041919	549	549
bbs, organization : bbs	2.664432	265	1156

The average time, number of files found and data retrieved size for proposed keyword search system is 2.33439225 seconds, 621.25 and 1541.75 respectively. It is calculated with the help of TABLE 2. Fig. 3 shows graphical representation of total execution time required for an existing system and the proposed system.

Table2. Analysis of Document Retrieval System for the Proposed System.

Input Keywords	Total Execution Time in Seconds	Number of Files Found	Data Retrieved Size in kb
Memorial university, organization: memorial university	3.515875	1802	3897
Us, organization: us	2.292187	602	2160
Information, organization : information	1.771045	40	40
bbs, organization : bbs	1.758462	41	70

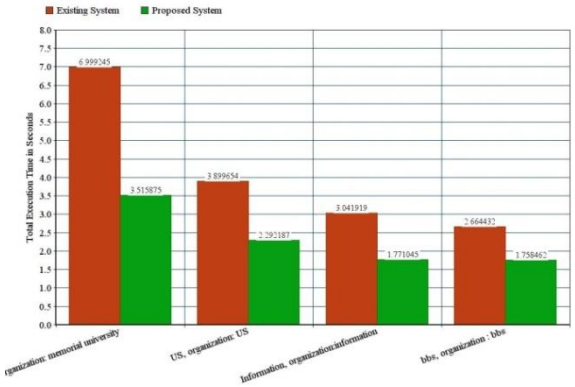

Fig.3: Comparison of existing and proposed system in terms of Total Execution Time.

Fig.4 shows graphical representation of number of files found in an existing and the proposed system. X axis represents specific queries keywords and on Y axis number of files found as shown below:

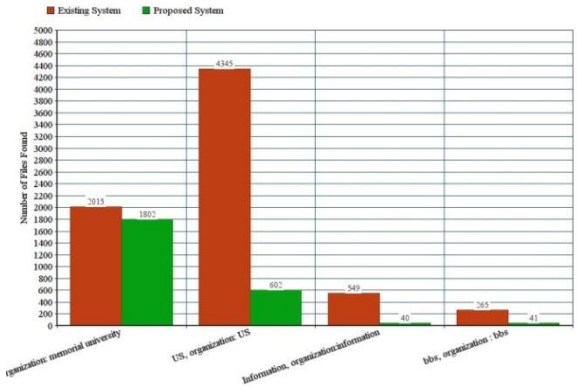

Fig.4: Comparison of existing and proposed system in terms of Number of Files Found.

Fig.5 shows graphical representation of data retrieved size in kilobytes for an existing and proposed system. X axis represents specific queries keywords and on Y axis represents data size in kilobytes as shown below:

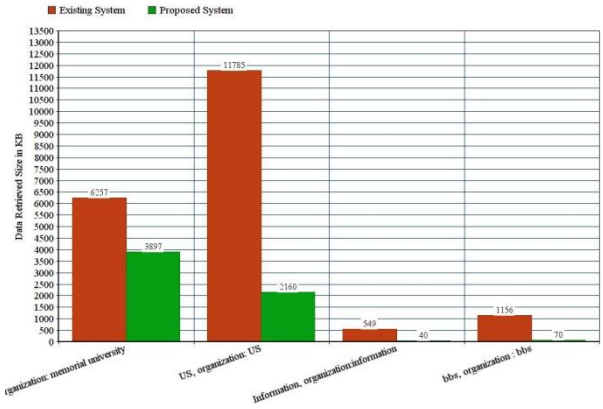

Fig.5: Comparison of existing and proposed system in terms of data retrieved size in kilobytes.

5. 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 reduced. This is basically achieving memory and data uses efficiently. The overall efficiency of the system is improved by 30 to 40 % 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 an input concept.

6. ACKNOWLEDGMENTS

I am extremely thankful to my Project guide Asst. 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.

7. REFERENCES

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

[2] R. Vernica and C. Li, "Efficient Top-k Algorithms for Fuzzy Search in String Collections" , ACM, KEYS'09, June 28, 2009.

[3] 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, pp. 729–738. [Online].Available: http://doi.acm.org/10.1145/1871437.1871531.

[4] 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, 2010 ,pp. 140–149.

[5] 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, pp. 431–440.

[6] 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, pp. 1005–1010.

[7] 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,pp. 836–845.

[8] 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, pp. 927–940.

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

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

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