International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 11 6132 - 6134

Comparative Analysis of Relational Keyword search Systems

Miss Kate Surekha B.
Department of computer engineering of JSCOE
Handewadi Road, Hadapsar,
Pune-411028, India
Surekha.kate@gmail.com

Prof.Ingle Madhav.D.
Department of computer engineering of JSCOE
Handewadi Road, Hadapsar,
Pune-411028, India
ingle.madhav@gmail.com

***Abstract*—** Today with the growth of the Internet, there has been a big growth in the number of users who want to access information without having a detailed knowledge of the query languages; even simple query languages are designed for them that are too complicated for people who dont have sufficient knowledge of language. A large number of methods and prototypes also proposed and implemented, but, there remains a several limitations. So that in this paper, we are overcoming the limitations of previous methods. In literature review indicating that existing systems are using document order so that they are not providing better ranking of keywords. In this paper we are using Top-K based algorithm, ranking function and presenting evaluation of performance of relational keyword search systems. top-k query processing provides highest ranked search results.

Keywords*—keyword search; perfomance evaluation and optimization.*

*****

I. INTRODUCTION

A. Overview

Today all Internet users use a search engine daily, performing billions of searches. But there is a lack of knowledge of the query language or underlying structure of the data. Internet users increasingly demand keyword search interfaces for accessing information, so it is natural to use relational data. In this paper we search query in dataset, calculate execution time for that particular search and explore factors varied in previous evaluations.

B. Motivation

Many search techniques showing that different evaluations of existing systems giving different execution time and ranking. So existing systems not providing optimized results. Our motivation is to help to the user and increase there knowledge about information. We support to user to get instant feedback even typing a partial query and give more choices to user, which helps the user formulate queries. We also retrieving data with high ranking score and minimizing execution time for particular search.

C. Background Need

In this paper we using top-k query processing and ranking function that providing highest rank results. We calculate execution time and rank score for particular search. Here we used dblp dataset in XML format. One important advantage of XML search is it allows users to explore data as they type, even they make minor errors in the keywords. Keyword search on XML data and relational data are different. We doing full text search on database as with multiple keywords as a single string that is typed by user. When user enter keyword, system will search over database and return highest ranked results.

II. LITERATURE **REVIEW**

The search performance is one of major concern in any of search query method presented by different researchers. There are many methods already presented and still in this area continue working is going with aim of improving the search results performances.

In keyword search over data graphs, an answer is a nonredundant subtree that includes the given keywords. An algorithm for enumerating answers is used within an architecture that has two main components: an engine that generates a set of candidate answers and a ranker that evaluates their score. To be effective, the engine must have three fundamental properties. It should not miss relevant answers, has to be efficient and must generate the answers in an order that is highly correlated with the desired ranking [3].

BANKS, a system which enables keyword-based search on relational databases, together with data and schema browsing. BANKS enables users to extract information in a simple manner without any knowledge of the schema or any need for writing complex queries[4]. Even relatively simple query languages designed for non-experts are too complicated for such users. Query languages for semi structured/XML data are even more complex, increasing the impedance mismatch further. Supporting keyword search on structured and semi-structured data, that including query result definition, rank- ing functions, result generation and top-k query processing, result clustering, snippet generation, query cleaning, perfor- mance optimization, and search quality evaluation.

Performance of existing relational keyword search systems is somewhat disappointing, particularly with regard to the number of queries completed successfully in our query workload[1]. In this paper the number of timeout witnessed. Following Table(refer paper[1]) lists the mean execution times of systems from three evaluations that use DBLP and IMDb databases. It indicates that execution time is different of different database for number of evaluations.

	execution time (s)						Evaluation
	DBLP			IMDb			
System	[17]	[13]	[18]	[17]	[13]	[18]	
BANKS [2]	14.8		5.9	5.0		10.6	
BANKS-II [17]	**0.7**	44.7	7.9	**0.6**	5.9	6.6	
BLINKS [13]		**1.2**	19.1		**0.2**	2.8	
STAR [18]			**1.2**			**1.6**	

Yi Chen, Wei Wang, Ziyang Liu, Xuemin Lin [6] given an overview of the state-of-the-art techniques for supporting keyword search on structured and semi-structured data, including query result denition, rank- ing functions, result generation and top-k query processing, snippet generation, result clustering, query cleaning, perfor- mance optimization, and search quality evaluation.

Surajit Chaudhuri and Gautam Das describes two types of challenges that are ranking Challenges and Query Processing Challenges in databases leverage information retrieval, traditional relational query processing, as well as more recent innovations in database algorithms [5].

III. PROBLEM DEFINITION

There are many methods already presented and still in this area continue working is going with aim of improving the search results performances, but there remains a severe lack of standardization for system evaluations. In this paper we search query in dataset and show all result, calculating execution time for that particular Search. We calculate Rank score for particular search and will Store result in Database.

IV. SYSTEM ARCHITECTURE

Many Internet users using keyword search system for performing daily billions of searches. First we search query in dataset and show all results. We calculate execution time for that particular Search, applying Top K algorithm and ranking function. Calculate Rank score for particular search and will Store result in Database. We are using dblp dataset in XML format. Using this dataset as an input. The work is divided into three parts.

A. Keyword Search Method

The aim of the keyword search module is to search for article in dblp.xml file. If keyword is already exist in system then it sent to keyword matching and return the similar documents to the user. If keyword is not existed in system then keyword is extracted from database and compared this with subset of relevant keywords in keyword matching component. Finally, similar documents returned to the user.

The keyword search module is composed following step:

1) Initializing data set.
2) Building of index(pre-processing)
3) Enter single/multiple keyword.
4) Submit query to processor.

B. Approximation Score Calculate

Aim of this module is to calculate approximate score. For query processing we are using top-k scoring function. DISCOVER system calculate the score of rank results.

C. Top-k based Return

The aim of this module is to return top-k based results that are in specific documents. To improve scalability, our system supports top-k algorithm because users wants only the highest ranked search results. Our system using scoring function for the return of approximate top-k based results.

$$\sum_{t \in Q} \frac{1+\ln(1+\ln tf)}{1-s+s\cdot\frac{dl}{avgdl}} \cdot qtf \cdot \ln\left(\frac{N+1}{df}\right)$$

Here Q is query, t is a query term in Q, s is a constant, qtf is the frequency of the query term, N is the number of documents, dl is the document length, avgdl is the mean document length, and df is the number of documents that contain query term t. To obtain the total score, the score of each attribute (i.e., a document) in the tree of tuples is summed.

Hence top-k based return module provides specific documents which are highest ranked.

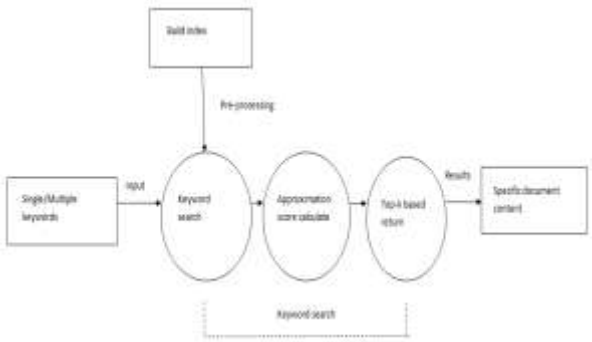

Fig 1 Data flow diagram

When user wants to search a query, he will input a single or multiple keyword to keyword search module. Then system initialize dataset. If keyword is already exists in dataset then it sent to keyword matching component and return the similar documents to the user. If keyword is not existed then keyword is extracted from database and compared this with subset of relevant keywords in keyword matching component. Finally, similar documents returned to the user.

V. PROPOSED SYSTEM

A. We apply fuzzy type-ahead search technique for improve ranking score.
B. To get high ranking score with easily retrieve data and provide optimized results that giving better results than existing methods.
C. Calculate rank score for proposed system.
D. Compare existing and proposed system
E. Show result in graph.

VI. CONCLUSION

Overall, the performance of existing relational keyword search systems is somewhat disappointing, particularly with regard to the number of queries completed successfully in our query workload . Given previously published results shown in table. Our system using top-k algorithm because users typically view only the highest ranked search results that improves scalability. We using scoring function for the return of approximate top-k based results in specific documents.

ACKNOWLWDGEMENT

I am thankful to Prof.M.D Ingle for her expert guidance and continuous encouragement throughout the work.

REFERENCES

[1] Joel Coffman, Alfred C. Weaver (2014). An Empirical Performance Evaluation of Relational Keyword Search Systems. Technical Report, University of Virginia Charlottesville, VA, USA.

[2] Guoliang Li, Shengyue Ji, Chen Li, Jiannan Wang, Jianhua FengEfficient, "Fuzzy Type-Ahead Search in TASTIER" *Tsinghua University, Beijing 100084, China. UC Irvine, CA 92697-3435, USA.*

[3] 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, June 2008, pp. 927–940.

[4] 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, February 2002, pp. 431–440.

[5] S. Chaudhuri and G. Das, "Keyword Querying and Ranking in Databases," Proceedings of the VLDB Endowment, vol. 2, pp. 1658–1659, August 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1687553. 1687622

[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, June 2009, pp. 1005–1010.

[7] H. Bast and I. Weber, "Type less, find more: fast autocompletion search with a succinct index," in *SIGIR*, 2006, pp. 364–371.

[8] S. Ji, G. Li, C. Li, and J. Feng, "Interative fuzzy keyword search," in *WWW 2009*, 2009, pp. 371–380.