Evaluating Performance of Keyword Search Systems

Kaveri A. Dighe[1], M. M. Naoghare[2]

[1]Department of Computer Engineering, Sir Visvesvaraya Institute of Technology, Nasik, Maharashtra, India

[2]Professor, Department of Computer Engineering, Sir Visvesvaraya Institute of Technology Nasik, Maharashtra, India

Abstract: ***In back few years many keyword system have been proposed. But the problem with them is that most of the system are defective or they do not give the exact search results. In this paper we are measuring the performance of all the keyword search systems, doing this will help to choose the correct keyword search system. The analysis of system that already exist will be done. In this paper we will also seek the relationship between time needed for execution and factors changed in previous performances. The analysis shows that previous factors have less influence on performance. The results here indicate that many systems that are existing do not give the satisfactory or needed performance for realistic retrieval tasks. There is need of standardization.***

Keywords: Performance metrics, evaluation, keyword search, Retrieval system, schema-based, semantic performance

1. Introduction

The way of people interacting with information is changing. Nearly half of the internet users use search engine daily, performing about 4 billion searches. Day by day as the searching information is increasing the demand for the keyword search systems is also increasing rapidly. There are many existing relational keyword search systems, but these systems do not acceptable performance. Biad et al. assert that existing systems have performance that cannot be predicted. So the independent evaluation is important.

Here we conduct the independent empirical evaluation of existing relational keyword search systems. In part, existing performance problems may be by experimental design decisions such as the choice of datasets or the construction of query workloads.

Keyword search on data that is not structured for example XML and relational data differs from IR. A inconsistency exists between the data’s physical storage and a logical view of the information. Relational databases are normalized to eliminate redundancy, and foreign keys identify related information. Search queries frequently cross these relationships which forces relational keyword search systems to recover a logical view of the information. The direct assumption of keyword search—that is, the search terms are related— complicates the search process because typically there are many possible relationships between two search terms. It is almost always possible to include other occurrence of a search term by adding new tuples to an existing result.Consider the example of keyword search in relational data. The query “SwitzerlandAustria” where the user to know how the two countries are related. Here the borders relation shows how two countries are related or adjacent. Now Switzerland also borders Germany and also borders France, which borders Germany etc , we can continue to construct the results by adding countries, and considering two relations and few tuples from a much larger databases. Unstructured text allows indexing information at the same as the desired results.

The major contributions are as follows:

- Conduct an independent, empirical performance evaluation of different relational keyword search systems or techniques.
- Existing search techniques performs poorly for large datasets.
- The parameters used in existing evaluation are at best loosely related to performance.
- The motive is to combine the performance and search effectiveness in the evaluation of such a large number of systems.

2. Literature Survey

Baid, I. Rae, J. Li, A. Doan, and J. NaughtonProposed , Keyword search (KWS) systems should return the allanswers they can produce fast and then provide users with options for exploring any portion of the answer space not covered by these answers. The basic idea is to generate answers that can be generated quickly as in today's keyword search systems, then to show users query forms that characterize the unshown portion of the answer space. Bringing together KWS systems with forms allows us to bypass the performance problems inherent to KWS without compromising query coverage. Here providing a proof of concept for this proposed approach, and discuss the challenges encountered in building this hybrid system. Finally, present experiments over real-world datasets to demonstrate the feasibility of the proposed solution.

KWS systems should return whatever answers they can produce fastly and then provide users with options for show any part of the answer space not covered by these answers. The basic idea is to get the answers that can be generated fastly as in today's keyword search systems, then to show users query forms that characterize the portion of the answer space. Combining keyword search systems with forms allows us to detail scan the performance problems inherent to KWS without compromising query.

Gaurav Bhalotia, Arvind Hulgeri, CharutaNakhe, Soumen Chakrabarti S. Sudarsha proposed, BANKS, a system which enables keyword-based search on relational databases, together with data and schema browsing. BANKS enables users to get the information in a easy manner without any knowledge of the schema or any need for writing tough queries. A user can get information by typing a few keywords, following hyperlinks, and interacting with controls on the shown results. BANKS models tuples as nodes in a graph form, connected by links induced by foreign key and other relationships. Answers to a query are modeled as rooted trees connecting tuples that match individual keywords in the query. Answers are ranked using a notion of proximity coupled with a notion of prestige of nodes based on links, similar to methods developed for Web search.

S. Chaudhuri and G. Das, With the uncountable of data sources exposed through web interfaces to consumers, simple ways of exploring contents of such databases are of increasing importance. Examples are the users wishing to search catalogs of homes, cameras, restaurants, and photographs. One method is that has been explored is to allow users to query such databases in the same ways as they explore web documents. Thus, it is suitable to be able to use the pattern of keyword querying and automated result ranking over contents of databases. However, the rich relationships and schema information present in databases makes a direct adaptation of information retrieval techniques inappropriate. This problem has attracted much attention in research as it presents a strong set of challenges from defining semantics of such querying model to build algorithms that ensure adequate performance.

Y. Chen, W. Wang, Z. Liu, and X. Lin, give overview of the state of the art methods for supporting keyword search on structured and semi-structured data, including query results definition, ranking functions, result formation and top-k query processing, snippet generation, result clustering, query cleaning, performance optimization, and search quality evaluation. The data models will be seen, including relational data, XML data, graph-structured data, data streams, and workflows. The description of the applications that are built upon keyword search, such as keyword related database selection, query generation, and analytical processing. Finally, identify the challenges and opportunities of future research to advance the field.

J. Coffman and A. C. Weaver proposed, With regard to keyword search systems for structured data, research during the past few years has largely focused on performance. They illustrate the wide variation in existing evaluations and present an evaluation framework designed to validate the next decennium of research in this field. There comparison of state-of-the-art keyword search systems contradicts the retrieval effectiveness purported by existing evaluations and reinforces the need for standardized evaluation. There results suggest that there remains considerable room for improvement in this area. It was found that many methods cannot even scale to even moderately-sized datasets that contain a million tuples. Given that existing databases are considerably larger than this threshold, results motivate the building of new algorithms and indexing techniques that seek to meet both current and future workloads.

Keyword search over databases has recently received significant attention. Many solutions have been developed. The task requires addressing many issues, including robustness, accuracy, reliability, and privacy. Current keyword search systems do not have predictable running times. In particular, for certain queries it takes too much time to produce answers, and for others the systems may even fail to return answers. The basic idea is to produce answers that can be generated quickly as in today's keyword search systems do.

Current keyword search systems falls into two categories: candidate network based systems and graph based systems. Examples of candidate-network based solutions: DISCOVER and DBXplorer. In this the candidate networks are generated by using an approach similar to BFS. Examples of graph based solutions: BANKS, BLINKS. And the solutions are not schema-aware. Results of keyword are usually modeled as trees that connect nodes that match the keywords.

3. Proposed System

The performance of existing relational keyword search systems is really disappointing, particularly with respectto the number of queries completed successfully in the query workload.

- The objective is to investigate not only the available algorithms but the overall, end-to-end performance of these retrieval systems.
- To underscores the need for Standardization
- To investigate the effectiveness of these retrieval systems.
- The goal is to investigate the scalability of the search techniques.

As shown in the figure 1 below is the block diagram for proposed system.

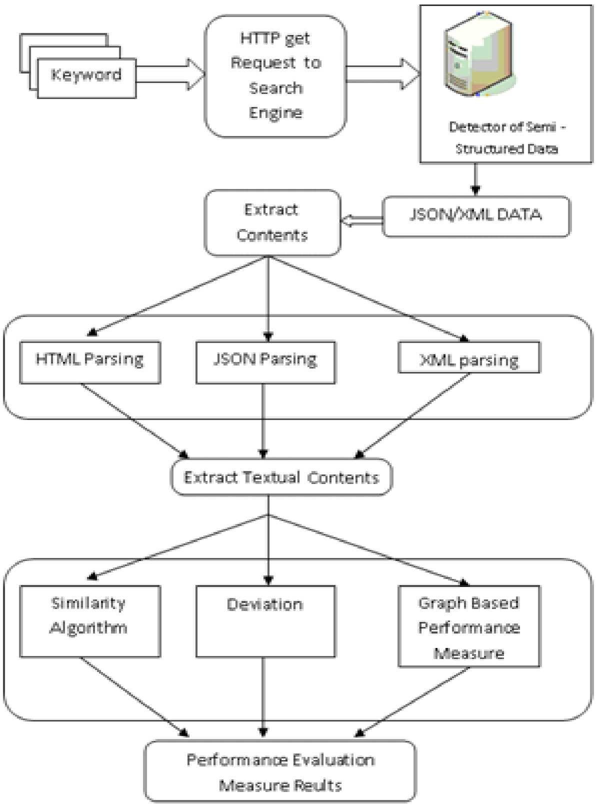
Keyword
HTTP get Request to Search Engine
Detector of Semi - Structured Data
JSON/XML DATA
Extract Contents
HTML Parsing
JSON Parsing
XML parsing
Extract Textual Contents
Similarity Algorithm
Deviation
Graph Based Performance Measure
Performance Evaluation Measure Reults

Figure 1: Block Diagram of Proposed System.

A. Search Engine

User enters keyword after that Http request is send to search engine.(Bing,yahoo etc.)
In response to that HTML response comes from server side in which related results are combined.

B. Detector

Here we implement detector for semi structured data. Like Json/XML.

C. Parsing Process

In this process, we can take structured as well as semi-structured data and perform parsing to separate this both data.

- XML format
- HTML format
- JSON format

D. Similarity Performance Measure

In this module following process will be done:

- Similarity Algorithm
- Cosine Similarity Measure

Cosine similarity is a measure of similarity between two vectors of inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgement of orientation and not magnitude: two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is specially used in positive space, where the outcome is bounded in [0,1].

These bounds apply for any number of dimensions, and Cosine similarity is most commonly used in high-dimensional positive spaces. For example, in Information Retrieval and text mining, each term is notionally assigned a various dimension and a document is characterized by a vector where the value of each dimension is with respect to the number of times that term appears in the document. Cosine similarity helps to give a useful measure of how similar two documents can be likely to be in terms of their subject matter present.

4. Conclusion

Many of the evaluations reported in the survey, This is designed to find not the available algorithms but the overall, performance of these retrieval systems. Hence, here help a realistic query workload instead of a larger workload with queries that are unlikely to be representative. The performance of existing relational keyword search systems is really not satisfactory, particularly with respect to the number of queries completed successfully in the query workload. Here specially surprised by the number of timeout and memory exceptions that we witnessed. Because the larger execution times might only reflect the choice to use larger datasets, the attention on two concerns that we have related to memory utilization.

5. Acknowledgement

It is a great pleasure to acknowledge those who extended their support, and contributed time and psychic energy for the completion of this project work. At the outset, I would like to thank my project guide Prof. M.M.Naoghare,who served as sounding board for both contents and programming work. Her valuable and skillful guidance, assessment and suggestions from time to time improved the quality of work in all respects. I would like to take this opportunity to express my deep sense of gratitude towards her, for her invaluable contribution in this project work. I am also thankful to Prof. S. M.Rokade, Head of Computer Engineering Department for his timely guidance, inspiration and administrative support without which my work would not have been in process.

References

[1] Joel Coffman, Alfred C. Weaver, “An Empirical Performance Evaluation of Relational Keyword Search Systems”, IEEE Transactions on Knowledge and Data Engineering,vol:26,Issue:1) Year: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, pp. 140–149, 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, February 2002, pp. 431–440.
[4] S. Chaudhuri and G. Das, “Keyword Querying and Ranking in Databases,” Proceedings of the VLDB Endowment,vol.2,pp.1658–1659,August2009.[Online].Available:http://dl.acm.org/citation.cfm?id=1687553. 1687622
[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, June 2009, pp. 1005–1010.
[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, October2010,pp. 729–738. [Online]. Available: http://doi.acm.org/10.1145/1871437.1871531
[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, pp. 1189–1204, 2008.
[8] V. Hristidis, L. Gravano, and Y. Papakonstantinou, “Efficient IR-style Keyword Search over Relational Databases,” in Proceedings of the 29th International Conference on Very Large Data Bases, ser. VLDB ’03, September 2003, pp. 850–861.
[9] H. He, H. Wang, J. Yang, and P. S. Yu, “BLINKS: Ranked Keyword Searches on Graphs,” in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’07, June 2007, pp. 305–316.

[10] C. D. Manning, P. Raghavan, and H. Sch¨utze,” Introduction to InformationRetrieval.” New York, NY: Cambridge University Press, 2008.

[11] Tan P-N ,Steinbach M. and Kumar V.,Introduction to Data Mining,Addison Wesley,2006

[12] Soumen Chakrabarti ,Morgan Kaufmann;1 edition (November 26,2008),”DataMining”

[13] “Global Search Market Grows 46 Percent in 2009,” http: //www.comscore.com/PressEvents/Press Releases/2010/Global Search Market Grows 46 Percent in 2009, January 2010.

[14] S. E. Dreyfus and R. A. Wagner, “The Steiner Problem in Graphs,” Networks, vol. 1,no.3,pp.195–207,1971.[Online].Available: http://dx.doi.org/10.1002/net.3230010302.

[15] D. Fallows, “Search Engine Use,” Pew Internet and American Life Project, Tech. Rep., August 2008, http://www.pewinternet.org/Reports/ 2008/Search-Engine-Use.aspx.

Author Profile

Ms. Kaveri A. Dighehas completed her B.E in Computer Engineering from Pune University and currently pursuing Master of Engineering from SVIT Chincholi, Nashik, India.

Prof. M. M. Naogharehas completed her B.E in Computer Engineering from College of Engineering, Badnera, Amravati University and M.E in Computer Science & Engineering from P.R.M.I.T & R, Badnera.