International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 2 Issue: 11 3411 – 3414

Review Paper Title: Research & Evaluation of Keyword Search Techniques over Relational Data

Mrs. Supriya Sanjeevan Mane
Computer Engineering
Bhivrabai Sawant Institute of Technology & Research
Wagholi, Pune-412207
Email: msupriya79@gmail.com

Prof. Mrs.S. A. Bajpai
Computer Engineering
Bhivrabai Sawant Institute of Technology & Research
Wagholi, Pune-412207
Email: sanchi.scriet@gmail.com

Abstract:--Currently the relational keyword based searches techniques consider the large number of data's to provide efficient result while the user searching. There is an issue of limited memory hence there is a need of the implementation of the novel techniques/ algorithm. To improve the search technique process by optimizing the query from that has to contain the memory optimization with the help of the genetic algorithm. The process is executed in the dynamic manner which is considered as the real time scenario in that have to execute the whole process as the dynamic based on the user given query. The proposed system is Research and Evaluation of Keyword Search Techniques over Relational Data. Results indicate that many existing search techniques do not provide acceptable performance for realistic retrieval tasks. Keyword Search with ranking so that our execution time consumption is less, file length and execution time can be seen, ranking can be seen by using chart.

Keywords:-*Relational Keyword Search with ranking, Datasets, Query workloads, Graph-Based System, Execution time.*

*****

I. INTRODUCTION

In the past decade, many efficient and effective techniques for keyword search have been developed. Keyword search is a well-studied problem in the world of text documents and Web search engines. The Informational Retrieval (IR) community has to use the keyword search techniques for searching large-scale unstructured data, and has produced various techniques for ranking query results and to determine their effectiveness. The Database (DB) community has mostly focused on large collections of structured data, and has designed artificial techniques for efficiently processing structured queries over the data. In recent years, emerging applications such as customer support, health care, and XML data management require high demands of processing more mixtures of structured and unstructured data. As a result, the integration of Databases and Information Retrieval (IR) technologies becomes very important. Internet users increasingly demand keyword search interfaces for accessing information, and it is natural to extend this paradigm to relational data (e.g: rows and columns). This extension has been an active area of research throughout the past decade. However, we are not aware of any research projects that have transitioned from proof-of-concept implementations to deployed systems. Keyword search provides great flexibility for analyzing both structured and unstructured data that contain abundant text information. Despite the significant number of research papers being published in this area, existing empirical evaluations ignore or only partially address many important issues related to search performance. Baid.et al. [1] asserts that existing systems have uncertain performance, which undermines their usefulness for real-world retrieval tasks. This claim has little support in the existing literature, but the failure for these systems to gain a foothold implies that robust, independent evaluation is necessary.

II. LITERATURE SURVEY

Existing evaluations of relational keyword search systems are ad hoc with little standards. Baid et al. [1] assert that number of existing keyword search techniques have uncertain performance due to unacceptable response times or fail to produce results even after to empty of contents of memory. Our results particularly the large memory footprint of the systems making sure this claim. A many of relational keyword search systems have been produced beyond those included in our evaluation. Chen et al. [3] and Chaudhuri and Das [2] both presented tutorials on keyword search in databases. Also Chen et al [3] proposed an overview of the state-of-the-art techniques for providing keyword search on structured and semi-structured data, including query result definition, top-*k* query processing ranking functions, and result generation, snippet generation, result clustering, query cleaning, performance optimization, and search quality evaluation. Various data process models will be examine by talk and work, including relational data, XML data, graph-structured data, data streams, and workflows. Webber [10] summarizes existing evaluations with regards to search effectiveness. Although Coffman and Weaver [4] developed the benchmark that we use in this evaluation, their work did not include any performance evaluation. Qin et al. [9] further this efficient query processing by exploring semi-joins.Yu et al. [11] provides an overview of relational

keyword search techniques. Here also focus on using SQL to compile all the interconnected tuple structures for a given keyword query. Also use three types of interconnected tuple structures to gain that and we control the size of the structures. We show that the commercial RDBMSs (Relational Data Base Management System) are powerful enough to provide support such keyword queries in RDBs efficiently without any additional new indexing to be construct and maintained. The main thought behind our approach is tuple reduction. Liu et al. [7] and SPARK [8] both propose modified scoring functions for schema-based keyword search. SPARK also produces a skyline sweep algorithm to minimize the total number of database probes during a search. Dalvi et al. [5] consider keyword search on graphs that cannot fit within main memory. Baid et al. [1] suggest terminating the search after a predetermined period of time and allowing the user to guide further exploration of the search space.

III. RELATIONAL KEYWORD SEARCH

A. Overview of Relational Keyword Search

Keyword search on semi-structured data (e.g., XML) and relational data differs considerably from traditional IR.A contradiction exists in between the data's logical view and physical storage of the information. Relational databases are normalized to eliminate redundancy, and foreign keys identifyrelated information. Search queries frequently cross these relationships (i.e., a subset of search terms is present in one tuple and the remaining terms are found in related tuples), which forces relational keyword search systems to recover a logical view of the information. The implicit assumption of keyword searchis, the search terms are related difficultto the search process because typically there are manypossible relationships between two search terms. It is almost always possible to include occurrence of a search term by adding tuples to an existing result.

Country

Code	Name	Capital
A	Austria	Vienna
CH	Switzerland	Berr
D	Germany	Berlin
F	France	Paris
FL	Liechtenstein	Vaduz
I	Italy	Rome

Borders

C_1	C_2	Length
A	D	784
A	I	430
CH	A	164
CH	D	334
CH	F	573
CH	I	740
F	D	451
FL	A	37
FL	CH	41

This realization leads to tension between the compactness and coverage of search results. Figure 1 provides an example of keyword search in relational data. Consider the query "Switzerland Germany" where the user wants to know how the two countries are related.

Query: "Switzerland Germany"
Results:

1 Switzerland ← [borders] → Germany
2 Switzerland ← [borders] → Austria ← [borders] → Germany
2 Switzerland ← [borders] → France ← [borders] → Germany
4 Switzerland ← [borders] → Italy ← [borders] → Austria ← [borders] → Germany
4 Switzerland ← [borders] → Italy ← [borders] → France ← [borders] → Germany
4 Switzerland ← [borders] → Liechtenstein ← [borders] → Austria ← [borders] → Germany
7 Switzerland ← [borders] → Austria ← [borders] → Italy ← [borders] → France ← [borders] → Germany

Fig1. Example relational data from the MONDIAL database(bottom) and Search results(above). The search results are ranked by size (number of tuples), which accounts for the ties in the list.

The borders relation indicates that the two countries are adjacent. However, Switzerland also borders Austria, which borders Germany; Switzerland borders France, which borders Germany; etc. Asshow in the figure, it can continue to construct results by adding intermediary countries, and we are only considering two relations and a handful of tuples from a much larger database! Implementing coherent search results from discrete tuples is the basic reason that searching relational data is significantly more complex than searching unstructured text. This task is impractical for relational data because an index over logical (or materialized) views is considerably larger than the original data [1], [6].Unstructured text allows indexing data/information at the same granularity as the desired results (e.g., by documents or sections within documents).

B. Contributions and Outline

This paper presents many relational keyword search systems approximate solutions to intractable problems. Researcher'sfrequently rely on empirical evaluation to validate their heuristics. This tradition by evaluating these systems using a benchmark designed for relational keyword search. The view of the retrieval process exposes the real-world tradeoffs made in the design of many of these systems. For example, some systems use alternative semantics to improve performance while others incorporate more sophisticated scoring functions to improve search effectiveness. These tradeoffs have not been the focus of prior evaluations.

The major contributions of this paper are as follows:

- The parameters varied in existing evaluations are at best loosely related to performance, which is likely due to experiments not using representative datasets or query workloads.
- It also conduct an independent, empirical performance evaluation of 7 relational keyword search techniques, which doubles the number of comparisons as previous work.

- This work is the combine first search effectiveness and performance in the evaluation of such a large number of systems.
- Considering search effectiveness and performance, these two issues in conjunction provides better understanding of these two critical tradeoffs among competing system designs.
- The results do not prove previous claims concerning the scalability and performance of relational keyword search techniques.
- Existing search techniques perform poorly for datasets exceeding tens of thousands of vertices.

IV. RELATED WORK

In the system they use the true data sets and the true queries to investigate the numerous tradeoffs in the search techniques. The system which is the only concept to satisfy the least standards recognized by the information retrieval community for the evaluation of the information retrieval systems. They also attain the two processes as the combinations they are the performance and the search effectiveness of the search techniques. In this we present the most extensive empirical performance evaluation of relational keyword search techniques to appear to date in the literature. Our results indicate that many existing search techniques do not provide acceptable performance for realistic retrieval tasks. In particular, memory consumption precludes many search techniques from scaling beyond small data sets with tens of thousands of vertices.

I VARIOUS TECHNIQUES USED IN SYSTEM

1. BANKS

It means Browsing and Keyword Searching. It represents the databases as graphs in which the tuples serve as nodes and the primary-key-foreign-key relationships as edges. It is easy to use and they have features interface for browsing and displaying stored data in the relational database.

2. DISCOVER

It is proceeds in the two main steps they are candidate network generator generates the candidate networks of relations. The second one is plan generator which is used to builds the plans for the efficient evaluation of the set of candidate networks of relations.

3. IR-Style

It means the information retrieval style. It is relevant to the ranking strategies. In the IR-style keyword search the few most relevant matches which is according to the relevance of the keywords.

II MOTIVATION FOR INDEPENDENT EVALUATION

A. Datasets

Some evaluation datasets are here, their content varies dramatically. Consider the evaluations of BANKS-II, BLINKS, and STAR.Only BANKS-II's evaluation includes the entire Digital Bibliography & Library Project (DBLP) and the Internet Movie Database (IMDb) dataset. The Backward expanding strategy used in BANKS [2] can deal with the general model. In brief, it does a best-first search from each node matching a keyword; whenever it finds a node that has been reached from each keyword, it outputs an answer tree. However, backward expanding search may perform poorly with respect to both time and space in case a query keyword matches a very large number of nodes. Consider the evaluations of BLINKS [13], and STAR [18]. Both first BLINKS focus on implementing efficient ranked keyword searches on schema less node-labeled graphs. On a large data graph, many substructures may contain the query keywords. Following the standard approach taken by other systems, we restrict answers to those connected substructures that areminimal and further provide scoring functions that rank answers in decreasing relevance to help users focus on the most interesting answers. And Second STAR use smaller subsets to facilitate comparison with systems that assume the data graph fits entirely within main memory. The literature does not address the representativeness of database subsets, which is a serious threat because the choice of a subset has a profound effect on the experimental results. Most natural semantics while computing near-optimal. Steiner trees with practically viable run-times. The approximation algorithm developalso even outperforms those prior methods that have worked with relaxed semantics.

B. Query Workloads

The query workload is another critical factor in the evaluation of these systems. IR, researchers realized that it was quite hard for users to formulate effective search requests. It was thought that adding synonyms of query words to the query should improve search effectiveness. The trend is for researchers either to maintain and create their own queries or to create queries from terms selected randomly from the corpus. The latter strategy is particularly poor because queries created from randomly-selected terms are unlikely to resemble real user queries. Only two evaluations that use realistic query workloads meet this minimum number of information needs. Not all words related to a query word are meaningful in context of the query. E.g., even though machine is a very good alternative for the word engine, this augmentation is not meaningful if the query is search engine.

V. CONCLUSION

Our results should serve as a challenge to this community because little previous work has acknowledged these challenges. Moving forward, we must address several issues. First, design Data structures, algorithms, and implementations that recognize that storing a complete graph presentation of a database within main memory is infeasible for large graphs. Instead, we should develop techniques that efficiently manage their memory utilization, exchange information/data to and from disk as compulsory. Techniques are improbable to have performance characteristics that are similar to existing systems but must be used if relational keyword search systems are to scale to large datasets (e.g., hundreds of millions of tuples). Second, evaluations should reuse datasets and query workloads to provide greater consistency of results, for even our results vary widely depending on which dataset is considered. Having the community coalesce behind reusable test collections would facilitate better comparison among systems and improve their overall evaluation

First, no system admits to having a large memory requirement. The memory consumption during a search has not been the focus of any previous evaluation. Existing evaluations focus on performance, handling large data graphs (i.e., those that do not fit within main memory) should be well-studied. Relying on virtual memory and paging is no panacea to this problem because the operating system's virtual memory manager will induce much more I/O than algorithms designed for large graphs [6] as evidenced by the number of timeouts when we allowed these systems to page data to disk Also show that storing the graph on disk can also be extremely expensive for algorithms that touch a large number of nodes and edges.

REFERENCES

[1] 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.

[2] 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

[3] 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.

[4] 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, October 2010, pp. 729–738.[Online]. Available:http://doi.acm.org/10.1145/1871437.1871531

[5] 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.

[6] 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.

[7] F. Liu, C. Yu, W. Meng, and A. Chowdhury, "Effective Keyword Search in Relational Databases," in Proceedings of the 2006 ACM SIGMODInternational Conference on Management of Data, ser. SIGMOD '06,June 2006, pp. 563–574.

[8] Y. Luo, X. Lin, W. Wang, and X. Zhou, "SPARK: Top-k Keyword Query in Relational Databases," in Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '07,June 2007, pp. 115–126.

[9] L. Qin, J. X. Yu, and L. Chang, "Keyword Search in Databases:The Power of RDBMS," in Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '09, June 2009, pp. 681–694.

[10] W. Webber, "Evaluating the Effectiveness of Keyword Search," IEEE Data Engineering Bulletin, vol. 33, no. 1, pp. 54–59, 2010.

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