A Novel Approach for Predicting Difficult Keyword Queries over Databases using Effective Ranking

G. Ramakrishnan[1]
Anna University, Computer Science and Engineering
Akshaya College of Engineering and Technology
Coimbatore, India

S. Uma Maheswari [2]
Anna University, Computer Science and Engineering
Akshaya College of Engineering and Technology
Coimbatore, India

Abstract - **Keyword queries on databases provide easy access to data, but frequently put up with the low ranking quality, i.e., low precision and recollect as shown in recent benchmarks. It would be useful to classify queries that are likely to have low ranking quality to improve the user satisfaction. For instance, the system may propose to the user alternative queries for such hard queries. In the existing work, analyses the characteristics of hard queries and propose a novel framework to measure the degree of difficulty for a keyword query in excess of a database, considering both the structure and the content of the database and the query results. However, in this system numbers of issues are there to address. They are time complexity is higher than the other system and reliability rate of the system is lowest. In order to overcome these drawbacks, we are proposing the improved ranking algorithm which is used to enhance the accuracy rate of the system. Our solution is principled, comprehensive, and efficient. This proposed system is well enhancing the reliability rate of the difficult query prediction system. From the experimentation result, we are obtaining the proposed system is well effective than the existing system in terms of accuracy rate, quality of result.**

Keywords— Query performance, query effectiveness, keyword query, robustness, and databases

I. INTRODUCTION

Data mining techniques can yield the benefits of automation on accessible software and hardware platforms, and can be implemented on new systems as existing platforms are upgraded and new crop developed. When data mining tools are implemented on sky-scraping performance parallel processing systems, they can analyse massive databases in minutes. More rapidly processing means that users can routinely experiment with more models to understand difficult data. High speed makes it practical for users to analyse huge quantities of data.

Data mining techniques are the result of a long process of research and artefact growth. This evolution begin when business data was first stored on computers, unremitting with improvements in data right of entry, and more freshly, generated technologies that allow users to find the way through their data in valid time. Data mining takes the right way process away from retrograde data access and direction-finding to potential and proactive in order delivery.

Data mining is organized for application in the business community because it is supported by three technologies that are now sufficiently nature.

- Massive data collection
- Powerful multiprocessor computers
- Data mining algorithms

Commercial databases are growing at exceptional duty. A modern META cluster assessment of data depot projects found that 19% of respondents are beyond the 50 gigabyte level, while 59% expect to be there by second quarter of 1996. The add-on need for improved computational engines can now be met in a cost-effective manner with parallel multiprocessor computer technology. Data mining algorithms exemplify techniques that have existed for at slightest 10 years, but have only a moment ago have been implemented as established, dependable, explicable rigging that constantly surpass older statistical methods.

In the growth from commerce data to company information, each new step has built leading the prior one. For example, active data access is decisive for pierce-through in data routing applications, and the ability to store hefty databases is vital to data mining..

Keyword query interfaces (KQIs) for databases have concerned much concentration in the last decade owing to their plasticity and simplicity of use in incisive and explore the data .Since several article in a data set that contains the query keywords is a budding answer, keyword queries typically have many doable answers. KQIs must spot the information needs last keyword queries and rank the answer so that the preferred answers appear at the top of the list .Unless otherwise noted, we refer to keyword query as query in the remainder of this paper.

Databases contain entities, and entities contain attributes that take attribute values. Some of the difficulties of answering a query are as follows: First dissimilar queries in languages like SQL, users do not habitually stipulate the preferred schema elements for each query term. For instance, query1: Godfather on the IMDB database does not specify if the user is interested in movies whose title is Godfather or movies distributed by the Godfather corporation.KQI necessity find the most wanted attributes associated with each term in the uncertainty. Second, the schema of the harvest is not specified, i.e., users do not give adequate in order to single out exactly their desired entities. For example, Q1 may return movies or actors or producers.

It is important for a KQI to recognize such queries and warn the user or employ alternative techniques like query reformulation or query suggestions. It may also use techniques such as query results diversification. To the best of our knowledge, there has not been any work on predicting or analysing the difficulties of queries in excess of database. Researchers have proposed several methods to detect difficult queries over plain text document collections. However, these techniques are not applicable to our problem since they ignore the structure of the database. In party KQI must consign each query expression to schema elements in the database. It must also discern the desired result types. We empirically show that direct adaptations of these techniques are ineffective for structured data.

I. RELATED WORK

The following are the some of the related work to the paper

- *Clarity-score-based*: The methods based on the concept of *clarity score* assume that users are interested in a very few topics, so they reckon a query easy if its fallout belong to very few topic(s) and therefore, sufficiently distinguish-able from other documents in the assortment. The difficulty of a query more accurately than pre-retrieval based methods for text documents. Some systems measure the distinguish ability of the queries results from the documents in the collection by comparing the probability distribution of terms in the results with the probability distribution of terms in the whole assortment. If these probability distributions are relatively similar, the query results contain information about almost as many topics as the whole collection the query is measured complicated. Several successors propose methods to improve the efficiency and effectiveness of clarity score.

- *Robustness-based*: Another group of post-retrieval methods argue that the results of an easy query are relatively stable against the perturbation of queries, documents or ranking algorithms. Our proposed query difficulty prediction model falls in this category

Some methods use machine learning techniques to learn the properties of difficult queries and predict their hardness. They have similar limitations as the other approaches when applied to structured data. Moreover, their success depends on the amount and quality of their available training data adequate and high quality guidance data is not normally available for many databases. Some researchers propose frameworks that theoretically explain existing predictors and combine them to achieve higher prediction accuracy.

II. EXISTING METHOD

There have been collaborative efforts to provide standard benchmarks and evaluation platforms for keyword search methods over databases. One effort is the data-centric tracks of INEX Workshop Queries were provided by participants of the practicum. Another effort is the series of Semantic exploration challenge .The results indicate that even with ordered data, finding the preferred answers to keyword query is still a hard task.

2.1 Structured Robustness Algorithm

The following are the methods followed in this algorithm**.**

I Noise Generation In Database

To define the noise generation model Fxdb-M for database DB. We will show that each attribute value is corrupted by a combination of three corruption levels: on the value itself, its attribute and its entity set. Now the details: Since the ranking methods for queries over structured data do not generally consider the terms in V that do not belong to query Q, we consider their frequencies to be the same across the original and noisy versions of DB. The corruption model must reflect the challenges about search on structured data, where we showed that it is important to capture the statistical properties of the query keywords in the attribute values, attributes and entity sets. We must introduce content noise (recall that we do not corrupt the attributes or entity sets but only the values of attribute values) to the attributes and entity sets, which will promulgate behind to the attribute values. For instance, if an attribute value of attribute title contains keyword Godfather, then Godfather may appear in any attribute value of attribute title in a corrupted database instance. Similarly, if Godfather appears in an attribute value of entity set movie, then Godfather may appear in any attribute value of entity set movie in a corrupted instance.

II Ranking In Original Database

The mapping probabilities anticipated as describe above, the probabilistic retrieval model for semi structured data PRMS can use them as weights for combining the score from each element into a document score.

$$P\left(\frac{Q}{d}\right) = \prod_{i=1}^{m} \sum_{j=1}^{n} P_{m}\left(E_j \backslash Q_i\right) P_{QL}\left(Q_i | E_j\right)$$

Here, the mapping probability P_M (Ej/w) is calculated and the element-level query-likelihood score P_{QL} (w/ej) is estimated in the same way as in the HLM approach.

$$P_{M(E_j|w)=\frac{P_{M(w|E_j)}P_{M(E_j)}}{P(w)}} = \frac{P_{M(w|E_j)}P_{M(E_j)}}{\sum E_K \in E P_{M(w|E_k)} P_{M(E_k)}}$$

The rationale behind this weighting is that the mapping probability is the result of the inference procedure to decide which element the user may have meant for a given query term. For instance, for the query term `romance', this model assigns higher weight when it is found in genre element as we assume that the user is more likely to have meant a type of movie rather than a word found in plot.

One may imagine a case where the user meant `meg ryan' to be words in the title and `romance' to be in the stratagem. Given that our goal is to make the best guess with the minimal information supplied by user, however, the PRMS will not rank movies that match this interpretation as highly as the more common meaning. Movies that do match this interpretation will, however, appear in the ranking rather than being rejected outright which would be the case if we were generating structured queries.

The experimental results based on collections and a query taken from the actual web services supports the claim that the common interpretation is usually correct.

IV PROPOSED METHOD

The characteristics of difficult queries over databases and propose a novel method to detect such queries. We take gain of the composition of the data to gain insight about the degree of the difficulty of a query given the database. It has implemented some of the most popular and representative algorithms for keyword search on databases and used them to evaluate our techniques on both the INEX and Sem-Search benchmarks. The results show that our method predicts the degree of the difficulty of a query efficiently and effectively.

The problem of predicting the degree of the difficulty for queries over databases. To analyze the reasons that make a query difficult to answer by KQIs. the Structured Robustness (SR) score, which measures the difficulty of a query based on the differences between the rankings of the same query over the original and noisy (corrupted) versions of the similar database, where the blare span on both the substance and the structure of the result entities.

An algorithm to compute the SR score, and parameters to true its performance. The efficient approximate algorithms to estimate the SR score, given that such a gauge is only valuable when it can be computed with a small time overhead compared to the query execution time.

4.1 Static Global Status Approximation

SGS Approx uses the following observation. Once we get the ranked list of top K entities for Q, the corruption module produces corrupted entities and updates the universal statistics of DB. Then, SR Algorithm passes the besmirched results and updated global statistics to the ranking module to compute the corrupted ranking list.

4.2 Query - Specific Attribute Values Only Approximation

QAOApprox corrupts only the attribute values that match at least one query term. Hence, we can significantly decrease the time spent on corruption if we corrupt only the attribute values that contain query terms. We add a check before Line 7 in SR Algorithm to test if A contains any term in Q. Hence, we skip the loop in Line 7. The second and third levels of corruption (on attributes, entity sets, respectively) corrupt a smaller number of attribute values so the time spent on corruption becomes shorter.

V PERFORMANCE EVALUATION

The performance offered Query-specific Attribute values Only Approximation (QAO-Approx) and Static Global Stats Approximation (SGS-Approx). The performance is evaluated by the parameters such as precision, time complexity rate and SR-score. Based on the comparison and the results from the experiment show the proposed approach works better than the existing system.

The proposed system is evaluated in requisites of the time complexity in other words computation time of the query approximation technique such as Query-specific Attribute values Only Approximation (QAO-Approx) and Static Global Stats Approximation (SGS-Approx).

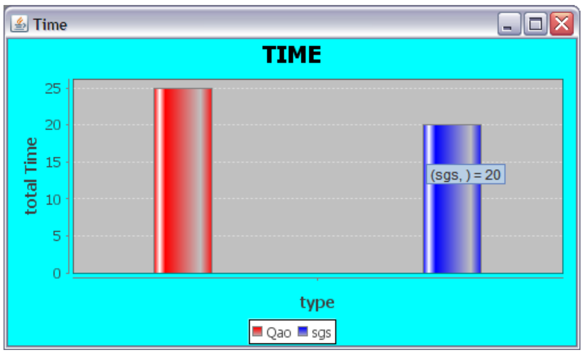

Figure 5.1Time Complexity Of The Query-Specific Attribute Values

Precision value is calculated is based on the retrieval of information at true positive prediction, false positive .In healthcare data precision is calculated the percentage of positive results returned that are relevant.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

In the below graph, we are comparing the precision value of the Query-specific Attribute values Only Approximation (QAO-Approx) and Static Global Stats Approximation (SGS-Approx). In this graph, x axis will be the methods (QAO-Approx and SGS-Approx) and y axis will be precision value. From the graph we can easily understand that the SGS-Approx has higher precision comparatively. So the SGS-Approx has higher precision rate compared to the QAO-Approx.It is shown in the below graph.

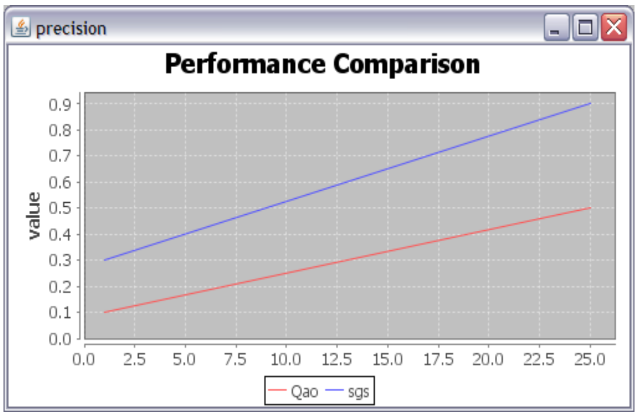

Figure 5.2 comparison graph of the precision value of the Query-specific Attribute values

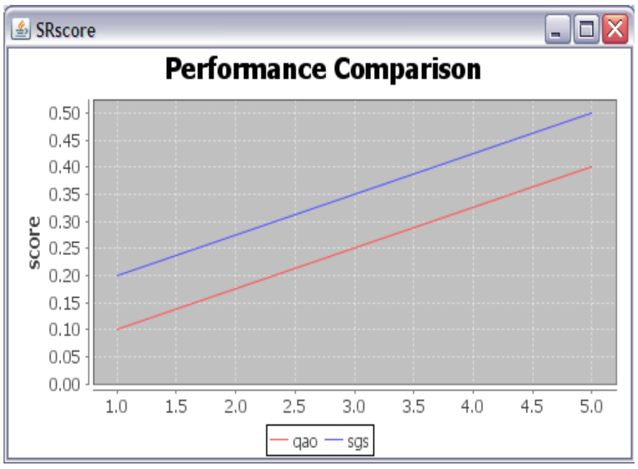

Figure5.3 Higher precision rate SGS-Approx

VI CONCLUSION AND FUTURE WORK

CONCLUSION

The novel problem of predicting the effectiveness of keyword queries over DBs. To showed that the current prediction methods for queries over unstructured data sources cannot be effectively used to solve this trouble. We set forth a honourable structure and proposed novel algorithms to measure the degree of the difficulty of a query more a DB, using the ranking robustness standard. Based on our framework, we propose novel algorithms that efficiently predict the effectiveness of a keyword query. Our wide-ranging experiments confirm that the algorithms predict the difficulty of a query with relatively low errors and negligible time overheads**.**

FUTURE WORK

Better ranking technique is used in our future work in this system. To improve ranking algorithm which are used to enhance the accuracy rate of the system This proposed system is well enhancing the reliability rate of the difficult query prediction system. In this work we propose an automated ranking approach for the Many-Answers Problem for database queries. Our solution is principled, comprehensive, and efficient. We summarize our contributions below. Any ranking function for the Many-Answers Problem has to look beyond the attributes specified in the query, because all answer tuples satisfy. However, investigating unspecified attributes is particularly tricky since we need to determine what the user's preferences for these unspecified attributes are. In this work we propose that the ranking function of a tuple depends on two factors: (a) a global score which captures the global importance of unspecified attribute values, and (b) a conditional score which captures the strengths of dependencies (or correlations) between specified and unspecified attribute values.

ACKNOWLEDGEMENT

The success of a work depends on the team and its cooperation. I take this opportunity to express my gratitude and sincere thanks to everyone who helped me in my project. First and foremost, I would like to thank the Management for the Excellent Infrastructure, facilities and the constant support provided for the successful completion of the project work.

I thank Associate Professor Mr P. Dhamodharan M.E., (Ph.D), Head of the Department, Computer Science and Engineering, for his valuable guidance, continuous support and suggestions to improve the quality of the project work. I express my special thanks to my guide, Assistant Professor, Mrs.S.Umamaheswari M.E., for her valuable guidance, insightful comments and continuous support to carry out the project work.

I express my deep sense of gratitude to all the Faculty members and supporting staff for their continuous support in completing this project work successfully. My sincere thanks to my family members and friends for their support and motivation to carry out this project successfully.

REFERENCES

1. V. Hristidis, L. Gravano, and Y. Papakonstantinou,(2003) "Efficient Irstyle keyword search over relational databases,"
2. Y. Luo, X. Lin, W. Wang, and X. Zhou,(2007) "SPARK: Top-k keyword query in relational databases," .
3. V. Ganti, Y. He, and D. Xin,(2010) "Keyword++: A framework to improve keyword search over entity databases," .
4. J. Kim, X. Xue, and B. Croft,(2009) "A probabilistic retrieval model for semistructured data," .
5. N. Sarkas, S. Paparizos, and P. Tsaparas,(2010) "Structured annotations of web queries," .
6. G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan,(2002) "Keyword searching and browsing in databases using BANKS,".
7. C. Manning, P. Raghavan, and H. Schütze,(2008) An Introduction to Information Retrieval.
8. A. Trotman and Q. Wang, "Overview of the INEX(2010)data centric track," in 9th Int. Workshop INEX (2010).
9. T. Tran, P. Mika, H. Wang, and M. Grobelnik,(2010) "Semsearch ´S10,".
10. S. C. Townsend, Y. Zhou, and B. Croft,(2002) "Predicting query performance,"
11. A. Nandi and H. V. Jagadish,(2007) "Assisted querying using instantresponse interfaces,".
12. E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl,(2010) "DivQ: Diversification for keyword search over structured databases,"
13. Y. Zhou and B. Croft, "Ranking robustness: A novel framework to predict query performance," in Proc. 15th ACM Int. CIKM, Geneva, Switzerland, 2006, pp. 567–574.

AUTHORS PROFILE

G.Ramakrishanan received the B.E. degree in Computer Science and Engineering from the Coimbatore institute of engineering and technology, Coimbatore, Anna University, Coimbatore, India, in 2011.Currently doing M.E. in Computer Science and Engineering in Akshaya college of engineering & technology,Kinathukadavu, Coimbatore, India.

Mrs.S.Umamaheswari received the M.E. degree in Computer Science and Engineering from Coimbatore Institute of Engineering and Technology and has 3 years of experience in Teaching. Her research interest includes Datamining, Networking and Cloud Computing, Image Processing.