ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Novel Framework to Measure the Degree of Difficulty for a Keyword Query Over a Database

N.Srinivas
M.Tech Student,
Department of CSE,
Global Institute of Engineering & Technology, Chilkur, RR District, Telangana.

Mr.Syed.Mazharuddin
Associate professor,
Department of CSE,
Global Institute of Engineering & Technology, Chilkur, RR District, Telangana.

Mrs. M.Jhansi Lakshmi
Associate professor & HOD,
Department of CSE,
Global Institute of Engineering & Technology, Chilkur, RR District, Telangana.

ABSTRACT:

Users query in search engines using keywords and it's a widely used form of querying. In response to a user need, or query, the task of an Information Retrieval system is to retrieve useful information from a large repository of data. Search engines usually do keyword matching only. As long as the query word is present in the document, it is fetched and presented to the user as the result of the query. If a user query is general and ambiguous, the query engine finds the query as a complex query as it cannot extract the exact result which the user expects. Search engines do not identify the needs behind the query hence most of the relevant documents are not retrieved.

The difficulty of this task will be affected by various factors, relating to the system or algorithms used, to the properties of the data to be retrieved, or to the inherent difficulty of the user's information need. The effect of these factors upon retrieval performance is often referred to as query difficulty or complexity and the query is said to be a complex query. The aim of this paper is to analyze and study about the efficient query retrieval and prediction over the databases. In this approach, the Structured Robustness (SR) score algorithm is implemented to predict the effectiveness of the keyword queries over databases. Query reformulation method is used where the search engine helps the user to reformulate their query by adding more specific keywords to it. The expected outcome effectively predicts the hard queries on structured data over databases. The ranking quality of the results provides a good user satisfaction.

Index Terms:

Query performance, query effectiveness, keyword query, robustness, databases.

INTRODUCTİON:

Data mining is the process of extracting useful information from large volumes of data. It is also known as Knowledge Discovery in Databases (KDD) and it is used for data extraction from various databases such as relational databases, object-oriented databases, data warehouses, transactional databases etc. With the help of data mining tools we can predict behavior and future trends and make knowledge-driven decisions. With the growth of the internet and enormous data, it is difficult for the user to get relevant documents. Information Retrieval (IR) system retrieves information from a large repository of data in response to a user query. Given a set of documents in the database and a query given by user, subset of documents relevant to the query is to be retrieved in any Information Retrieval system. In Relational databases, the data is commonly searched using Structured Query Language (SQL). Information needed to answer a keyword query is often split across the tables/tuples. If the user knows the schema of the database he can form suitable query for his needs.

The Mean Average Precision (MAP) of the best perform in method(s) in the last data-centric track in INEX Workshop and Semantic Search Challenge for queries are about 0.36 and 0.2, respectively. These results indicate that even with structured data, finding the desired answers to keyword queries is still a hard task. More interestingly, looking closer to the ranking quality of the best performing methods on both workshops, we notice that they all have been performing very poorly on a subset of queries. For instance, consider the query ancient Rome era over the IMDB data set. Users would like to see information about movies that talk about ancient Rome. For this query, the state-of the- art XML search methods which we implemented return rankings of considerably lower quality than their average ranking quality over all queries.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Hence, some queries are more difficult than others. Moreover, no matter which ranking method is used, we cannot deliver a reasonable ranking for these queries. Table 1 lists a sample of such hard queries from the two benchmarks. Such a trend has been also observed for keyword queries over text document collections .In case of online databases, user usually does not have detailed knowledge of schema or query languages. Hence the desired results are not obtained. Keyword Query Interfaces (KQI) is the type of computer-human interface used for selecting the required data. Keyword queries on databases are used to provide easy access for the data to be searched. But, these data have low ranking quality in real world scenario. In response to a user‟s query the Search engines usually do keyword matching and return ranked list of all the documents containing the keywords specified in the query. The relevant documents may not be retrieved and/or retrieved instances may not be relevant (i.e. low precision and/or recall). Keyword search provides an alternative and easy way of querying in relational databases. One important advantage of keyword search is users need not have prior knowledge about the structures of the underlying data or the knowledge of complex structured query languages (e.g., SQL) for querying their information needs.

In the event that the query is general, it is hard to recognize the specific record on which the user is interested. The users are required to filter through a not insignificant rundown of off-subject reports. The queries which are hard to answer effectively by the Information Retrieval (IR) system are called hard queries or complex queries. Thus, the query engine should identify and search the desired attributes associated with each term given in the query. Thus, it is important to distinguish such queries that are liable to have low positioning quality with a specific end goal to enhance the user fulfillment level. The topic of how to find data of interest of user in the World Wide Web is raised by the Web Search Problem. Majority of queries requested to the Internet search engines by the users are general, few words in length. Low quality searches are many and therefore methods of dealing with the results of such queries are needed. One method is filtering of the ranked list of documents, varying from simple pruning techniques to advanced Artificial Intelligence algorithms. Although they limit the total length of the ranked list, it is difficult for users to locate the specific documents they searched for. The search engines can also help the users to refine their query by adding more specific keywords to it.

Query engine must assign each query words to schema elements in the database. In this regard, this paper we study the properties of complex queries over databases and proposes a method to detect such queries. The structure of the data in the given database is used to study about the degree of the complexity of the query. The method used predicts the degree of the complexity of a query efficiently. Structured Robustness (SR) score, 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 same database. Finally, thresholding approach is used to define query difficulty metric.

LITERATURE SURVEY:

1 “Efficient IRstyle keyword search over relational databases,”

Applications in which plain text coexists with structured data are pervasive. Commercial relational database management systems (RDBMSs) generally provide querying capabilities for text attributes that incorporate state-of-the-art information retrieval (IR) relevance ranking strategies, but this search functionality requires that queries specify the exact column or columns against which a given list of keywords is to be matched. This requirement can be cumbersome and inflexible from a user perspective: good answers to a keyword query might need to be “assembled” -in perhaps unforeseen ways- by joining tuples from multiple relations.

This observation has motivated recent research on free-form keyword search over RDBMSs. In this paper, we adapt IR-style document-relevance ranking strategies to the problem of processing free-form keyword queries over RDBMSs. Our query model can handle queries with both AND and OR semantics, and exploits the sophisticated single-column text-search functionality often available in commercial RDBMSs. We develop query-processing strategies that build on a crucial characteristic of IR-style keyword search: only the few most relevant matches -according to some definition of “relevance”- are generally of interest.

Consequently, rather than computing all matches for a keyword query, which leads to inefficient executions, our techniques focus on the top-k matches for the query, for moderate values of k. A thorough experimental evaluation over real data shows the performance advantages of our approach.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

2. "SPARK: Top-k keyword query in relational databases,"

With the increasing amount of text data stored in relational databases, there is a demand for RDBMS to support keyword queries over text data. As a search result is often assembled from multiple relational tables, traditional IR-style ranking and query evaluation methods cannot be applied directly.In this paper, we study the effectiveness and the efficiency issues of answering top-k keyword query in relational database systems. We propose a new ranking formula by adapting existing IR techniques based on a natural notion of virtual document. Compared with previous approaches, our new ranking method is simple yet effective, and agrees with human perceptions. We also study efficient query processing methods for the new ranking method, and propose algorithms that have minimal accesses to the database. We have conducted extensive experiments on large-scale real databases using two popular RDBMSs. The experimental results demonstrate significant improvement to the alternative approaches in terms of retrieval effectiveness and efficiency.

3. ''A framework to improve keyword search over entity databases,"

Keyword search over entity databases (e.g., product, movie databases) is an important problem. Current techniques for keyword search on databases may often return incomplete and imprecise results. On the one hand, they either require that relevant entities contain all (or most) of the query keywords, or that relevant entities and the query keywords occur together in several documents from a known collection. Neither of these requirements may be satisfied for a number of user queries. Hence results for such queries are likely to be incomplete in that highly relevant entities may not be returned. On the other hand, although some returned entities contain all (or most) of the query keywords, the intention of the keywords in the query could be different from that in the entities. Therefore, the results could also be imprecise. To remedy this problem, in this paper, we propose a general framework that can improve an existing search interface by translating a keyword query to a structured query. Specifically, we leverage the keyword to attribute value associations discovered in the results returned by the original search interface. We show empirically that the translated structured queries alleviate the above problems.

4. "A probabilistic retrieval model for semi-structured data,"

Retrieving semistructured (XML) data typically requires either a structured query such as XPath, or a keyword query that does not take structure into account. In this paper, we infer structural information automatically from keyword queries and incorporate this into a retrieval model. More specifically, we propose the concept of a mapping probability, which maps each query word into a related field (or XML element). This mapping probability is used as a weight to combine the language models estimated from each field. Experiments on two test collections show that our retrieval model based on mapping probabilities outperforms baseline techniques significantly.

5. "Structured annotations of web queries,"

Queries asked on web search engines often target structured data, such as commercial products, movie show times, or airline schedules. However, surfacing relevant results from such data is a highly challenging problem, due to the unstructured language of the web queries, and the imposing scalability and speed requirements of web search. In this paper, we discover latent structured semantics in web queries and produce Structured Annotations for them. We consider an annotation as a mapping of a query to a table of structured data and attributes of this table. Given a collection of structured tables, we present a fast and scalable tagging mechanism for obtaining all possible annotations of a query over these tables. However, we observe that for a given query only few are sensible for the user needs. We thus propose a principled probabilistic scoring mechanism, using a generative model, for assessing the likelihood of a structured annotation, and we define a dynamic threshold for filtering out misinterpreted query annotations. Our techniques are completely unsupervised, obviating the need for costly manual labeling effort. We evaluated our techniques using real world queries and data and present promising experimental results.

EXISTING SYSTEM:

Researchers have proposed methods to predict hard queries over unstructured text documents. This project can broadly categorize these methods into two groups: pre-retrieval and post-retrieval methods.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Pre-retrieval methods predict the difficulty of a query without computing its results. These methods usually use the statistical properties of the terms in the query to measure specificity, ambiguity, or term-relatedness of the query to predict its difficulty. Examples of these statistical characteristics are average inverse document frequency of the query terms or the number of documents that contain at least one query term. These methods generally assume that the more discriminative the query terms are, the easier the query will be. Empirical studies indicate that these methods have limited prediction accuracies. Post-retrieval methods utilize the results of a query to predict its difficulty and generally fall into one of the following categories. Clarity-score-based: The methods based on the concept of clarity score assume that users are interested in a very few topics, so they deem a query easy if its results belong to very few topic(s) and therefore, sufficiently distinguishable from other documents in the collection. Researchers have shown that this approach predicts the difficulty of a query more accurately than pre-retrieval based methods for text documents. Some systems measure the distinguishability 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 collection.

PROPOSED SYSTEM:

In this project, this project analyze the characteristics of difficult queries over databases and propose a novel method to detect such queries. This project take advantage of the structure of the data to gain insight about the degree of the difficulty of a query given the database. This project introduce the problem of predicting the degree of the difficulty for queries over databases. This project also analyze the reasons that make a query difficult to answer by KQIs. This project propose 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 same database, where the noise spans on both the content and the structure of the result entities. This project present an algorithm to compute the SR score, and parameters to tune its performance.

Data and Query Modeling:

* In this module, first we develop a System Model for our proposed System.

We model a database as a set of entity sets. Each entity set S is a collection of entities E. For instance, movies and people are two entity sets in IMDB.

* We ignore the physical representation of data in this paper. That is, an entity could be stored in an XML file or a set of normalized relational tables. The above model has been widely used in works on entity search and data-centric XML retrieval [8], and has the advantage that it can be easily mapped to both XML and relational data.

Ranking for Structured Data:

* In this module we present the Ranking Robustness Principle, which argues that there is a (negative) correlation between the difficulty of a query and its ranking robustness in the presence of noise in the data.
* The degree of the difficulty of a query is positively correlated with the robustness of its ranking over the original and the corrupted versions of the collection. We call this observation the Ranking Robustness Principle.

Corruption Module:

* The first challenge in using the Ranking Robustness Principle for databases is to define data corruption for structured data. For that, we model a database DB using a generative probabilistic model based on its building blocks, which are terms, attribute values, attributes, and entity sets.
* A corrupted version of DB can be seen as a random sample of such a probabilistic model.

Ranking Module:

* Each ranking algorithm uses some statistics about query terms or attributes values over the whole content of DB. Some examples of such statistics are the number of occurrences of a query term in all attributes values of the DB or total number of attribute values in each attribute and entity set. These global statistics are stored in M (metadata) and I (inverted indexes) in the SR Algorithm pseudocode.
* SR Algorithm generates the noise in the DB on-the-fly during query processing. Since it corrupts only the top K entities, which are anyways returned by the ranking module, it does not perform any extra I/O access to the DB, except to lookup some statistics. Moreover, it uses the information which is already computed and stored in inverted indexes and does not require any extra index.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

RESULTS:

The database used for this work contains 500 records. When a user issues a query, which is not ambiguous, relevant records were fetched by the retrieval system. For a complex query, having difficulty metric above the threshold, the user is asked to provide description about the query with one or more keywords. Based on the keywords provided by the user relevant records were fetched. Numbers of records fetched were less for easy query, where as for complex query which is ambiguous, three or more records fetched with the most relevant record at the top of the list. The ranking quality of the results provided a good user satisfaction. The algorithm predicts the complexity of queries with fewer errors and in negligible time.

CONCLUSION AND FUTURE WORK:

This paper focuses on main problem of retrieving appropriate top results for a keyword query and predicting the difficulty level of the query. In this paper, we analyze the properties of complex queries and measure the degree of complexity of a keyword query over a database. SR algorithm is used for difficult keyword queries prediction over databases. We measured the degree of the complexity of a query over a database, using the ranking robustness principle. The framework efficiently predicts the effectiveness of a keyword query. The future work can be extending this framework to estimate query difficulty by using different entity sets and also on other ranking problems on databases. It can be extended for semi-structured keyword queries with operators and supporting phrases provided by the user.

REFERENCES:

[1] V. Hristidis, L. Gravano, and Y. Papakonstantinou, "Efficient IRstyle keyword search over relational databases," in Proc. 29th VLDB Conf., Berlin, Germany, 2003, pp. 850–861.

[2] Y. Luo, X. Lin, W. Wang, and X. Zhou, "SPARK: Top-k keyword query in relational databases," in Proc. 2007 ACM SIGMOD, Beijing, China, pp. 115–126.

[3] V. Ganti, Y. He, and D. Xin, "Keyword++: A framework to improve keyword search over entity databases," in Proc. VLDB Endowment, Singapore, Sept. 2010, vol.3, no. 1–2, pp. 711–722.

[4] J. Kim, X. Xue, and B. Croft, "A probabilistic retrieval model for semistructured data," in Proc. ECIR, Tolouse, France, 2009, pp. 228–239.

[5] N. Sarkas, S. Paparizos, and P. Tsaparas, "Structured annotations of web queries," in Proc. 2010 ACM SIGMOD Int. Conf. Manage. Data, Indianapolis, IN, USA, pp. 771–782.

[6] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan, "Keyword searching and browsing in databases using BANKS," in Proc. 18th ICDE, San Jose, CA, USA, 2002, pp. 431–440.

[7] C. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrieval. New York, NY: Cambridge University Press, 2008.

[8] A. Trotman and Q. Wang, "Overview of the INEX 2010 data centric track," in 9th Int. Workshop INEX 2010, Vugh, The Netherlands, pp. 1–32,

[9] T. Tran, P. Mika, H. Wang, and M. Grobelnik, "Semsearch ´S10," in Proc. 3rd Int. WWW Conf., Raleigh, NC, USA, 2010.

[10] S. C. Townsend, Y. Zhou, and B. Croft, "Predicting query performance," in Proc. SIGIR '02, Tampere, Finland, pp. 299–306.

[11] A. Nandi and H. V. Jagadish, "Assisted querying using instantresponse interfaces," in Proc. SIGMOD 07, Beijing, China, pp. 1156–1158.

[12] E. Demidova, P. Fankhauser, X. Zhou, and W. Nejdl, "DivQ: Diversification for keyword search over structured databases," in Proc. SIGIR' 10, Geneva, Switzerland, pp. 331–338.

[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.

[14] B. He and I. Ounis, "Query performance prediction," Inf. Syst., vol. 31, no. 7, pp. 585–594, Nov. 2006.

[15] K. Collins-Thompson and P. N. Bennett, "Predicting query performance via classification," in Proc. 32nd ECIR, Milton Keynes, U.K., 2010, pp. 140–152.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[16] A. Shtok, O. Kurland, and D. Carmel, "Predicting query performance by query-drift estimation," in Proc. 2nd ICTIR, Heidelberg, Germany, 2009, pp. 305–312.

[17] Y. Zhou and W. B. Croft, "Query performance prediction in web search environments," in Proc. 30th Annu. Int. ACM SIGIR, New York, NY, USA, 2007, pp. 543–550.

[18] Y. Zhao, F. Scholer, and Y. Tsegay, "Effective pre-retrieval query performance prediction using similarity and variability evidence," in Proc. 30th ECIR, Berlin, Germany, 2008, pp. 52–64.

[19] C. Hauff, L. Azzopardi, and D. Hiemstra, "The combination and evaluation of query performance prediction methods," in Proc. 31st ECIR, Toulouse, France, 2009, pp. 301–312.

[20] C. Hauff, V. Murdock, and R. Baeza-Yates, "Improved query difficulty prediction for the Web," in Proc. 17th CIKM, Napa Valley, CA, USA, 2008, pp. 439–448.

[21] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques. San Francisco, CA: Morgan Kaufmann, 2011.

[22] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow, "Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval," in Proc. 28th Annu. Int. ACM SIGIR Conf. Research Development Information Retrieval, Salvador, Brazil, 2005, pp. 512–519.

[23] J. A. Aslam and V. Pavlu, "Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions," in Proc. 29th ECIR, Rome, Italy, 2007,pp. 198–209.

[24] O. Kurland, A. Shtok, S. Hummel, F. Raiber, D. Carmel, and O. Rom, "Back to the roots: A probabilistic framework for queryperformance prediction," in Proc. 21st Int. CIKM, Maui, HI, USA, 2012, pp. 823–832.

AUTHOR DETAILS:

Author 1:

N.Srinivas,M.tech,Department of CSE,Global Institute of Engineering and Technology ,Chilkur,RR District,Telangana

Author 2:

Mr.Syed.Mazharuddin,Associate professor, Department of CSE,Global Institute of Engineering and Technology Chilkur,RR District,Telangana

Author 3:

Mrs. M.Jhansi Lakshmi,Associate professor,HOD of CSE,Global Institute of Engineering and Technology,Chilkur,RRDistrict,Telangana