ISSN 2395-1621

Complex Queries Prediction Approach using SR Algorithm and SGS Approx

[#1]Mr. Mandar Nalawade, [#2]Mr. Dinesh Bhange, [#3]Mr. Sudhir Mulkhede,
[#4]Mr. Rahul Birajdar, [#5]Prof. Santosh Waghmode

[2]dinesh.g.bhange@gmail.com
[3]sudhirmulkhede@gmail.com

[#1234]Department of Information Technology
[#5]Prof. Department of Information Technology

JSPM's
Imperial College of Engineering & Research, Wagholi, Pune.

ABSTRACT

Keyword queries on information bases offer easy accessibility to data, however usually suffer from low ranking quality, i.e., low exactitude and/or recall, as shown in recent benchmarks. it'd be helpful to spot queries that square measure probably to own low ranking quality to improve the user satisfaction. for example, the system could recommend to the user different queries for such onerous queries.. We set forth a high principled framework and proposed novel algorithms to live the degree of the issue of a question over a dB, exploitation the ranking strength principle. supported our framework, we tend to propose novel algorithms that with efficiency predict the effectiveness of a keyword question. Our intensive experiments show that the algorithms predict the issue of a question with comparatively low errors and negligible time overheads.

Keywords: **Query performance, query effectiveness, keyword query, robustness, databases.**

ARTICLE INFO

Article History

Received: 25th May 2016

Received in revised form :

25th May 2016

Accepted: 31th May 2016

Published online :

31th May 2016

I. INTRODUCTION

Keyword query interfaces (KQIs) for databases have attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Since any entity in a data set that contains the query keywords is a potential answer, keyword queries typically have many possible answers. KQIs must identify the information needs behind keyword queries and rank the answers so that the desired answers appear at the top of the list. Unless otherwise noted, it refers to keyword query as query in the remainder of this project. Databases contain entities, and entities contain attributes that take attribute values. Some of the difficulties of answering a query are as follows: First, unlike queries in languages like SQL, users do not normally specify the desired schema element(s) for each query term. For instance, query Q1: Godfather on the IMDB database (http://www.imdb.com) does not specify if the user is interested in movies whose title is Godfather or movies distributed by the Godfather Company. Thus, a KQI must find the desired attributes associated with each term in the query. Second, the schema of the output is not specified, i.e., users do not give enough information 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 over databases. Researchers have proposed some 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 particular, as mentioned earlier, a KQI must assign each query term to a schema element(s) in the database. It must also distinguish the desired result type(s).

II. LITERATURE SURVEY

"Efficient IR style keyword search over relational databases" IEEE conference interfaces (KQIs) for databases have 2003.

In this we studied that keyword query attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Since any entity in a data set that contains the query keywords is a potential

answer, keyword queries typically have many possible answers. KQIs must identify the information needs behind keyword queries and rank the answers so that the desired answers appear at the top of the list.

"An Introduction to Information Retrieval" IEEE Conference New York, NY: Cambridge University Press, 2008.

In this paper we studied that Databases contain entities, and entities contain attributes that take attribute values. Some of the difficulties of answering a query are as follows: First, unlike queries in languages like SQL, users do not normally specify the desired schema element(s) for each query term. For instance, query Q1: 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 company. Thus, a KQI must find the desired attributes associated with each term in the query. Second, the schema of the output is not specified, i.e., users do not give enough information to single out exactly their desired entities. For example, Q1 may return movies or actors or producers.

II. PROJECT OBJECTIVE

1. It would be useful to identify queries that are likely to have low ranking quality to improve the user satisfaction.

2. Suggest to the user alternative queries for such hard queries, considering both the structure and the content of the database and the query results.

III. PROPOSED SYSTEM

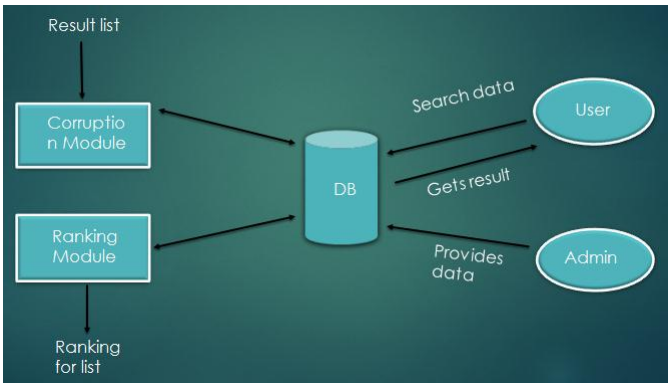

Fig 1. System architecture

Data and Query Modules:

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. It depicts a fragment of a data set where each sub tree whose root's label is movie represents an entity. Each entity E has a set of attribute values Ai, $1 \leq i \leq |E|$. Each attribute value is a bag of terms. Following current unstructured and (semi-) structure retrieval approaches, we ignore stop words that appear in attribute values, although this is not necessary for our methods. Every attribute value A belongs to an attribute T written as $A \in T$. For instance, Godfather and Mafia are two attribute values in the movie entity shown in the sub tree rooted at node 1 in Fig. 1. Node 2 depicts the attribute of Godfather, which is title.

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 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 collection. If these probability distributions are relatively similar, the query results contain information about almost as many topics as the whole collection, thus, the query is considered difficult. Several successors propose methods to improve the efficiency and effectiveness of clarity score.

Ranking-score-based:

The ranking score of a document returned by the retrieval systems for an input query may estimate the similarity of the query and the document. Some recent methods measure the difficulty of a query based on the score distribution of its results. Zhou and Croft argue that the information gained from a desired list of documents should be much more than the information gained from typical documents in the collection for an easy query. They measure the degree of the difficulty of a query by computing the difference between the weighted entropy of the top ranked results' scores and the weighted entropy of other documents' scores in the collection. Argue that the amount of non-query-related information in the top ranked results is negatively correlated with the deviation of their retrieval scores. Using language modeling techniques, they show that the standard deviation of ranking scores of top-k results estimates the quality of the top ranked results effectively. We examine the query difficulty prediction accuracy of this set of methods on databases, and show that our model outperforms these methods over databases.

Ranking Robustness Principle for Structured Data:

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. Mittendorf has shown that if a text retrieval method effectively ranks the answers to a query in a collection of text documents, it will also perform well for that query over the version of the collection that contains some errors such as repeated terms . In other words, 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. They compute the similarity between the rankings of the query over the origin land the artificially corrupted versions of a collection to predict the difficulty of the query over the collection. They deem a query to be more difficult if its rankings over the original and the corrupted versions of the data are less similar.

Hard Queries on Databases:

The more diverse the candidate answers of a query are, the more difficult the query is over a collection of the text documents. We extend this idea for queries over databases and propose three sources of difficulty for answering a query over a database as follows:

1. The more entities match the terms in a query, the less specificity of this query and it is harder to answer properly.

For example, there are more than one person called Ford in the IMDB data set. If a user submits query Q2: Ford, a KQI must resolve the desired Ford that satisfy the user's information need. As opposed to Q2, Q3: Spielberg matches smaller number of people in IMDB, so it is easier for the KQI to return its relevant results.

2. Each attribute describes a different aspect of an entity and defines the context of terms in attribute values of it. If a query matches different attributes in its candidate answers, it will have a more diverse set of potential answers in database, and hence it has higher attribute level ambiguity. For instance, some candidate answers for query Q4: Godfather in IMDB contain its term in their title and some contain its term in their distributor.

3. Each entity set contains the information about a different type of entities and defines another level of context (in addition to the context defined by attributes) for terms. Hence, if a query matches entities from more entity sets, it will have higher entity set level ambiguity.

IV. CONCLUSION

We introduced the novel problem of predicting the effectiveness of keyword queries over DB's. We showed that the current prediction methods for queries over unstructured data sources cannot be effectively used to solve this problem. We set forth a principled framework and proposed novel algorithms to measure the degree of the difficulty of a query over a DB, using the ranking robustness principle. Based on our framework, we propose novel algorithms that efficiently predict the effectiveness of a keyword query. Our extensive experiments show that the algorithms predict the difficulty of a query with relatively low errors and negligible time overheads.

V. FUTURE SCOPE

In the future we present our system on different domain datasets and extend the features according to user requirements.

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

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

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

[5] 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,

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

[7] O. Kurland, A. Shtok, D. Carmel, and S. Hummel, "A Unified framework for post-retrieval query-performance prediction," in *Proc. 3rd Int. ICTIR*, Bertinoro, Italy, 2011, pp. 15–26.

[8] S. Cheng, A. Termehchy, and V. Hristidis, "Predicting the effectiveness of keyword queries on databases," in *Proc. 21st ACM Int. CIKM*, Maui, HI, 2012, pp. 1213-1222.

