Efficient Prediction of Difficult Queries Over

Nutan S. Dalvi\textsuperscript{1}, Snehal R Mahale\textsuperscript{2}, Mayur P. More\textsuperscript{3} & Surabhi B. More\textsuperscript{4}

\textsuperscript{1,2,3,4}K. K. Wagh Institute of Engineering Education \& Research, Nashik, Maharashtra, India.

Abstract—Keyword queries on databases allows us to access data easily, but most of the time suffer from low ranking that is, low precision or recall. It would be useful to identify queries that are likely to have low ranking quality to improve the user satisfaction. At that instance, the system may suggest some alternative queries to the user such hard queries. In the existing work, predict the characteristics of hard queries and propose a novel framework to measure and analyze the degree of difficulty for a keyword query over a database, considering both the structure and the content of the database and the query results. However, in this system there are number of issues. They are searching quality is lower than the system and reliability of the system is lowest. In order to overcome these drawbacks, to perform operation in three levels includes attribute level, attribute value level and entity set level in the database. Propose system is well enhancing the reliability rate of the difficult queries prediction system. The system can predict the characteristics of hard queries and propose the novel framework to measure the degree of difficulty for keyword queries over database. Time requires for predicting the difficult queries keyword over large dataset can be minimize and process can become more robust and accurate.

Index Terms—Clustering, Query performance, query effectiveness, keyword query, Robustness, databases.

I. INTRODUCTION

Keyword query interfaces for databases have attracted much attention in the last decade due to their flexibility and ease of use in searching and exploring the data. Keyword queries characteristically have many possible answers. Keyword query interfaces must identify the information needs behind keyword queries and rank the answers so that the expected answers appear at the top of the list. Databases consist of entity set and their attributes values. In some cases problems of giving results of query are likely to have users do not specify the preferred schema element for each query term. For example, keyword BigBrother on the movie database does not specify that user is interested in title or Distributor Company. So, a keyword query interface must find the desired attributes associated with each term in the query and users do not give sufficient information about their desired entities. For example, keyword may return movies or actors or producers. Recently, there have been joint efforts are taken for giving standard benchmarks and evaluation platforms for keyword search methods over databases. One of the effort is the data-centric track of INEX Workshop where keyword query interfaces are evaluated on the well-known IMDB data set which contains structured information about movies and people.

One more effort is the series of Semantic Search Challenges in this workshop where the data set is the Billion Triple Challenge. It is extracted from Wikipedia. The queries are taken from any website Keyword query log. Users have provided relevance judgments for both benchmarks. Results of these judgment indicate finding the preferred answers to keyword queries with structure data is also a hard task. In both workshops ranking quality of methods are used, observed that they performing very poorly on a subset of queries. E.g., consider the query ancient Rome era over the IMDB data set. Users want to see information about movies that talk about ancient Rome.

For this query, the XML search methods which we can implement return rankings of considerably lower
quality than their average ranking quality of all queries. Hence some queries are more difficult than others. Furthermore, It does not matter which method of ranking we are using, it cannot deliver a reasonable ranking for these queries. It is important for a Keyword Query Interface to identify such queries and warn the user or employ alternative techniques like query reformulation or query suggestions. It may also use different techniques such as query results diversification. But there has not been any work on analyzing or optimizing the difficulties of queries over databases.

In this paper our goal is to present a new and improved method for difficult keyword prediction. In this paper approach we are extending the framework we are applying the K-means clustering to dataset into number of clusters those having legitimate information’s. Then we are calculating SR score Due to this, Time required for predicting the difficult keywords over large dataset is minimized and process becomes robust and accurate.

II. LITERATURE SURVEY

Literature survey is one of the most important process in software development process. Before developing the system it is necessary to determine the, economy ,time factor and company strength. Once these things are determined, some more next steps are to determine which operating system and language can be used or implementing the tool. Once the programmers start implementing the system the programmers need lot of external support. This support can be obtained from senior programmers, from reference book or from many other related websites. Before starting implementing the system the above consideration are taken for implementing the proposed system. Here in this paper the analysis of the features of difficult queries over databases is done. It propose improved method to analyze such queries. Here we take advantages of the structure of the data to gain insight about the level of the difficulty of a query on given the database. We can also built some most popular and well enhancing algorithms for searching keyword on databases such as structure Robustness algorithm and can be used them to implement our system.

There are many approach or method for predicting the hard queries over database. It can be as categorized in to method:

1) **Pre-retrieval method**:
Pre-retrieval method is to predict the difficulty of a query without evaluating its results. This method usually use the statistical method of the terms in the query to measure ambiguity, specificity and term-relatedness of the query to analyze and predict its difficulty.

2) **Post-retrieval methods**:
Post-retrieval methods method evaluates the results of given query to predict and analyze its difficulty and they are having two types:

- Clarity-score-based:
The Clarity-score-based methods are based on the concept of clarity score that assume the users are interested in a very less topics.
- Ranking-score-based:
The ranking score of a given document returned by the retrieval systems for an input query may built the similarity of the document and query.

III. ALGORITHM

A. **Structured Robustness Algorithm**:
Algorithm given shows the (SR Algorithm) Structured Robustness Algorithm, which calculate the exact SR score based on the top K result.

```
Algorithm 1 CorrectTupleResults(Q, L, M, I, N)
Input: Query Q, Top-K result list L of Q by ranking function g.
Metadata M, Inverted index I, Number of corruption iteration N.
Output: SR score for Q.
1: Initialize SR score for Q, C ← { } // Cachex, λ(I) for keywords in Q
2: FOR i = 1 to N DO
3: I’ ← L; M’ ← M; L’ ← L; //Corrupted copy of I, M and L.
4: FOR each result R in L DO
5: FOR each attribute value A in R DO
6: A’ ← A; //Corrupted version of A
7: FOR each keywords w in Q DO
8: Compute k of w in A’ by Equation 10, if λ(I’,w), λ(I,w) needed but not in C, calculate and cache them
9: IF w varies in A’ and A THEN
10: Update A’ and M’ and entry of w in I’;
11: Add A’ to R’;
12: Add R’ to L’;
13: Rank L’ using g, which returns L’, based on I’, M’;
14: SR ← Spearman(L’, L), //Spear computes Spearman correlation
15: RETURN SR ← SR/N, if NNG score over N rounds
```

Each ranking algorithm uses some of the statistics related to attributes values or query terms over the database. Some of the examples of such statistics are the no. of occurrences of a query term in all the values the attribute of the database or total number of values of attribute in each attribute and entity set. These all statistics which are global are stored in (metadata) m and (inverted indexes) I in the Structured Robustness Algorithm pseudo code.
B. K-means algorithm:
Here we can apply the K-means clustering to divide input database into many number of clusters those having legitimate information’s.

Steps for algorithm to create the cluster are:
1.lace K points into the space represented by the objects that are being clustered.
2. allocate each object to the group that are having the closest centroid.
3. When all objects have been allocated, again calculate the positions of the K centroids.
4. Repeat the above Steps 2 and step 3 until the centroids has no longer move. This creates a separation of the objects into many groups from which the metric can be minimized can be calculated.

In this paper the main goal is to represent new efficient and improved approach for prediction of difficult keywords by avoiding the limitations of dataset flexibility, string approximation and scalability. This New novel approach is nothing but enhanced framework in which before calculating the robustness score we are going to apply K-means clustering to divide input dataset into no. of clusters those having legitimate information’s. Because of this, required time require for predicting the difficult keywords on large dataset is minimized and process then becomes more accurate and robust. Also in addition, spatial approximate string query is also presented. In advance we will be using linguistic features information Such as syntactical features, morphological features, semantic features for efficient prediction of difficult keyword queries on dataset.

VI CONCLUSION
This New approach is nothing but extended framework in which before going to calculate SR score we are applying the K-means clustering to divide input dataset into number of clusters those having legitimate information’s. As well it will measure the degree of the difficulty of a keyword query over a database, using the ranking robustness principle. The algorithms predict the difficulty of a query with relatively low errors and negligible time overheads.

REFERENCES
[4] Survey on optimized prediction of hard keyword queries over databases, Mr. Sameer P Mamadapure 1, Prof. P. D. Lambhate 2, Volume IX, Issue VI, and June 2015.