International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 08 | Nov-2015 www.irjet.net p-ISSN: 2395-0072

Traditional Approach to Predict Hard Queries using Keyword Analyzer over Data bases

E Samuel Raju [#] P Srinivas[*] D D D Suri Babu[**]

[#] Student , M.Tech (CSE) , DNRCET
**Assistant Professor, Dept. of CSE , DNRCET*
***Associate Professor, Dept. of CSE , DNRCET*

Abstract : ***Querying is a common task to search any information with respect to huge database. Finding correct results from the given query is a challenging task. To predict such hard queries we propose a novel framework which uses association analysis to find the top k results from the search keyword. In this paper we propose an algorithm to find the top k searched keyword items to the user data with combination of keywords. This probabilistic method will predict the results quickly. The SR algorithm is also applicable along with this for ranking the top k results. Here we use IMDB database to predict the results by applying different keywords to the query Q .In this traditional approach it uses fp tree to generate the frequent patterns and find set of rules for re- ranking. We can also implement the keyword classifier to predict difficult keyword queries.***

Keywords : Classifier , dataset , Structured Robustness, attributes, analyzer

INTRODUCTION

Keyword Query search is really typical tasks to search engines. Data can be reside in either structured or unstructured format. The main aim of this work is to retrieve the user interested data from the data bases for hard queries. Queried results are ranked so that the required data is shown at the top of the search. Data sets contains entities and entities contains set of attributes. if we ask the query as the required attribute the answers are very predictable . other wise it will search all related documents of particular attribute entities. Datasets can be present in either xml formats or relatioinal data format. INEX workshops data set having data in the form of xml (structured data) it is collection of structured data. for example KQI must find the desired attribute associated with each keyword in the query. For example query q god father in IMDB[1] database ,it doesn't specify whether user asks for movie or director or distributer of particular attribute it collects all relevant information of the attribute from data sets and produce to the users. Results proven that even in structured data it is very hard to predict subset of queries with ranking. INEX and semantic search gives almost equal equal results. If we consider a keyword present in all documents we couldn't provide effective ranking to the documents. In this scenario the user interestingness was measured before search the data with combination of keywords. Predicting hard queries from unstructured documents can be done in two ways.

1. Pre retrieval and

2. Post retrieval methods

In pre retrieval, predicting hard queries is done without computing results, internally it uses statistical analysis

Post retrieval methods [1] utilize results of a query to predict its difficulty .it has three approaches

1. Clarity score based

2. Ranking score based

3. Ranking robustness

Clarity score based: in this approach only few documents are retrieved based on query difficulty so it is very easy predict the results

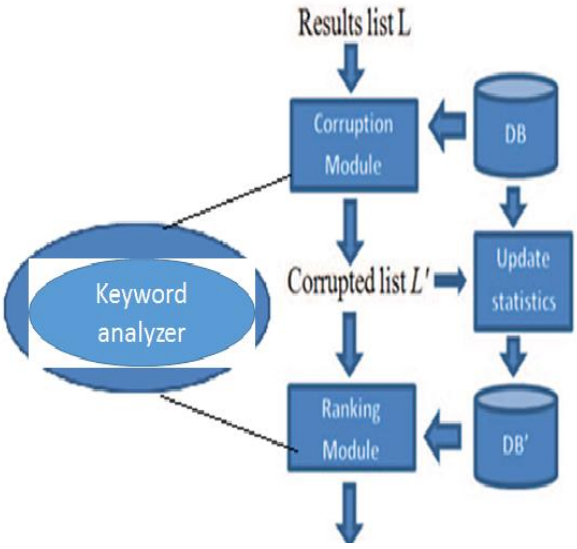
Ranking score based: they measure the degree of difficulty of a query by computing difference between weighted entropy of top ranking result scores and weighted entropy of documents

Robustness based: there is a negative correlation between the difficulty of a query and its ranking robustness in the presence of noise in the data

2. Estimation techniques for ranking the query

Query results are shown in some order called as top K Results in the database. A structured data set contains large amount of information stored in structured format say XML like in Data centric [6] workshops (INEX). We usually corrupt the top K results and re ranked the results over corrupted data base .we can retrieve semi structured data also using probabilistic approach [5]. Several estimation techniques were already proposed so we are here to define an algorithm for re ranking the results

3. Architecture



In this model we extend S-R Algorithm by adding a keyword analyzer. The keyword analyzer will collect all tuples of related keywords by using fp growth algorithm. It finds the frequency of each keyword and also its combination keywords. The probability of each keyword has to be estimated based on the frequency count and results will be updated. This probabilistic method is so simple because of using traditional algorithms. In this approach finding keywords from unstructured data is difficult. Structured documents are easily ranked for difficult queries. We can extend this approach for finding different attributes and entities

Keyword Analyzer

Keyword analyzer is a software component which is used to find all frequent keyword count and generate the corresponding tree rank the results .In this model it simply rank the keywords based on the previous user hits. We can use structured robustness algorithm to rank top k results from the generated list. SR Score based approximation is not needed in this approach while comparing the results it gives the same results with minimal overhead.

Keyword analyzer algorithm in query evaluation

Step 1: Select input as set of keyword tokens

Where T = { t1, t2, t3,.. }

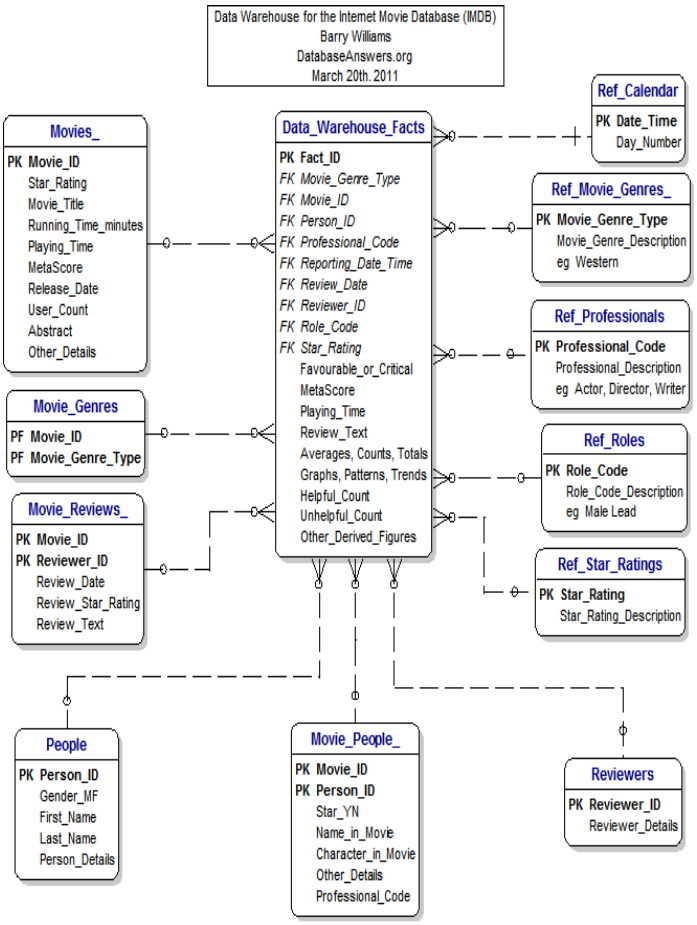
Step 2: generate most frequent hits for combination of tokens K

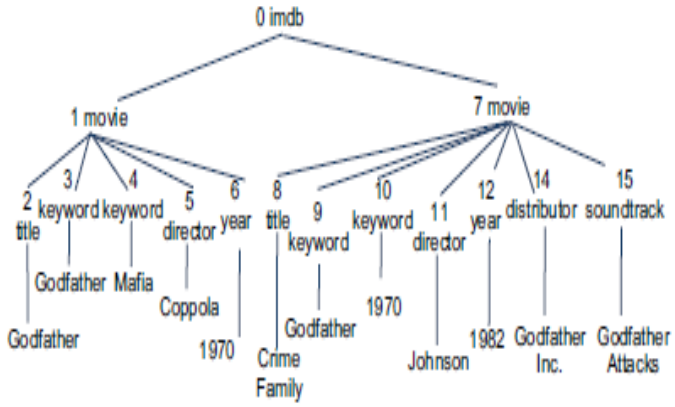
Step 3: Generate Frequent hit pattern tree

Step 4: retrieve most frequent combination keyword

Step 5: rank the results using SR Algorithm

Consider movie database IMDB,





In this above data the keyword godfather is considered with different combinations like movie, director, and actor and generates the frequency count for each combination then the results will be projected. By using keyword analyzer it is very easy to understand and the results are almost similar without using any approximation algorithms. The key is the usage of xml while using xml documents. We can easily navigate tag attributes from one object to another. The more key terms matched will give the best results in keyword analyzer approach. We can view all attribute lists for a given entity in keyword analyzer technique. The prediction of results can also be estimated [7]

S R Algorithm

Structured Robustness algorithm generates the noise over database during query processing. It loops over all attributes in corrupted list to check whether it is corrupted or not. One entity may have hundreds of attribute values. So it takes considerably some more time to work.SR Algorithm has to re rank the top k entities for corruption. SR Algorithm takes long time to loop that re ranks the corrupted result. Along with keyword analyzer it assigns a frequent count of each keyword in the documents to rank the results. It will generate better results without using any approximation algorithms for improving structured robustness.

Conclusion

In this paper, we propose an efficient SR Algorithm without finding approximations. We built a component called keyword analyzer. Keyword analyzer works on combination of keywords to built a model and generate the corrupted results of top K Ranked queries efficiently. Using of keyword analyzer will generate the frequent occurring top K Ranked queries over corrupted data bases. Consider god father as keyword in IMDB Database. It generates set of rules over movies directors and actors and rank the results in appropriate manner. This model is applicable to both structured data and normal text documents. We will improve this algorithm efficiency in future for unstructured data also. It works fine over xml document formats also.

References:

[1] *Shiwen Cheng; Termehchy, A.; Hristidis, V., "Efficient Prediction of Difficult Keyword Queries over Databases," in Knowledge and Data Engineering, IEEE Transactions on , vol.26, no.6, pp.1507-1520, June 2014*

[2] V Hristidis , L Gravano and Y Papakonstantinou " Efficient IR Style keyword search over relational database "

[3]Y Lou , X Lin , W Wang and X Zhou , "SPARK" Top –k keyword query in relational databases

[4]J Kim X Xue and B Craft " A Probabilistic retrieval model for semistructured data" ECIR, Tolouse , France

[5]N Sarkas S Paparizos and P Tsaparas " Structured Annotations of web queries " ACM SIGMOD , USA

[6]A Trontman and Q Wang " Overview of INEX 2010 data centric track " Workshop INEX 2010 , Netherlands

[7] S C Townsend Y Zhou and B Croft " Predicting Query Performance " , Finland