Enhancing Query hardness detection with answer extraction using Ontology over the relational databases

Jyoti Katiyar
Department of Computer Engineering
SKN Sinhgad Institute of Technology and Science
Lonavala, Pune, India
Email: Jyotikatiyar.cs@gmail.com

Prof. Deepali Khatri
Department of Computer Engineering
SKN Sinhgad Institute of Technology and Science
Lonavala, Pune, India
Email: dmkhatri.sknsits@sinhgad.edu

***Abstract*—** Most of the information retrieval systems in the relational databases need the simpler keywords for the specific attribute in the database. But as the searching system becomes more and more popular the databases are also growing accordingly in gigantic manner. So it is obvious that users are always using complex queries to extract the much desired data from the database if they are not having any prior knowledge about the database structure. If the system is able to provide the complexity or difficulty of the query to the user then he/ she can rearrange the query in much simpler form to get the desired answer. So many systems are been existed to provide the answers for the difficult queries which are working in one or two aspects of the solving issues. So this paper represents an idea of extracting the hardness with the answer for the difficult query in more meaningful manner. The proposed idea is enriched with ontology for the semantic relations of the extracted features from the fired difficult query and inverted indices are used to catalyze the retrieval process more efficiently.

Keywords:-*Spearman correlation, Ontology, features, inverted index, hardness.*

*****

I. Introduction

In information retrieval system it is very difficult to fire the precise query as the user not know the content or type of the data on which he/she is going to retrieve the answer. In such system there are some queries which are unable to produce the answers in spite of being well managed nature, such queries are known as a difficult queries. Hence to retrieve the data for such difficult queries is a challenging task. Numbers of systems are introduced to predict the difficulty of the query. More the difficulty in the query less precise will be the answer. The systems that predict the query difficulty are generally categorized in two steps.

- Pre-retrieval
- Post-retrieval

In pre-retrieval scheme the system doesn't wait for result of the query computation. Before results calculate it predict the query based on the unique words of the query. In case of the post retrieval scheme the system waits for the answer extraction. After extraction it gives the query difficulty based on the nature of the answer.

Now a day Keyword query interfaces (KQI) is at glance because of its functionality in the area of searching and extracting data. In normal information retrieval system if data contains the query word then the system will raise that data. But it increases the size of the extracted data as raised data may contain the data which is nearby to the query word. KQI solves this problem very smoothly, it find the information need behind the query and thus it rank the desired answers which reduces the size of the extracted data to the great extent. Recent study shows that the poor quality of the query to fetch the data from the database reduces the quality of the answer to be extracted.

In case of a database like MySQL, it contains the entities and further that entity has attributes to store the attribute value. Whenever user fires a query on the database he/she will never going to specify the desired schema. Consider the query Black, here database don't know that what exactly Black is. Is it movie? Or it is name of the color? So KQI plays a very important role in such scenario. Here KQI will identify the desired attributes of the query word. And in second step KQI find the schema associated with the Black word.

To set some set of standards a study has been done. In first study INEX workshop is conducted where number of participants are participated in the workshop. In this an IMDB dataset is used as a repository. Here KQI's applied on the IMDB dataset which contains a lots of information regarding movies and the peoples related with the movies.

In second study semantic search workshop is conducted where semantic search is given as a challenge where billion trillion dataset from the website www. Vmlion.deri.de is taken as repository. From these two studies researchers calculated the mean average precession of the searching words which is 0.36 for the INEX and 0.2 for the semantic search. By looking to the figure it has been proved that it is very difficult to find out the exact answer for the queries.

Table 1 shows some of such hard queries observed in the study

INEX	SemSearch
Ancient Rome era	Austin Texas
Movies Klaus Kinski actor good ranking	Carolina
True story drug addiction	Earl May

Table 1: Some Difficult Queries from Benchmarks

Hence to avoid irritation of the user some artificial intelligence is needed that will warn the user if system unable to find out the answer.

Our system makes use of ontologies to overcome some of the problems of the keyword searching methods. Ontologies are used to keep shallow representation of the information with the interrelated relationship. Traditional information retrieval approaches are failed to search precisely, because system will unable to find the semantic of the keyword. Ontologies help in finding schema of the keyword by taking background knowledge of the keyword. Thanh Tranet al [1] give a good example of ontology based information retrieval. Here he try to map the user query to the expressive queries. Because of this task the efficiency of the extracted result is increase to great extent. Ontologies can be used to transform the natural language to the SPARQL. Along with ontologies lexicons can be used to accomplish the task.

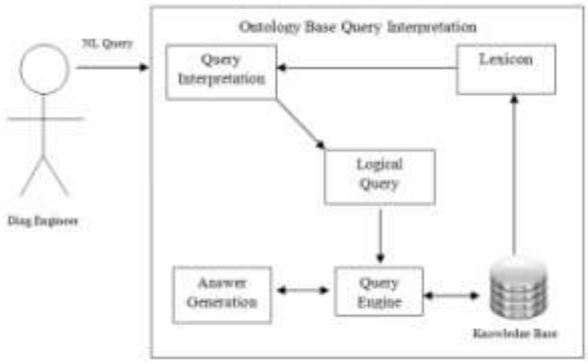

Figure 1: A general approach to ontology-based query interpretation.

So for this reason we are proposing an idea with a scenario of a university, where university web application is having thousands of student's data, with around 20 attributes. Now user can fire a complex query to the system to get the answer. So our system identifies the hardness of the query using the generalized inverted index merged with the spearman correlation along with extracted answer for the given query using well blended ontology with strong NLP protocols.

The rest of the paper is organized as follows. Section 2 discusses some related work and section 3 presents the design of our approach. The details of the results and some discussions we have conducted on this approach are presented in section 4 as Results and Discussions. Sections 5 provide hints of some extension of our approach as future work and conclusion.

II. LITERATURE SURVEY

This section represents all the related works of technologies used in our project

Pre-processing is technique of reducing the size of the data to avoid the unnecessary attributes of the data. It reduces the size of the data to a great extent hence it plays a very important role in an applications where size of the data is very large along with the unnecessary data. Suppose input query is "What is the distance between Pune to Mumbai? "After pre-processing the system will give only three words: "distance, Mumbai, Pune".Initial query contains 8 words, after pre-processing it gives only three words. So data gets reduced from 8 to 3.Hence we can say that pre-processing is used to remove the supporting data which will not going to change the meaning of the query.

Normally pre-processing is carried out in four important steps.

Generalized preprocessing has four steps as below.

1. Data Cleaning
2. Data Integration
3. Data transformation
4. Data reduction

Stemming is one of the important sub processes of preprocessing. Here derived words are converted to the base form. Similar to the stemming lemmatization is also there. Both of this having very slight difference. Stemming refers to a process the word where semantic of the word is not considered while in lemmatization semantic of the word is considered.

[2] Narrates the exact difference of stemming and lemmatization approach. Here six various techniques are explained very well. Advantage and disadvantage of each system with another are elaborated. All this approaches are well putted in tabular format so it will get easy for the user to understand.

[3] Explains a context aware stemming approach. The main intention of the method is to decrease the morphological difference of the source query. The working of the algorithm is inspired from well-known stemming algorithm known as port stemmer. To accomplish the task of bringing the words to its root form rule based approach is used. Vast varieties of

algorithms are proposed in a same stream with slight difference. Affix removing is one of such algorithm effectively used for the stemming purpose.

Feature extraction is a method of choosing the important part that contributes the result by discriminating the unwanted part. Depend on the application to be developed a list of features will get change. Feature extraction is vital process used for data mining and web mining. Feature extraction methods are briefly classified in two sub parts as below

1. *Principal Component Analysis*
2. *Linear Discriminant Analysis*

Feature extraction process is normally carried out in four important steps i.e. generate subset, evaluate subset, stop condition and validate the result.

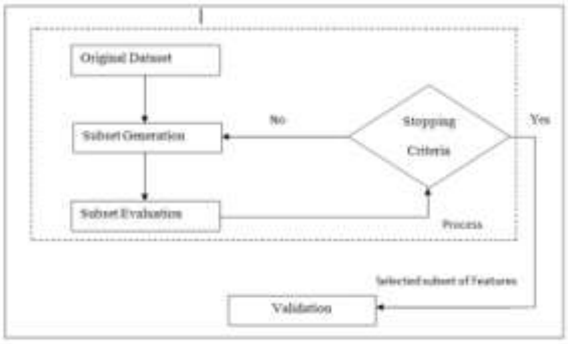

Figure 2: Feature extraction process

- Subset generation

In this process a certain strategy is applied to generate the subset of features from the original dataset.

- Subset Evaluation

Here generated subsets are tested and evaluated against the evaluating criterion. The criterion is set to obtained the good quality of the subsets.

- Stopping criterion

Here a criterion is set to stop feature extraction process.

- Result validation

Once feature extraction process stops, the generated subsets are validated by using the prior knowledge about the data.

[4] Presents a SR algorithm for finding the query difficulty. Author addressed the problems that are faced while finding the difficulty of the query. There are some methods proposed in the same area but they are not giving the desired results. Here forth framework is proposed to find the percentage of the difficulty. Also the ranking robustness principle is used as supporting percentage finder technique. Developer makes use of IMDB: movie dataset for the implementation purpose.

[5] Elaborates a novel approach to overcome the problem of time complexity presents in earlier systems. To accomplish the task author makes use of ranking algorithms. Finally they concludes that the system has a great efficiency in terms of accuracy rate and quality of the result. [6] explores a new approach for finding the drifting percentage of the query.To bring down the idea in to reality author states that the standard deviation is a best way.

To find the characteristics of hard queries [7] sets a good benchmark. Shiwen Chenget. Al sets a theory to find out the query difficulties and the characteristics which makes the system more difficult. While putting the theory on paper both the nature of the database i.e. structure and content of the database is considered. As explained in introduction, to evaluate the performance of the system two benchmarks i.e. INEX and SemSearch are used. With great confidence author concludes that the system achieves a high degree of accuracy compared to the systems of the same stream.

In order to overcome the problem of imprecise result [8] gave a one solution. In this general framework proposer coverts the keyword query to the structured query. An algorithm is proposed that maps the keyword to its predicates. To find the predicators a tokenization method is used which separates the complete query to the tokens. Because of this provision exact answers for nearby queries can be find out. [9] Presents a good technique for predicting the query difficulty. In order to find the difficulty complete query result and sub query results are taken. Since it takes full query and sub query it give the precise and more accurate result.

Because of lack of availability of automated methods [10] explored an automatic method to get the task done. Differencing's completely from other traditional methods it returns set of scoring functions using the input query.

III. PROPOSED METHODOLOGY

In this section, we describe our framework for query hardness detection and answer extraction with the below mentioned steps as shown in figure 3.

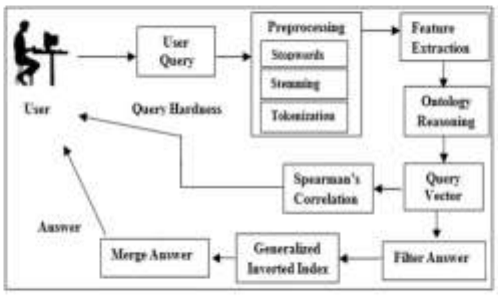

Figure 3: Overview of the proposed work

Step 1: Here user enters a difficult query to get the desired answer from the data over the university respiratory.

Step 2: This is the step where preprocessing is conducted, by the following methods

- ✓ Sentence segmentation
- ✓ Tokenization
- ✓ Stop word removal
- ✓ Stemming

Step 3:-*Feature extraction: As* mentioned in earlier segments the features of a text play a vital role in semantic categorization techniques. So our system makes use of two different features, which are extracted from query entered by the user as mentioned below.

- ✓ Numeric data
- ✓ Term weight.

Step 4: Ontology Reasoning – Here in this part of the system the features words need to be identify for their respective attributes in the relational database. So System read an owl (web ontology language) file which is created with the help of the famous ontology tool protégé. Protégé helps to design the hierarchy of the attributes for the university database respiratory as shown in the below figure 4. Where the parent of the each leaf represents the specific attribute name in the database.

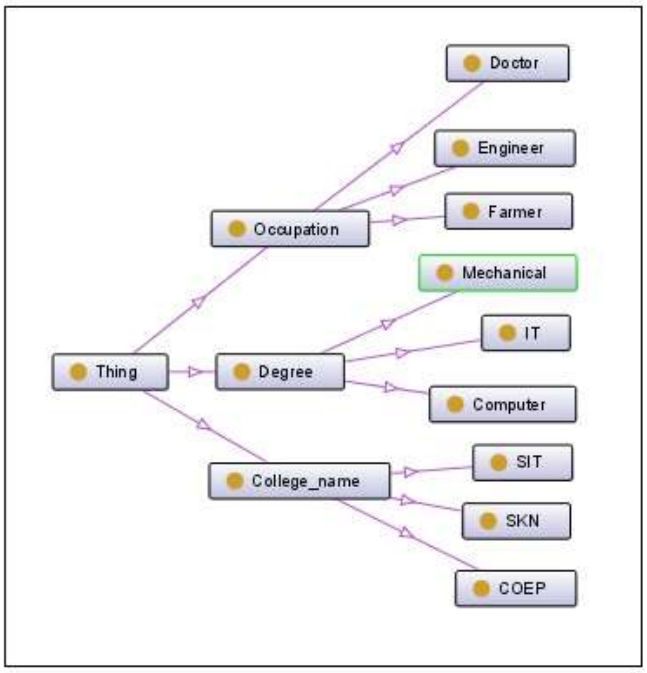

Figure 4: Ontology graph created by protégé

So by reading this pre developed owl file which is in RDF format of xml system efficiently identifies the hierarchy and then find the proper attributes in the database.

Step5: Hardness calculation using spearman correlation : In this step query hardness is identified with respect to the query vector using the similarity measure of the spearman rank correlation which is a non-parametric measure of correlation

(A) Query hardness can be found using spearman's rank correlation as shown below

$$p = 1 - \frac{\left(6 \sum di^2\right)}{n(n2-1)} \text{------------------(1)}$$

d= difference between two numbers of pairs of ranks belongs to two answers

n= no of pairs of result Vector

This step compute the similarity of the answer lists using Spearman rank correlation. It ranges between 1 and −1, where 1, −1, and 0 indicate perfect positive correlation, perfect negative correlation, and almost no correlation, respectively. This directly puts efforts to yield the hardness factor for the given query.

Step 5: This is the last step of the process where all individual keywords are gathered from the query vector to form the proper queries . Then queries are been committed linearly which is been supported by the generalized inverted indices to minimize the searching space to speed up the process.

(B) The Generalizes inverted index can be represent by following below two equations

$$I(x) = \int_0^n \sum Q \in A(x) \text{ ....(2)}$$

I(x) =inverted index
Q=Query Set
A=answer vector
n=no of words in query

$$G_n(x) = \lim_{x \to n} I_n(x) \cap \lim_{x \to m} I_{n'}(x) \text{ ..(3)}$$

G_n = Generalized inverted index
I_n= Inverted Index Set

And then answers are collected in a common vector. These answers are been filtered and then merged to get the fine gained result for the entered difficult query.

The complete process of hardness measurement and answer extraction can represent by the following pseudo code

OVERALL SYSTEM PSEUDEO CODE

```
Step 0: Start
Step 1: Read query Q
Step 2: Preprocess the query Q
```

Step 3: Find Numeric data
Step 4: Find Top Records
Step 5: Divide query Q into records and store in a vector V
Step 6: For i=0 to length of V
Step 7: Find semester S and Percentage P
Step 8: By reading ontologies find father occupation F, College C, and department D
Step 9: End for loop
Step 10: Fetch data D from dataset by using the obtained attributes
Step 11: Using spearman correlation find hardness of query H
Step 12: Return H and D
Step 13: Send mail of extracted answer and hardness of query to the respected user
Step 14: End

IV. RESULTS AND DISCUSSIONS

To show the effectiveness of the proposed system some experiments are conducted on java based windows machine using Apache tomcat as the server. To measure the performance of the system we set the bench mark by entering the possible hard queries on our university dataset based on the following tests.

4.1 Performance Time

As it is clearly explained in the previous section that Our system uses the inverted indices to fast the searching process. When our system is set to measure the timing for the answer extraction method for linear search method and Ginix method we got the following results as shown in the figure 5.

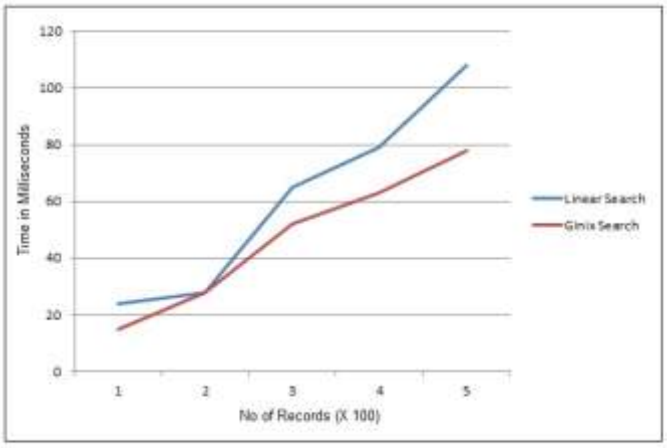

Figure 5 : Performance graph for linear search and Ginix Search

Above figure clearly indicates the using of ginix is always an advantage for our system for better performance in searching process as ginix is taking considerably less time for searching records from the database compare to linear search .

4.2 Precision Calculation

To determine the performance of the system, we examined how many relevant queries are detected for their proper hardness based on the spearman correlation and ontology.

To measure this precision is considering as the best measuring techniques. So precision can be defined as the ratio of the number of relevant extracted records for the hardness for the given hard query to the total number of irrelevant and relevant extracted for the hardness records. It is usually expressed as a percentage. This gives the information about the relative effectiveness of the system.

When the experiment is conducted to measure the precision system is producing the results as shown in the figure 6.

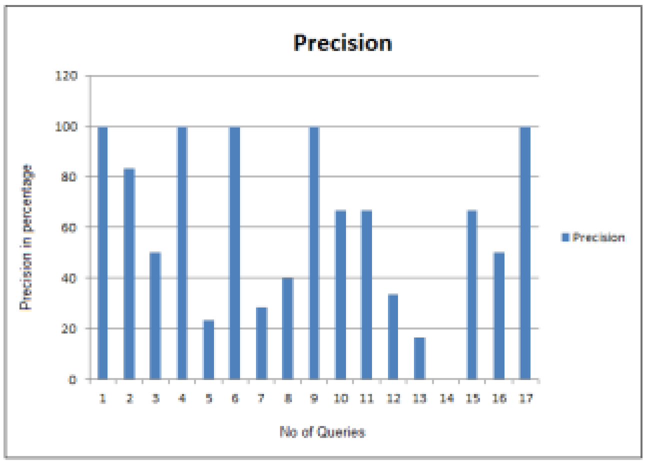

Fig.6. Average precision of the proposed approach

For more clarity precision can be calculated as

$$Precision = (A/ (A+ C))*100$$

Where
A=Relevant records in T for the measured hardness
C= Number of irrelevant records extracted for the measured hardness

In Fig. 6, we observe that the tendency of average precision for the measured hardness for the proposed system is about 0.6029 whereas the measured hardness identification of the for method proposed by Shiwen Cheng, Arash Termehchy, and Vagelis Hristidis [11] is around 0.507. So hardness detection which is powered by the ontology for the given hard query is always giving good precision over the methods proposed in [11].

V. CONCLUSION AND FUTURE SCOPE

Many of the systems are existed to measure the hardness of the complex queries which are fired over the relational databases using the similarity or correlation methods. These methods are not consistence as they are not based on linguistic reasoning of the query words. So our system presented a heuristic approach where system divides the query based on their semantics using NLP and ontology. This actually provides us individual small replacement queries for the complex query. Then our system makes use of these individual queries to get the hardness using spearman correlation. Then meaningful answer is also extracted from the database by successfully merging and filtering the answer which is powered by generalized inverted index.

As future work of this framework, this can be enhancing to get the answer and to detect the hardness for all different database schemas by developing this as advanced plugin software by providing some easily Integratable API even in distributed paradigm.

This can be achieved by allowing the developer of the project to do some setting by mentioning

- ✓ Database attributes
- ✓ Ontology classes
- ✓ Ontology relations

As our API parameters. This complete process enhances the system of many searching system which are dealing with the complex query problems.

REFERENCES

[1] Peter Haase, Daniel Herzig, Mark A. Musen, and Thanh Tran. Semantic wiki search. In The Semantic Web: Research and Applications, 6th European Semantic Web Conference,ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings (LNCS 5554), pages 445–460, 2009.

[2] MichalKonkol and MiloslavKonopík , “Named Entity Recognition for Highly Inflectional Languages: Effects of Various Lemmatization and Stemming Approaches” Named Entity Recognition for Highly Inflectional Languagesby M Konkol - 2014 - Sep 12, 2014

[3] K.K. Agbele, A.O. Adesina, N.A. Azeez , A.P. Abidoye “Context-Aware Stemming Algorithm for Semantically Related Root Words” © 2012 Afr J Comp & ICT.

[4] “Review: Predicting The Efficiency of Difficult Keyword Queries Over Databases” Miss. VarshaVetal, Mrs. SanchikaBajpai. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 11, November 2014

[5] “A Novel Approach for Predicting Difficult Keyword Queries over Databases using Effective Ranking” G. Ramakrishnan1, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV4IS030251 www.ijert.org (This work is licensed under a Creative Commons Attribution 4.0 International License.) Vol. 4 Issue 03, March-2015

[6] “Predicting Query Performance by Query-DriftEstimation” Anna Shtok1, Oren Kurland1, and David Carmel2by A Shtok - 2012

[7] “Predicting the Effectiveness of Keyword Queries on Databases” Shiwen Cheng, ArashTermehchy October 29–November 2, 2012, Maui, HI, USA.

[8] “Keyword++: A Framework to Improve Keyword Search Over Entity Databases” VenkateshGanti∗ Proceedings of the VLDB Endowment, Vol. 3, No. 1 Copyright 2010 VLDB Endowment 2150-8097/10/09

[9] Elad Yom-Tov, Shai Fine, David Carmel, Adam Darlow, ” Learning to Estimate Query Difficulty”www.yom-tov.info/Papers/SIGIR2005

[10] Cronen-Townsend, S., Zhou, Y., Croft, W.: Predicting query performance. In: In Proceedings of the ACM Conference on Research in Information Retrieval (SIGIR). (2002)

[11] Shiwen Cheng, Arash Termehchy, and Vagelis Hristidis, " Efficient Prediction of Difficult Keyword Queries over Databases" , IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 6, JUNE 2014