

Query Recommender System using Fragments and Projection values Based F-measure and MongoDB

Laxmi Patil¹, Manjusha Tatiya²P.G. Student, Department of Computer Engineering, ICEM Engineering College, Parendewadi, Pune, India¹Associate Professor, Department of Computer Engineering, ICEM Engineering College, Parendewadi, Pune, India²

ABSTRACT: A key challenge in getting the intended output from the data warehouse is query preparation. Even for the expert user to search the information from available data is a tough task. This project primarily aims to provide the easiest way to search and recommend the information as per users need. This project is useful to create recommendations for those systems which are based on the SQL query interface and operates considering the users are unaware about the actual schema. This in fact provides them the information about schema and their attributes for search. It also facilitates options for all conditions which a user wants to apply to filter the available data. The search truly works effectively as all the cells have been identified with the specific value; this also enhances the accuracy the precision of the recommendation results. The system has been designed keeping in mind the time required to generate a query recommendation and the computing complexity associated with it. This enables user to search a data with less efforts simply by selecting the available options in the drag-down list. The list contains all the possible options for each condition and ensures no result is missed. The system is implemented with the MongoDB database which work on unstructured database and compare the result with the system implemented with the SQL database which is structural database system.

KEYWORDS: Collaborative Data Mining, Fragment based and Tuple based recommendation method, MongoDB, Relational Database.

I. INTRODUCTION

Data Mining is the procedure of discovering valuable data from the information. Information might be quantitative or subjective. In numerous applications (for instance advertising or business related), the information might be vast (otherwise called huge information). Finding data from the gigantic measure of information might be troublesome for the end clients who need SQL mastery. In such circumstances database investigation assumes an interesting part. Significant information disclosure is troublesome for clients and proposed method is an answer for such trouble as Recommendation System. Recommendation system is an active research area in the data mining and machine learning areas. The popular ones are probably movies, music, news, books, research articles, search queries, social tags, and products in general.

In this project the recommendation system is considered for the applications which use filtered information through SQL query. When a new user interacts with database system, they have to face some problem e.g. type of database and how to retrieve needed information from that database. Even skilled people are also facing the same problem to execute some complicated query. The Genome browser or Sky Server are the examples of scientific database application where user is not aware about the actual content of the relation and want to search some record based information. The data size of that application is also large that new user cannot predict expected result by executing query. To overcome this problem, system helps user by providing user friendly method to select a relation, their fields and also apply number of conditions on that relation by only selecting basis. By that easy method user can easily generate a SQL query without any complicated task. Query recommendation system will continuously monitor the user's querying behavior and find matching patterns in the system's query log, and to identify previous users with similar in formation needs. Subsequently, system uses those similar users and their queries to recommend queries that

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 9, September 2016

current user may find interesting. Despite the availability of querying tools over large database, users often have difficulties in understanding the underlying schema and formulating queries. Recommendation systems are a subclass of information filtering system that seek to predict the rating or preference that user would give to a particular item. This system also store all the history of users with executed query result and also provides recommendation from the other user session history and provides edit option in every situation. The implementation of this system is based on fragment method over the relational database in which the recommendation on the basis of word to word matching and projection value is calculated. This also implements part of system on MongoDB which is non-structured database and compare result from both the systems.

II. LITERATURE SURVEY

Study done by G. Chatzopoulou¹ for Interactive Database Exploration [1] illustrates that Social database frameworks are turning out to be progressively famous in the scientific group for intelligent investigation of substantial amount of information. In such situation, clients utilize a question interface (ordinarily, an online customer) to the progression of inquiries that expect to break down the information and search it for intriguing data. To help clients in this connection, Author designed A “Web recommender frameworks” which help to the utilization of customized question suggestions. [1]. A Survey done by K. Stefanidis [2] discussed that, inclinations have been generally contemplated in logic, brain research, and financial matters and connected to choice making issues. Databases convey an entire new viewpoint to the investigation of point of interest. From a representational point of view, the focal inquiry is the way system can effectively speak to inclinations and join them in database questioning. The reason for this study is to give a structure to putting already present works in context and concentrate on basic difficulties to serve as a springboard for scientists in database frameworks [2]. B. Smith conclude in Item-to-Item Collaborative Filtering [3], most suggestion calculations begins by finding an arrangement of clients whose bought and appraised things cover the client's obtained and evaluated items. The calculation totals things from these comparative clients, wipes out things the client has as of now acquired or evaluated, and prescribes the remaining things to the client. Two mainstream forms of these calculations are cooperative separating and bunch models. [3]. Hierarchical Classification [4] method used by S. Patil discussed that, in information distribution centers, heaps of information are accumulated which are explored and investigated for logical purposes. For master individuals, to handle such extensive information is an extreme errand. Taking care of such voluminous information is more troublesome undertaking for non-master clients or for clients who are not acquainted with the database diagram. The point of this paper is to help this class of clients by prescribing them SQL questions that they may utilize. These SQL suggestions are chosen by following the clients past conduct and contrasting them and different clients. The questions are recorded and looked at utilizing various level arrangements which is then repositioned. The pertinent inquiries are recovered utilizing clients questioning conduct. Clients utilize an inquiry interface to issue a progression of SQL questions that expect to examine the information and mine it for fascinating data [4].

III. PROPOSED ARCHITECTURE

The recommendation mechanism used for Query processing is the combination of Tuple based system and fragment based system.

The system work as follows:

1) Query Processing & Mapping

As shown in Figure 1. Input query which is accepted by user is processed for finding redundant and irrelevant data and then preprocessed and converted in to an analyzable format. An example of preprocessing is shown below:

```
SELECT Id, COUNT (field) AS title0field FROM AREA WHERE TYPE LIKE 'MAHARASHTRA' OR Id > 10.0 GROUP BY Id AND NAME HAVING COUNT (*) > 1
```

```
SELECT Id, COUNT (*) FROM AREA WHERE TYPE [STATE] OR Id COMPARE NUM GROUP BY Id AND NAME HAVING COUNT (*) COMPARE NUM
```

In the above example > symbol is replaced with COMPARE LIKE 'MAHARASHTRA' is replaced with PATMATCH and numeric values are replaced with NUM.

2) Fragment Generation

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 9, September 2016

Text mapping section in Figure 1 extracts the fragments from each query and these fragments are separated and stored in the database. Pattern matching is done to identify the fragments. Then for every fragments projection value get selected. Here system assume some value for every attributes.

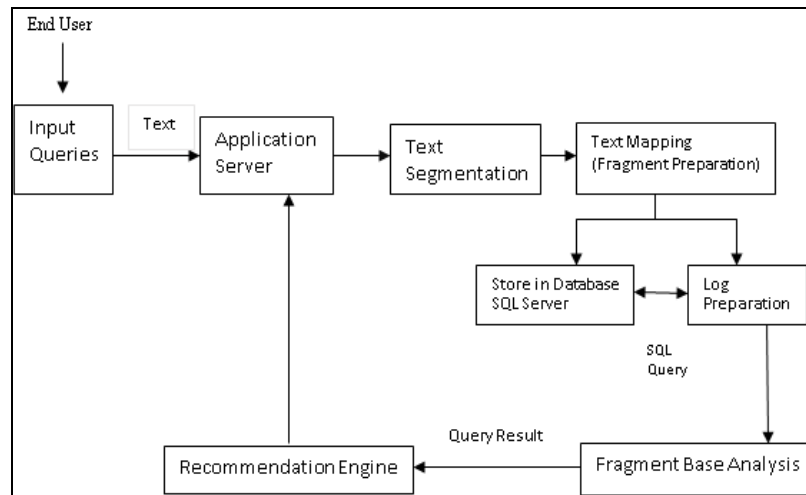


Figure 1: System Architecture

3) RECOMMENDATION ENGINE

This system provides Recommendations by combining the fragment based recommendation and context aware fragment recommendation, So that the relevance of the recommendation can be improved.

IV. RELATED WORK

To give a recommendation to user about query as per data they were entered, first system want to find out the weighted keywords from the query. For this task system used fragment based method.

1. Fragment bases method: Fragment based method is works on the similarity of query fragments like table name/relation name, column name, and predicated conditions. In this method, system needs to find out fragments that co-appear in several queries posed by different user. The session summary vector S_i for a user I consists of all the query of the users past queries.
2. Query Form: This system use "Query Form" concept, which is used to store the result of query. Query form is an interface which is used for querying database. The essence of DQF is to capture user interests during user interactions and to adapt the query form iteratively.
3. Projection Value: This method used the projection value (between 0 to 1), assigned to the every attributes present in database. The system use F-measure to calculate the goodness of the query.

Given a set of projection attributes A and a universe of selection expressions σ , the expected precision and expected recall of a query $Q = (A_f, R_f, \sigma_f)$ are Precision (Q) and Recall (Q) respectively, i.e.,

$$\text{Precision (Q)} = \frac{\sum_{d \in DAQ} P_u(d_{AQ}) P(d_{AQ}) P(\sigma_f|d)N}{\sum_{d \in DAQ} P_u(d_{AQ}) P(\sigma_f|d)N} \dots\dots(1)$$

$$\text{Recall (Q)} = \frac{\sum_{d \in DAQ} P_u(d_{AQ}) P(d_{AQ}) P(\sigma_f|d)N}{\alpha N} \dots\dots(2)$$

Here,

Q= Query

R_f = Set of relation involve in Query

σ_f = Set of selection attribute in query

N= number of data instances is D

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 9, September 2016

A_f = Set of projection attribute present in query. D = is the collection of Data instances in R_f

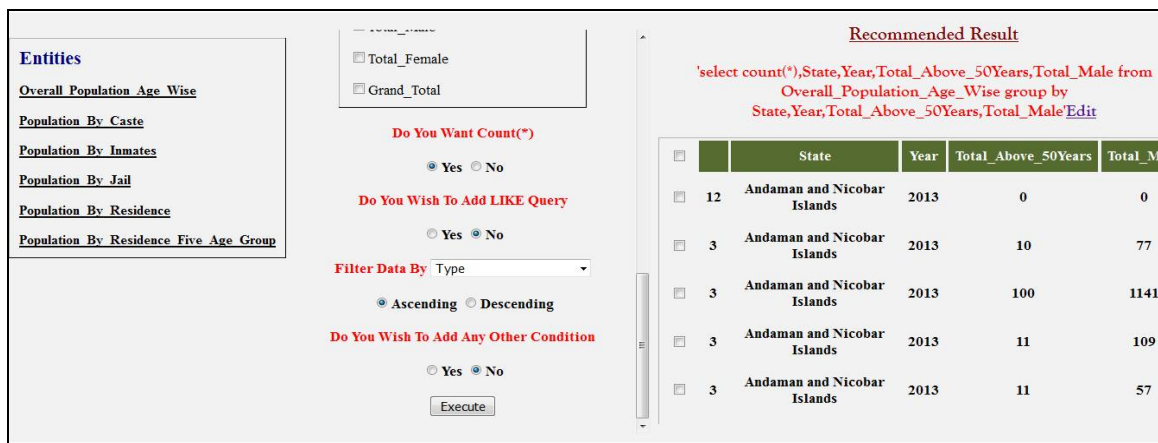
Base on this Precision and Recall values, system calculate the F-measure value to assign the rank to every query. Here system considers threshold value to recommended highly weighted query. And then recommendations are shown to user.

4. Ranking: Query forms are designed to return the user’s desired result. There are two traditional measures to evaluate the quality of the query results: precision and recall. Query forms are able to produce different queries by different inputs, and different queries can output different query results and achieve different precisions and recalls, so we use expected precision and expected recall to evaluate the expected performance of the query form. Intuitively, expected precision is the expected proportion of the query results which are interested by the current user. Expected recall is the expected proportion of user interested data instances which are returned by the current query form. The user interest is estimated based on the user’s click-through on query results displayed by the query form. For example, if some data instances are clicked by the user, these data instances must have high user interests. Then, the query form components which can capture these data instances should be ranked higher than other components. Next we introduce some notations and then define expected precision and recall.

5. Ranking Entities: The ranking score of an entity is just the averaged FScore of that entity’s attributes. Intuitively, if one entity has many high score attributes, then it should have a higher rank.

V. EXPERIMENTAL RESULTS

This system generates Query recommendation as Follows:



The screenshot shows a web interface for query generation. On the left, under 'Entities', there are several options: Overall Population Age Wise, Population By Caste, Population By Inmates, Population By Jail, Population By Residence, and Population By Residence Five Age Group. In the center, there are checkboxes for 'Total_Female' and 'Grand_Total', followed by 'Do You Want Count(*)' (Yes/No), 'Do You Wish To Add LIKE Query' (Yes/No), a 'Filter Data By' dropdown set to 'Type', 'Ascending/Descending' radio buttons, and 'Do You Wish To Add Any Other Condition' (Yes/No) with an 'Execute' button. On the right, under 'Recommended Result', a SQL query is shown: `'select count(*),State,Year,Total_Above_50Years,Total_Male from Overall_Population_Age_Wise group by State,Year,Total_Above_50Years,Total_Male'Edit`. Below the query is a table with columns: State, Year, Total_Above_50Years, and Total_M. The table contains five rows of data for Andaman and Nicobar Islands in 2013.

	State	Year	Total_Above_50Years	Total_M
12	Andaman and Nicobar Islands	2013	0	0
3	Andaman and Nicobar Islands	2013	10	77
3	Andaman and Nicobar Islands	2013	100	1141
3	Andaman and Nicobar Islands	2013	11	109
3	Andaman and Nicobar Islands	2013	11	57

Figure2: Selection Of Table, Attribute And condition

As shown in Figure2, system provides easy way for user to generate query only by selecting table name, column name, and condition. Base on that selection system gives the all recommendation query list which is fetch from the query log which match the current query fragment with the all query present in the log. All top rated query are show to the user as a recommendation list, as shown in figure 3.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 9, September 2016



Figure3: Recommendation Generated for User given Query

As shown in figure3, all top ranked queries are extracted from database by calculating F-measure. User can select query and make changes as needed.

Total Queries = 56 Recommended Queries = 34 Elapse Time = 00 : 00 : 5 milliseccs

Snapshot of the Results for Recommended Query Is Shown Below -

select count(*) as count1,State,Female_16_18years from Overall_Population_Age_Wise where State like '%ch%' or Year like '%20%' group by State,Female_16_18years		
count1	State	Female_16_18years
27	Andaman and Nicobar Islands	0
27	Andhra Pradesh	0
27	Arunachal Pradesh	0
27	Assam	0
27	Bihar	0
27	Chandigarh	0

Figure4: Query result from recommended list with count and Elapse Time

Figure4 shows the result from the recommendation query list with the count of all query present in log and only match query count, and also shows the time required for matching the fragments.

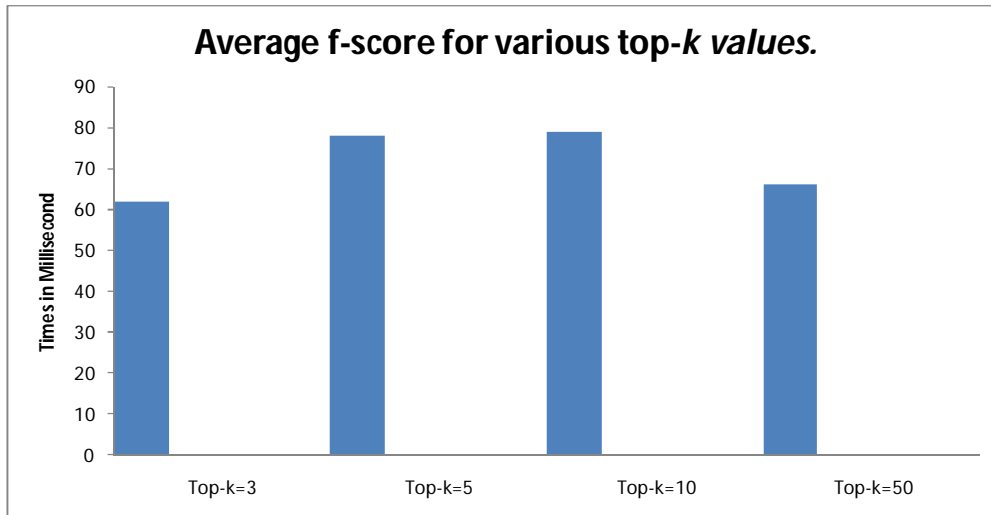


Figure5 Average f-score for various top-k values

Figure5 shows system employs the top-k fragments of previous queries in order to generate recommendations, show the effect of the choice of k on the average precision and F-score for the recommendations. I notice that the accuracy of the recommendations increases, as expected, with the value of k. However, for very large values of k ($k > 10$), the accuracy starts decreasing again. This is completely justifiable, since when k is a very large number, the notion of “most similar” fragments does no longer hold and barely similar items are included in the recommendation process, as shown in Figure5.

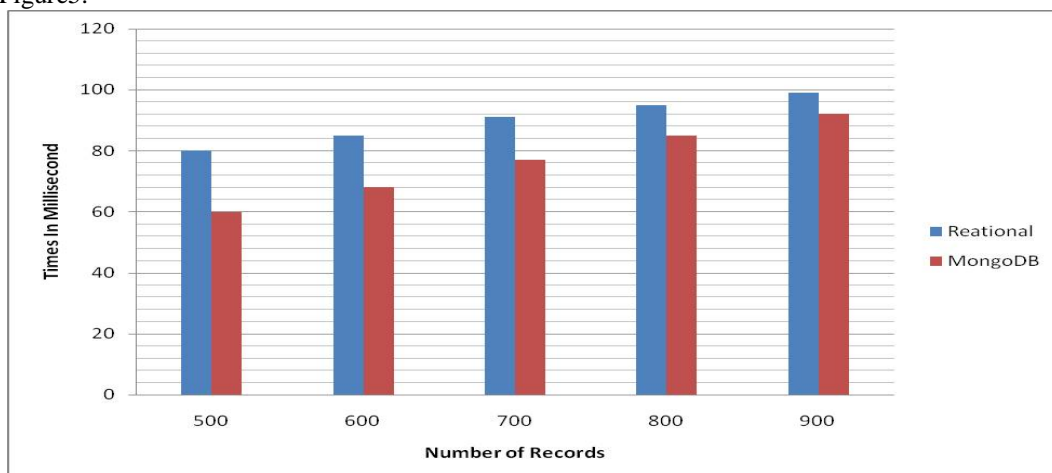


Figure6: Compare result for the retrieved records from the Relational Database and MongoDB database

VI. CONCLUSION

In spite of the accessibility of questioning over expansive databases, clients frequently experience issues in comprehension of the hidden mapping and planning inquiries. Intuitive Exploration of huge database to help clients who need SQL ability or commonality with the database pattern is a test. Shared Item based proposal with Fragment based recommendation and Tuple based recommendation gives proficient suggestion to the user. Context mindful section based strategy enhances the adequacy of the recommendation. This system concludes that the projection value calculation method gives higher precision result than word to word matching technique. MongoDB database decreases the response time and increases the scalability with respect to the Relational database.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 9, September 2016

REFERENCES

- [1] Gloria Chatzopoulou, Magdalini Eirinaki, and Neoklis Polyzotis, "Query Recommendations for Interactive Database Exploration," Computer Science Dept., University of California Riverside, USA chatzopd@cs.ucr.edu, 2009.
- [2] Kostas Stefanidis "A Survey On Representation, Composition And Application Of Preferences In Database Systems" Usa And Evaggelia Pitoura University Of Ioannina, Greece.
- [3] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering", IEEE Internet Comput., vol. 7, no. 1, pp. 7680, Jan./Feb.
- [4] Shruti Y. Patil, Prof. Dinesh D. Patil, "Revisiting SQL Query Recommender System Using Hierarchical Classification" ISSN: 2321-8169 ,Volume: 3 Issue: 6 4190 4193.
- [5] Arnaud Giacometti, Patrick Marcel, and Elsa Negre, "Recommending Multidimensional Queries", DawaK 2009, LNCS 5691, pp. 453466, 2009. Springer-Verlag
- [6] Magdalini Eirinaki, Suju Abraham, Neoklis Polyzotis, and Naushin Shaikh, "QueRIE: Collaborative Database Exploration" , VOL. 26, NO. 7, JULY 2014
- [7] Jiawei Han and Jing Gao "Research Challenges for Data Mining in Science and Engineering". Recommender systems A Computer Science Comprehensive Exercise Carleton College, Northfield, University of Illinois at Urbana-Champaign,.
- [8] MN. Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, "Dynamic Query Forms for Database Queries", IEEE Transaction on Knowledge and Data Engineering, VOL. 26, NO. 9, SEPTEMBER 2014
- [9] Google. (2007). Freebase [Online]. Available: <http://www.freebase.com>
- [10] Giovanna Castellano¹, Anna Maria Fanelli¹, Maria Alessandra Torsello¹, and Lakhmi C. Jain², Innovations in "Web Personalization, Springer" Verlag Berlin Heidelberg 2009.
- [11] Cristóbal Romero¹, Sebastián Ventura¹, Jose Antonio Delgado¹, and Paul De Bra², "Personalized Links Recommendation Based on Data Mining in Adaptive Educational Hypermedia Systems", Cristóbal Romero¹, Sebastián Ventura¹, Jose Antonio Delgado¹, and Paul De Bra², 2007.
- [12] Kostas Stefanidis Chinese University of Hong Kong, Hong Kong Georgia Koutrika IBM Almaden Research Center, USA and Evaggelia Pitoura University of Ioannina, Greece, "A Survey on Representation, Composition and Application of Preferences in Database Systems", V, No. N, M 20Y