Collaborative Database Exploration for Generic Approach

Satish Pokharkar, Umesh Wadnere, Kapil Tupe, Mahesh Gaikwad

***Abstract*—The user approached to DBMS applications to discover the knowledge by formatting queries with the help of declarative SQL query language. User faces the difficulties in formulation of SQL queries as they lack the proficiency in the domain of SQL language. Recently in the field of knowledge and data engineering Collaborative Database Exploration framework has been introduced to address the important problem of assisting users in the interactive exploration of a large database. This framework is called as Query which assists users of ad-hoc or from-based query environments by presenting them with personalized query recommendations, data scarcity, not scalable and generic etc. To address these problems, the hybrid and extended system called GSQueRIE (Generic-Scalable QueRIE) will be proposed with the goal of improving the overall scalability and flexibility of collaborative database exploration. Using this system, user will be able to explore the database interactively to get Top N query recommendations. GSQueRIE will accept SQL query as a input and generates list of Top N recommended queries with the help of Matrix Factorization method. Also it will compare the performance of existing QueRIE and proposed GSQueRIE for precision, recall and F-score metrics.**

Index Terms*— *DBMS, Collaborative Database Exploration, Generic-Scalable QueRIE, Matrix Factorization, Query recommendation.

I. Introduction

In With the growing Information Technologies everybody is directed to use computers and different technologic-cal devices for their needs. These needs may be personal, academic, business etc. Wide spread of internet has made the people to do online shopping, search for the required information on web and many others. Social networks are also used all the times. This has generated large amount of data. Such huge amount of data is stored in Database Management Systems. Data belong to various types and is accessed and analyzed by different users such as scientists manage and analyze large amount of experimental data, financial experts process financial data etc. All the users tend to achieve same goal of obtaining valuable and useful information out of large data volumes. Due to rapidly increase in data, complexity in accessing data also increases. In this situation, both technical and non-technical users need assistance to query and retrieve the data. But to query and retrieve the data, user need to use Structured Query Language (SQL). As most of the users are not familiar with SQL, they face difficulties in interacting with the database as the database has hundreds of schemas and thousands of attributes [1].

Many Database Management Applications such as SAP, Easy Query, and Microsoft Access are available to solve these user problems. Since these applications involve the user interaction, manual editing of the SQL query components and the user is unacquainted about the underlying database structure, it becomes difficult for the user to formulate the query. Hence, the idea of recommender system comes into existence. Recommender systems [2] are software tools and techniques providing suggestions for items to be of use to a user.

The goal of a recommender system is to deliver lists of personalized recommended object which is evaluated based on predictions. The recommendation ratings are given to the object those are unknown to the user and objects with the highest recommendation scores will be recommended to the active users. Therefore, there is need to design a Query Recommendation tool which will assist the technical and non-technical users in forming SQL queries.

Highlight a section that you want to designate with a certain style, and then select the appropriate name on the style menu. The style will adjust your fonts and line spacing. **Do not change the font sizes or line spacing to squeeze more text into a limited number of pages.** Use italics for emphasis; do not underline.

II. WORKFLOW OF RECOMMENDATION AND EXISTING SYSTEM

In the workflow of recommendation, active user puts up the queries in the interface provided. This query is forwarded to both DBMS and Recommendation Engine. The DBMS processes each query and return set of results. Simultaneously, the query is stored in query log and Recommendation Engine combines the current user's input with information gathered from the database interactions of past users, as recorded in the Query Log. Then it generates a set of query recommendations that are returned to the user. This is shown in Fig-1.The existing QueRIE system follows this approach and generates recommendations for the users. In this QueRIE, Fragment based approach is used to generate query recommendations. QueRIE has also compared the precision, recall and F-score of the system. However, QueRIE framework is having below listed research problems to address in future work:

- Required to measure the impact of query relaxation process that has in the quality of recommendations.
- Data Scarcity exits.

- Cold start problem exits
- Need to explore a sequence-based approach.
- Existing framework is not scalable and generic.

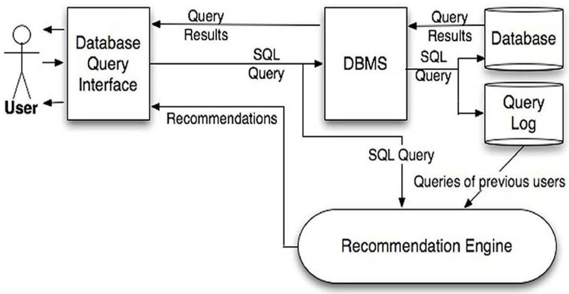

Fig. 1 Workflow of Recommendation

A. Problem Statement

In the field of knowledge and data engineering, the Collaborative Database Exploration QueRIE framework has been introduced to assist the users by providing ad-hoc or form-based query environments. QueRIE helped the user to formulate SQL queries and generate recommendations. But this system is not scalable and does not measure the impact of the query relaxation process. Hence, GSQueRIE (Generic-Scalable QueRIE) system is designed with the goal of improving the overall scalability and flexibility of collaborative database exploration which generates the Top N query recommendations along with certain ranks by using Matrix Factorization method. The comparison between the performance of existing QueRIE and proposed GSQueRIE for precision, recall and F-score metrics will be done.

B. Methodology

In proposed solution active user can: i) formulate a query, ii) select a recommended query and submit, iii) select a recommended query and edit it before submitting. When user submits a query, query is preprocessed, similarity is computed and Matrix Factorization method is applied and query recommendations are generated. GSQueRIE mostly focuses on using Matrix Factorization and generating the recommendations. The main aim is to compare the existing system and proposed system in terms of Precision, Recall and F-score. In proposed GSQueRIE system, the following concepts are important.

C. Scalable Similarity

The weight of each SQL Fragment will be computed by using weighting scheme. For this, the result based weighting scheme will be used. The weight is computed by using the TFIDF method. Similarity between current user and number of logged users will be identified and user predicated summary Vector Up red will be computed for each user. To compute similarity, Euclidean distance based similarity measurement method will be used.

D. Precision, Recall & F-score

The effectiveness of each recommended query can be described with the help of Precision, Recall and F-score. Recall and F-score. Precision metric shows the percentage of "interesting" query to the user with respect to all recommended queries. It describes how many selected items are relevant. The recall metric captures the hit ratio of each recommended query with respect to the last query of the user. It describes how many relevant items are selected. In other words, high precision means that an algorithm returned substantially more relevant results than irrelevant. High recall means an algorithm returned most of the relevant results. F-score is a measure of a test's accuracy. The F-score can be intercepted as a weighted average of precision and recall, in which F-score has its best value at 1 and worst score at 0. The comparison between existing system and proposed system will be done in terms of precision, recall and F-score..

III. SYSTEM ARCHITECTURE

A. User Design

When a new user is logged to the system, user will be asked for registration. This module consists of user frontend, login and registration pages. The user profile is also created. The active users will interact with the form based user interface in which they will be able to formulate the query. User needs to type query in the provided textbox and click on submit button. Query can be simple query, select query or complex queries such as join, projection, aggregate etc. The backend database will also be designed. The result of this module is generalized query for next processing. Hence, the major work of this module is:

- Design and implement user frontend, login and registration pages
- Design backend database for product recommendations
- Implement approach of query preprocessing
- Result of this module is generalized query for next processing

System architecture is a shown in *fig.2*. There are 4 modules in the proposed GS QueRIE system.

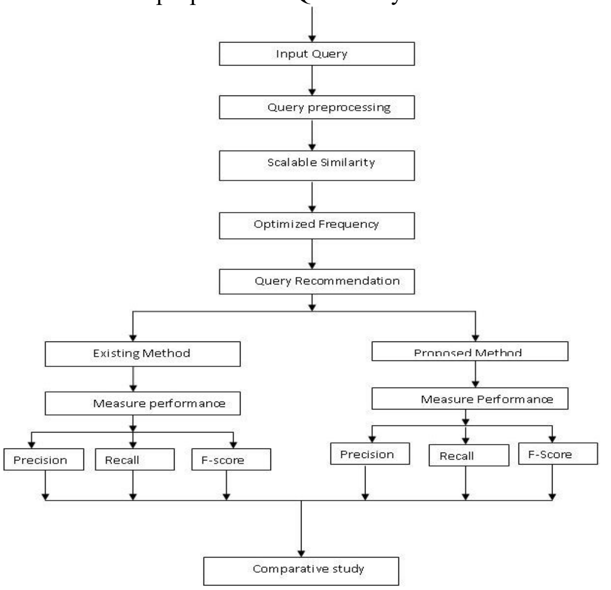

Fig.2 Proposed System Architecture

B. Similarity and Recommendation

Once the queries are generalized, they are converted into the fragments. The weight of each fragment will be computed by applying weighting scheme. For this, the result based weighting scheme will be used. In this method, the weight is computed by using the TF-IDF method. Similarity between current user and number of logged users will be identified and user predicated summary vector up red will be computed for each user. To compute similarity, Euclidean distance based similarity measurement method will be used. Similarity between current user and past user will be identified and predicated summary vector pared will be computed. Hence, the major work of this module is:

- Implementation of Scalable similarity calculations
- Implementation of optimized fragment based approach

C. Result and Comparative Study

By taking the similarities from above module, Matrix Factorization method will be applied for finding the latent features for the prediction and recommendations are generated. List of top N query recommendation s are provided along with the certain ranks as output to the users. Along with this, performances of both existing and proposed system will be measured in terms of precision, recall and f-score are compared. After viewing of resultant recommendations, the generated recommended query will be stored in user's template for future use. Hence, the major works of this module are,

- Implementation of Matrix Factorization algorithm for generating the recommendations.
- Measuring the performances in terms of precision, recall and f-score for both existing and proposed system.
- Compare the performance.

IV. MATHEMATICAL MODEL

In this Query I is submitted to state q1 where the query log is maintained then it is passed to state q2 where the session summaries are extracted then in state q3 the target topples are generated and in q4 state re-ranking is done on Clarity score formula and the output is generated in final state O.

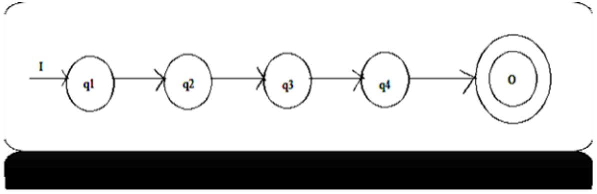

Fig.3 Mathematical Model

Input Parameter (I):

I= I1 where I is set of Input.
I1= query which is submitted.

Functional Parameter (Q):

Q= q1, q2, q3, q4
Where, Q is functions/process done by recommendation engine.
q1= Maintaining of query log.
q2= Extraction of session summary.
q3= Generation of target tuples.
q4= Re-ranking based on clarity score.

Output Parameter (O):

O=O1
Where, O is an Output Parameter.
O1=Result generated by the Recommendation Engine.

V. CONCLUSION

In the proposed system, User will be able to formulate SQL query and get Top N Query recommendation list. Active user can: i) formulate a query, ii) select a recommended query and submit, iii) select a recommended query and edit it before submitting. GSQueRIE will work on simple query, select query, join, projection or aggregate query. When user submits a query, query will be preprocessed and weighted scheme will be applied. Similarity will be computed and matrix Factorization method will be applied to generate recommendations. The output will be Top N query recommendations list. The most important is that GSQueRIE focuses on using Matrix Factorization method and generating the recommendations. The aim is to compare the existing system and proposed system in terms of Precision, Recall and F-score and to prove that proposed solution provides better results.

REFERENCES

[1] MagdaliniEririnaki, SujuAbraham, Neoklis Polyzotis, and Naushine Shaikh, "QueRIE: Collaborative Database Exploation,"in IEEE Transaction on Knowledge and Data Engineering, 2013.

[2] S. Mittal, J. S. V. Varman, G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "QueRIE: A Recommender System supporting Interactive Database.

[3] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen, " Dynamic Query Forms for Database Queries",in IEEE Transaction on Knowledge and Data Engineering, 2013.

[4] Xiaoyuan, Taghi, and M. Khoshgoftarr, "Review Article- A Survey of Collaborative Filtering Techniques", in Advances in Artificial Intelligence, vol. 2009, Article Id 421425, Aug 3, 2009.

[5] Yehuda Koren, Yahoo Research, Robert Bell and Chris Volinsky, AT & T Labs-Research, "Matrix Factorization Techniques for Recommender System", in IEEE Computer Society, AT & T Labs, 2009.

[6] Christopher Miles, "More like This: Query Recommendation for SQL".

[7] Gloria Chatzopoulou, MagdaliniEririnaki, and NeoklisPolyzotis, "Query Recommendations for Interactive Database Exploration", in Springer SSDBN 2009, LNCS 5566, PP. 3-18, 2009.

[8] L. Lu, M. Medo, C.H. Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and T. Zhou, "Recommender System", Feb 7, 2012.

[9] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "Collaborative filtering for interactive database exploration," in Proc. of the 21st Intl. Conf. on Scientific and Statistical Database Management,2009.

Prof. Satish T. Pokharkar, M.Tech [CSE] is Assistant Professor in Shri Chhatrapati Shivaji College of Engineering, Rahuri Factory, Rahuri, Dist-Ahmednagar, Pin-413706 Maharashtra, INDIA. He is having 5 years of teaching experience. His research interests include Data Mining Information Retrieval Engineering.

Prof. Umesh P. Wadnere, M.Tech [CSE] is Assistant Professor in Shri Chhatrapati Shivaji College of Engineering, Rahuri Factory, Rahuri, Dist-Ahmednagar, Pin-413706 Maharashtra, INDIA. He is having 3.5 years of teaching experience. His research interests include Image Processing, Data Mining Information Retrieval Engineering.

Prof. Kapil A. Tupe, M.E [CSE-Appear] is Assistant Professor in Shri Chhatrapati Shivaji College of Engineering, Rahuri Factory, Rahuri, Dist-Ahmednagar, Pin-413706 Maharashtra, INDIA. He is having 3 moths of teaching experience. His research interests include Big Data, Data Mining Information Retrieval Engineering.