Scientific Journal of Impact Factor(SJIF): 3.134

e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 2,Issue 10, October -2015

Personalized Query Recommendationand Exploration of theDatabase

Sujata A. Pardeshi[a], Suresh. K. Shirgave[b]

[a] *M. E. Research Scholar, D. Y. Patil College of Engineering & Technology, Kolhapur, India.*
[b]*Head of Department, Information Technology,D.K.T.E Society's Textile & Engineering Institute, Ichalkaranji, India.*

Abstract*: The business user always approached to the database to explore the databy forming SQL queries where they faces the difficulties as they have not proficiency in the domain of structured query language. To assist such users, the proposed system proposes the query recommendation framework that generates personalized recommended list of query objects in context of user and query characteristics domain. In proposed framework, the active user session is captured based on their earlier recorded static preferences from which an individual user profile is formed. As the number of usersare logs to the system, based on their working style behaviour and recorded common preferences, the certain group profiles are formed & used in the recommendation process. The similarity between active user session and the past users sessions is identified in the context of query fragments by using the cosine and adjusted cosine similarity metrics that results in the fragment to fragment similarity matrix. The similarity matrix is used to form the predicted summary vector which is used to form the recommendation seed and used to recommend the top N queries to the current user. The comparative evaluation is made between the cosine similarity metric and adjusted cosine similarity metric. Hence, the proposed work is strongly concentrated in recommending the list of query objects in all interest of user directions by providing the interactive assistant tool to the user to get the desired result over the large database.*

Keywords*:Query Fragments, Similarity Metric, SQL Query Object, Query Recommendation, Recommendation Seed.*

I. INTRODUCTION

The rapid development in the business world across different verticals results in generation of huge amounts of data to gain the business intelligence. Such data and information is organized, analyzed and acquired with the help of Database Management Systems which are very large and complex in nature. In business intelligence framework, the different technical and non-technical users always interacted with the applications to get the desired summary level and analytical level results. These applications are not so helpful and supportive for the users to get the desired result as per their interest and in such situations users are interested in the use of their ways to retrieve the required result. To get this, it is required for the user to design and post their requested question/query in the form of the query language and most of the timethe userfaces the difficulties, since the database applications contain multiple, complex underlying schemas, multiple attributes and other database functionality. However, the design of required queries depends on the user's manual editing of the query characteristics and if the user is not familiar with the database schema then multiple underlying attributes would confuse to them. The technical user which has the database knowledge also faces the difficultieswhile issuingthe complex queries over the large datasets to acquire the required knowledge. As a result, different ways are provided so that the user will interact with the database by formulating their queries that results inthe list of top *N* recommended queries [1] [10] [11][12] and this supportive framework is called as the Query Recommendations.

The query recommendation technique is a way of Information Retrieval based on the well-established recommendation algorithms which is used in most of the e-commerce web applications. The different recommendation algorithms have been used by Amazon.com to generate the personalized product list for the users for online stores. In October 2006, the competition was organized by the Netflix to generate top *N* movie recommendations from the dataset was containing 100 million anonymous movie rating and made challenges to the researchers which started the journey towards developing the recommendation system to generate predictions about the movie ratings as per peoples likings. From this competition, the company got a better recommendation system that was supposed to improve the user satisfaction and provides support in improving the predictions by identifying the customer needs. The use of such web based recommendation system that motivates the researchers and developers to make the use of such recommendation algorithms in the context of SQL Query Recommendation where it plays a vital role in database application domain which is used to provide the assistant tool [3] to assist the users in formulating and refining the query to get personalized information retrieval across the huge dataset.

To assist the user, the proposed system provides the form based interface to formulate the SQL query which is represented with the recommendation of top *N* query objects. The proposed framework is built by using the recommendation approach used by the web recommender applications, in which the items purchased by the group of customers would be recommended to the other customers based on their likings. The recommendation approach used in the web recommendation applications generates the prospective in database exploration activity. The user performs the

exploration of database by means of placing a question by using the declarative structured query language (SQL). As one user formulates a SQL query to explore the database may be useful to other users in future.However, it may possible to store the SQL queries posed by the users while exploring the database and suggest some of them to the other users by identifying the similarity among the querying behavior of the users.The suggestions of queries resulted in recommendation of query objects which is represented to the user to easily explore the database to get the desired result. Therefore, the relaxation is offered to the users who are not familiar with the SQL language. In web recommendation applications, the similarities among the customers are identified based on their rate/visit/purchase history. The web recommendation approach cannot be directly applicable in the context of query recommendation as the two SQL queries may vary in their syntax but generates the same result. This feature of the declarative query language complicates the process of identifying the similarities among the two users. Theone of the important and challenging task in query recommendation domain liesin the identification of specific query objects that plays important role in identifying the similarities among the users.The two challenges are solved by using the closed loop approach, in which the proposed framework decomposes each input query inthe basic components that captures the principle logic of the parsed query. These basic components are called as query fragments, and are used to compute the similarities among the user. The recommendations of query objects are generated by choosing the remarkable queries from the query log which are well matched with the logical behavior of user queries.

The rest of the paper is organized as follows: section 2 describes the related work, the section 3 concentrates on the conceptual framework of the proposed system, the section 4 describes the implementation details of the system, and the section 5 specifies theexperimental evaluation to be done to evaluate the proposed system in respect of the similarity metric. The section 6 concludes the system with the future work.

II. RELATED WORK

A. QueRIE: Collaborative database exploration

MagdaliniEririnaki, Suju Abraham, Neoklis Polyzotis, and Naushine Shaikh [1] describes the assistant tool for the users to formulate the query and retrieve the scientific dataset with scientific database like Genome and Sky Server required in their study by using item to item collaborative recommendation method.The authors have designed the query recommendation framework for the scientific users, called as QueRIE which is used to form the SQL querieswhile exploring the database and at the same time, recommends the list of top *N*personalized SQL queries to the current user.In web recommendation system, item to item similarity method is applied to find out the similarity between the items. In query recommendation, the items which are exist in terms of the fragments of SQL queries are used to find out the similarities between the SQL queries posed by the different users in the recommendation process. It computes the offline similarities among the query fragments of the past users recorded in the query logs with the query fragments involved in the session summary of the current user.

Afterwards, the similarities among the query fragments are used to predict the rank of each fragment with respect to the active user session and the highest ranked query fragments are used to mine the query logs. Then the top *N*remarkable candidate queries are extracted from the query log and are recommended to the user. The precision and recall method is used for evaluating a recommendation framework.The authors have applied user-based as well as item-based collaborative filtering methods and there is need to design the more generic and scalable recommendation framework. To design generic framework, the proposed system identifies the relationship between user and item with respect to characteristics of items. This is done with the support of matrix factorization method and applied in the domain of query recommendation.

B. A Learning Approach to SQL Query Results Ranking Using Skyline and User's Current Navigational Behavior

The *Zhiyuan Chen, Tao Li, and Yanan Sun* [2] has observed that, the user preferences are heterogeneous in nature and always have insufficient data about it since the user preferences are collected by means of users navigational behaviour. The user viewed the resultant dataset in the form of cluster view or by means of automatic categorization techniques and then user performed navigational action, such as click and rank, click and expandto get interested resultant dataset. The chosen query is sent for processing to which informative label is allocated to set the priority about the chosen resultant dataset and will beused in future while generating recommendations.To overcome the probability about the user preference data, in the proposed work, the preference profile for the users are formed and on top of such user profile layer the specific group profiles are generated to extend recommendations in user domain as per their working styles.

C. A Survey of Collaborative Filtering Techniques: Review Article

The *Xiaoyuan Su, Taghi, and M. Khoshgoftarr* [3] have introduced the collaborative filtering (CF) technique, challenges and categories of the collaborative filtering techniques: memory-based, model based and hybrid CF algorithms. The different approaches of collaborative filtering are described which are used to identify the similar items.The survey reviewedthe use of similarity metrics in the context of collaborative filtering techniques.

D. Matrix factorization Techniques For Recommender Systems: A Survey

The *Yehuda Koren, Robert Bell, and Chris Volinsky* [4] used matrix factorization method which becomes a leading methodology within collaborative filtering recommendation approach and it is superior to the classical nearest

neighborhood technique. This method also offers a compact memory efficient model that is useful in collecting feedback in multiple forms.The review of this article states that, to generate the efficient recommendations it may possible to make the use of matrix factorization method.

III. CONCEPTUAL VIEW OF THE PROPOSED SYSTEM

A. Problem Statement

The users approached to the Database Management Systems to discover the knowledge by forming the SQL queries with the help of declarative SQL query language where they faces the difficulties in the formulation of queries. This motivates to design and built the assistant tool by using the well-known web recommendation tools and techniques that enables the user to explore the database interactively to get the personalized recommendation of top N query objects.

B. Conceptual Framework of the Proposed System

The user explores the database to discover the required data by formulating the SQL query and the exploration activity is taken as a visit to the database. During a visit, suppose the user posed a sequence of SQL queries Q_i as shown in Example 1, to explore the underlying database with the aim of discovering the interesting information. In this scenario,the user takes a view of the resultant dataset of the queries one by one which are posed in a sequence and based on that, the user can formulate the next query to get the desired result. For example, the following session shows the sequence of queries:

Example 1:

Query 1: SELECT PPLAYERID, PFIRSTNAME, PLASTNAME, PYEAR, PTEAM FROM PLAYER_PLAYOFFS

Query 2: SELECT PPLAYERID, PFIRSTNAME, PLASTNAME, PYEAR, PTEAM FROM PLAYER_PLAYOFFS WHERE PTEAM='PHI'

Query 3: SELECT PPLAYERID, PFIRSTNAME, PLASTNAME, PYEAR, PTEAM FROM PLAYER_PLAYOFFS WHERE PYEAR=2008 AND PTEAM='PHI'

The Example 1 shows that, the user will be able to form a third query, based on the query 1 and 2 to get the desired result. The execution pattern of the queries clearly specifies that the user is not familiar with the underlying structure of the database and spends lots of time in getting the desired result. Therefore, it may possible to save the time while exploring the database if the appropriate query is directly recommended to the user immediately after the first attempt is made in his/her visit. This will happen if the similar query sessions which are already existing in the query logare used in the recommendation process as the candidate queries.This leads to the design and development of the framework that not only assist to the user in formulating the SQL queries but also recommends the existing similar queries to the current user. To do this, we have represents a visit to the database as a session summary S_i. The summary $S_i$take a review of the characteristics of the queries which are posed during the session of the current user in the context of the query components, which are defined the relation, the attributes of the relation, and the aggregate functions. The session summary $S_i$can be represented either based on the query components to be accessed by the user or based on the actual result that is viewed by the user. Therefore, the session summary which is formed for the query components called as the crude summary. Most of the time, the representation of the session summary in both the way plays a vital role in the recommendation process since the user is not familiar with the underlying structure of the database. The crude summarycontains the importance/weight for the query components which is computed based on the number of the queries that have taken the reference of them. The crude summary is used to identify the similarity among the users. The detail summary is formed for the query components to be viewed in resultant dataset.

The main motivation of the proposed system is to extend the theme provided by authors [1] to the business users, since most of the time, the users are not familiar with the structure of the underlying business and cannot do the exploration of database by using the SQL queries. It is observed that, while exploring the database, the requirements of the users are heterogeneous in nature in terms of accessing the query components involved in the SQL queries. With this aim, the proposed system strongly gives the attention towards the formation of a user profile that contains the static preferences of their choices with rating/rank assigned by the user. The preferences are specified as the relations, the attributes, and certain aggregate functions to which the rank is specified by the user. The rating/rank is specified in the form of numerical terms as 1 to 5. The requirements of the users while exploring thedatabase are most of the time similar with each other with respective to the working style of the other users.This observation takes a lead in the proposed system to identify the similarity among the group of users in the context of their working environment and use it in the process of recommendation to diversify the recommendation as per their working style.The study made in terms of the probability of the preferences while exploring the database and identifying the similarities among the group of users with respective of their working style put forwards the architecture diagram of the proposed system, shown in Fig. 1. The conceptual view state that, as a new user is logged to the system, the user is asked for registration and the user static preference profile is formed. If the user is already registered, then the previously recorded static preferences are used in the recommendation process. Based on the user profiles, the certain numbers of group profiles are formed by finding the similarities between the preferences of all logged active users to diversify the recommendations as per their working style. In the proposed system, we use S_i that represents the session summary for the userU_i , where $i = 0$ always

represent the current active user for which the recommendations are generated and $i = 1, \ldots, n$ represents the sessions of the past users. To generate the recommendations for the current user, the predicted summary S^{Pred} is computed that captures the predicted degree of interest S_0^{Pred} for the current user. The predicted degree of interest is measured with respective to the query components which are appearing in the session summary of the current user and also new one which are not used in the queries posed by the current user. The predicted summary S^{Pred} is used as a recommendation seed to generate the recommendations and it is defined as a function shown in Equation 1.

$$f = (\alpha, S_0, Predicted\ Summary\ ) \quad (1)$$

The proposed system provides the recommendations in a variety of ways, with respective of finding the similarities among the sessions of currently logged users and logged users or the group usersorthe past users. Therefore, as then numbers of users are logged to the system, the similarities among them are computed by using the crude summary and it is resulted into the fragment to fragment similarity matrix. The similarity matrix is used to predict the weight for not referenced query components and results into the predicted summary vector.This step puts forward, the user predicated summary vector U^{Pred} used with a recommendation seed to generate the recommendations in the context of the logged users. The same approach is used to identify the similarities among the group profiles that results into the group predicated summary vector G^{Pred} and used with recommendation seed togenerate the recommendations in the context of the working style of the users. The query data of the past users is mined from the query log file and the similarity between the query componentsof the past users is computed through which the predicated degree of interest P^{Pred} is generated. The past user predicted summary P^{Pred} is used with recommendation seed togenerate the recommendations in the context of the past users.

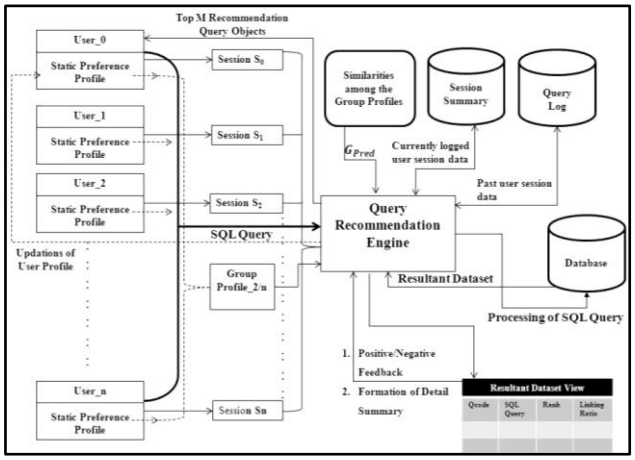

Fig. 1System Architecture of the Recommendation System

The expected outcomes the user predicted summary U^{Pred} , the group predicted summary G^{Pred} and the past user predicted summary P^{Pred} are used in the recommendation process to generate the recommendation for the current user session S_0 in a different of ways. To recommend the top N query objects, a closed loop approach is used in which the proposed framework decomposes each query posed in an active user session S_0 into the query fragments. The query fragments are used to identify the similarities between the users. To start the recommendation, 1) the user session similarity approach 2) the group user session similarity approach 3) the past user session similarity approach are used. The first approach makes the use of active user session S_0 and predicated summary for the logged user U^{Pred} . In the second approach, the active user sessionS_0 and the predicated summary of the group users G^{Pred} is being used. In the third approach, the active user sessionS_0 and the summary of the past users P^{Pred} is used in the recommendation process.To generate the recommendations of top N query objects in the context of the classical item to item collaborative filtering approach, the predicted summary U^{Pred} , G^{Pred} and P^{Pred} are used. The seed of recommendation S^{Pred} is computed asa similarity function based on the predicated degree of interest by using the Equations 2-4.The predicted interest is captured during the session similarity approaches applied for the user sessions, the group user sessions and the past user sessions.

$$f = (\alpha, S_0, U^{Pred}) \quad (2)$$

$$f = (\alpha, S_0, G^{Pred}) \quad (3)$$

$$f = (\alpha, S_0, P^{Pred}) \quad (4)$$

The recommendation is also accountable in the use of the content based filtering approach, where the current active user is interested in computing the rank for queries in their own session to know the first query to be executed in a

sequence of queries to get the desired result. In this case, the seed of recommendation S^{Pred} is computed as similarity functions $f = (\alpha, S_0, Q_i)$. The hybrid recommendation approach is also applied by taking the active user queries and the past user queries to extend the recommendation beyond the preferred classical item to item collaborative recommendation approach. The mixing factor α is introduced as a parameter of the proposed system to mix the current user session summary S_0 with the session summaries of the past users, the logged users, and the group users to tune and generate the recommendations with respective of recommendation types. The recommendation seed, S^{Pred} is used to identify the remarkable candidate queries from the query log file which have the same characteristic as like that ofthe queries posed by the current user in a session summary S_0. The rank is generated for each candidate queries and top N queries are recommended to the current user. The user can choose any recommended query and can take a view of the resultant dataset generated by the Database Management System. After viewing the resultant dataset of a chosen query, the user can provide the feedback on that query. The specified feedback is used in the recommendation process to identify the goodness of the query. The user also has a facility to store the recommended query in a query template and refereed it in future while exploring the database. We have used the two approaches for generating the recommendations: 1) fragment based approach 2) matrix factorization based approach. The query recommendation framework consists of the following components: (a) recommendation process (b) selection of recommendation approach (c)formation of the session summary model (d) computation ofa recommendation seedS^{Pred} and generatingthe recommendation of top N queries. The summary of notations which are used throughout this paper is included in Table 1.

Table 1

Notation Summary

S_i	Session summary for user i/ User i
S_0	Session summary for current user
Q_i	Set of queries in session S_i
S_Q	Single query vector of query Q.
$S_i[\Phi]$	Importance of fragment Φ in session S_i.
S^{Pred}	Predicted Summary for current user.
U^{Pred}	Predicted Summary for logged users.
G^{Pred}	Predicted Summary for Group Profiles.
P^{Pred}	Predicted Summary for past users.

IV. IMPLEMENTATION DETAILS

A. Recommendation Process: Query Pre-processing and Generalization

In the currentsession S_i, each query is decomposed and pre-processed to from the query fragments. Each query is generalized in which WHERE and HAVING conditions are relaxed and generalized in the form of strings by usingthe approach specified in [13]. For example, the condition "where a1=10" is relaxed as "where a1 EQUI NUM".The relational operators used in a where or a having condition string such as "<=",">=", "=" are replaced by using the strings EQUI and COMPARE. The framework proposed by [1] uses the regular expressions and designates the Start and End parsing keyword to decompose the generalized query inthe query fragments based on the SPJ (SELECT, PROJECT, JOIN) form of a SQL query and the similar approach is applied to decompose the fragments during the decomposition process of a query. The Table 2 shows the designated Start and End parsing keywords.

Table 2 Parsing Keyword

Fragment name	Start keyword	End keyword
Attribute string	SELECT	FROM
Relation string	FROM	WHERE, GROUP BY, ORDER BY, End of query
Where string	WHERE	GROUP BY, ORDER BY, End of query
Group by reference string	GROUP BY	ORDER BY, HAVING, End of query
Having string	HAVING	ORDER BY, End of query

Example 1: SELECT COUNT (PLAYERS.FIRSTNAME) FROM PLAYERS WHERE PLAYERS.FIRSTNAME='JHON M'. The query given in Example 1 has the following fragments and are defined as follows in respect of their fragment category.

Relation String	PLAYERS
Attribute String	COUNT(PLAYERS.FIRSTNAME)
Where String	PLAYERS.FIRSTNAME EQUI STR
Group By Reference String	NONE
Having String	NONE

In the query generalization process, the fragments of WHERE clause is not differentiated based on the actual values but are differentiated based on the attributes used in the filtering conditions. For example, suppose query Q_1 has a where filtering condition "table1.a1>=10" and other query Q_2 has where filtering condition "table1.a1>=20" are represented by the same fragment as "table1.a1 COMPARE NUM". Therefore, during the recommendation process, both Q_1 and Q_2 queries are considered as the similar queries and are recommended to the user. Hence it is possible to recommend the queries which are syntactically different, but possess the similar semantics. Afterwards, the fragment data is used to form the query profile and the session vector $S_i[\Phi]$. The query profile along with the fragment data is stored in the query log file.For the new fragment which is not previously recorded in the fragment data file, the new string identifier with respect to the fragment category is assigned to it and it is stored in the fragment data file. The structure of query log file is designed by considering approach which was previously studied to clean the Sky Server Database [5].

B. **Recommendation Approach**

FragmentBased Recommendation Approach

The item-based collaborative filtering approach is used to generate the recommendations, in which the similarities among the query fragments is identified. The current user session summary Si is represented as a weighted vector of top k query fragments. . The user-item matrix is made of user sessions containing the query objects and the top k fragments. The rates of items are represented as the count of fragments with respect to the query objects. For recommending the queries in the context of the past user predicted summary, the fragments which are co-appear in many several queries posed by the past users in their session summariesare identified to form the user-item matrix. For recommending the queries in the context of the logged users, the fragments which are involved in the user predicted summary are used to form the user-item matrix.For recommending the queries in the context of the group users, the fragments which are involved in the group predicted summary are used to form the user-item matrix.The user-item matrix is used to form the fragment to fragment similarity matrix.

Matrix Factorization Based Recommendation Approach

In this technique, the user-item matrix is taken that contains the rating/importance of the query fragments assigned by the users. The singular value decomposition (SVD) method [9] is used to identify the hidden rating of query fragments of the user-item matrix. In this technique, the user-item matrix is factored into the user-feature, item-feature, and a singular values matrix which are defined as U, $S^{1/2}$ and V^T respectively. Afterwards, it computes the compound matrices X and Y to represent the user model and the item model respectively by using following Equations 5 –6.

$$X = U\ X\ S^{1/2} \tag{5}$$

$$Y = S^{1/2}\ X\ V^T \tag{6}$$

To predict the rate of particular unknown item of the user-item rating matrix, the dot-product between matrices X and Y is taken by using the Equation 7. Afterwards, the user-item matrix with the known rating for each item is prepared and it is taken to form the fragment to fragment similarity matrix.The user-item matrix is used to form the fragment to fragment similarity matrix. Thefragment to fragment similarity matrix is used to predict the importance/rank for each fragment which is not involved in the session summary S_0 of the current user.

$$\hat{r} = X_u\ * Y_i \tag{7}$$

C. **Formation of Session Summary Model**

The current user session summary Si for the user U_i is represented as the weighted vector of top k query fragmentΦwhose cell is denoted as $S_i[\Phi]$. The cell contains the non-zero weight for the fragment Φif it isappeared in at least any one of the query of the current user session S_i. The weight of the query fragment Φ in S_i is computed by using the weighted scheme. In this approach, the weight is computed as a sum of weight of the query fragment $\Phi$with respective tothe query objects $Q_1 \dots Q_n \in Q_i$involved in a single session. The fragments that appear more than once in a single user session will receive a higher weight than others. It is formulated as follows:

$$Si = \sum_{Q \in Qi} S_{\mathrm{Q}} \tag{8}$$

However, the weighted vector Si contains the top N query fragments which may or may not involve in the queries$Q_1 \dots Q_n$ posed by the current user. The query vector S_Q is constructed as a single query vector forthe queries$Q_1 \dots Q_n \in Q_i$by summing up the weight of the witnessed fragment $\Phi \in Q_i \dots Q_n$.Thequery vector$S_Q[\Phi]$contains the not witnessed query fragments for which the weight is predicted by using the predicated mathematical model. The predicated mathematicalmodel is shown in Equation 9.

$$P_{u,i} = \frac{\sum_{N \in similarTo\ (i)} (S_{i,N} * R_{u,N})}{\sum_{N \in similarTo\ (i)} (|S_i, N|)} \tag{9}$$

D. **Computation of Recommendation Seed-S^{Pred}**

To predict the weight for not witnessed query fragments of $\mathcal{S}_Q[\Phi]$, the fragment to fragment similarity matrix, $|F|X|F|$ is formed as a similarity function, defined as $sim\ \ (p,\Phi)$where $p,\Phi \in F$, that identifies the similarities among the fragments.The similarity matrix is formed by using the session similarity approaches specified by the users, discussed in the Section 3. The computed similarity score is used as the input data in predicting the recommendation seed. We have employed the cosine similarity metric and the adjusted cosine similarity in the context of the weighted method to form the fragment to fragment similarity matrix. Finally, the recommendation seed, defined as $\mathcal{S}_0^{Pred}$ is computed. It is the predicted summary vector that contains the importance/weight of each of the query fragmentwhich is not involved in the active user's session summary $\mathcal{S}_0$. The weight of each coordinate of $\mathcal{S}_0^{Pred}\ \ [\Phi]$ is estimated with regard to the active user's session summary $\mathcal{S}_0$ and the top *k* similar fragments involved in the fragment to fragment similarity matrix by using Equation 10.

$$\mathcal{S}_0^{Pred}\ \ [\Phi] = \frac{\sum_{p\in R}\mathcal{S}_0[p] * sim\ \ (p,\Phi)}{\sum_{p\in R} sim\ \ (p,\Phi)} \tag{10}$$

$$\mathcal{S}_0^{Pred}\ \ = \alpha.\mathcal{S}_0 + (1-\alpha).\sum_{i=1,\cdots,\cdots n} sim\ \ (\mathcal{S}_i,\mathcal{S}_0)\mathcal{S}_i \tag{11}$$

In Equation 11, R represents the set of the top *k* similar query fragments, $\mathcal{S}_0[p]$represents the vector of fragments which are rated by the current active user. The query fragmentΦ obtains a higher weight if the session summary $\mathcal{S}_0$of the current user contains the fragments that co-occur frequently in the queries of other users.The predicted summary vector$\mathcal{S}_0^{Pred}$ is formed by using the Equation12.The recommendation is tuned with respect to the recommendation types: the content filtering, the classical item to item collaborative filtering and the hybrid recommendation by introducing the mixing factor $\alpha \in [0,1,0.5]$. The recommendation seed $\mathcal{S}_0^{Pred}$ is tuned to generate the content based recommendations by using the value of the mixing factor $\alpha \in 1$ that generates the recommendation of top *N* query objects by considering the queries posed in his/her session summary $\mathcal{S}i$ .The classical item to item collaborative filtering is followed by using $\alpha \in 0$ and the hybrid recommendation approach is used by means of setting the value of mixing factor $\alpha \in 0.5$. The step ofthe formation of the predicted summary vector is followed by choosing of the highly weighted fragment $\emptyset_{1\ldots n} \in \mathcal{S}_Q$which is used to remark and extract the highly correlated candidate queries from the query log file which are posed by the past users. To generate and represent the top *N* recommended query objects to the current user, the rank is computed for the remarkable candidate queries. The rank is computed by applying the similarity metric between the predicted summary vector $\mathcal{S}_0^{Pred}$ and each of the candidate query extracted from the query log file. Therefore, aftergenerating the rank to each remarkable candidate query, the highly ranked top *N*candidate queries are returned to the useras the set of recommendation of queries with the liking ratio of the other users. The current user chose a query of their interest and sent to the processing to the database system to get the desired result. Afterwards, the user stores the query in a query template to take a reference of it in future while exploring the underlying database.

V. Experimental Evaluation

To evaluate the proposed system, the PlayersDB dataset is used. The experiment represents an evaluation of the similarity metrics used to identify the similarities between the query fragments that results in the fragment to fragment similarity matrix in the context of the fragment based approach and the matrix factorization approach. The holdout set methodology is used, in which the dataset is divided into two disjoint sets, the training set and the test set. The pairwise similarity among query component is computed against the training set.In the test set, each user session is divided as an active user queries and unseen queries. The active user queries from the test set and the pre-computed similarity among the query components is used to generate a set of recommended query objects. Afterwards, the recommended query objects are compared with the unseen queries of the test set. The precision and recall metric is used to measure the effectiveness of the similarity metrics. The precision, recall and F_{score} is computed for each query by using the Equations 12 – 13.

$$Precision\ \ = |Q_r \cap Q_u|/|Q_r| \tag{12}$$

$$Recall\ \ = |Q_r \cap Q_u|/|Q_u| \tag{13}$$

$$F_{score}\ \ = 2 * Precision\ \ * Recall\ \ / Precision\ \ + Recall \tag{14}$$

The Q_r and Q_u in Equations 12 – 14, represents the query components of the recommended queries and the unseen queries. To evaluate the similarity metrics, the following parameters are taken with their default values and are defined in Table 3.

TABLE 3
Parameters with the Default Values

Top *K*query fragments	5
Size of recommendation, top *m*	5
Mixing factor	$\alpha \in [0, 0.5]$

A. Evaluation of Similarity Metrics: Fragment Based Recommendation Approach

The proposed system uses the cosine similarity metric and adjusted cosine similarity to identify the similarities among the query fragments that results into the fragment to fragment similarity metric. In the experiment, the evaluation is made for the similarity metrics by using the value of missing factor, $\alpha \in 0$to evaluate the system in the context of the classical item to item collaborative filtering approach. The Fig.2 and Fig.3 shows the effect of the choice of similarity metric in the perspective of the item to itemcollaborative filtering on the average precision and F-score for the recommendations.The proposed system achieves the precision of 0.76 and average F-score of 0.72 for the cosine similarity metric with a weighted method for $\alpha \in 0$. The proposed system archives the precision of 0.74 and average F-score of 0.65 for the adjusted cosine similarity.

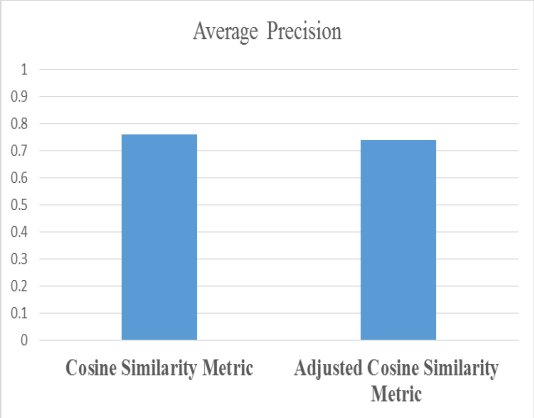

Fig.2 Average precision for similarity metrics, $\alpha \in 0$

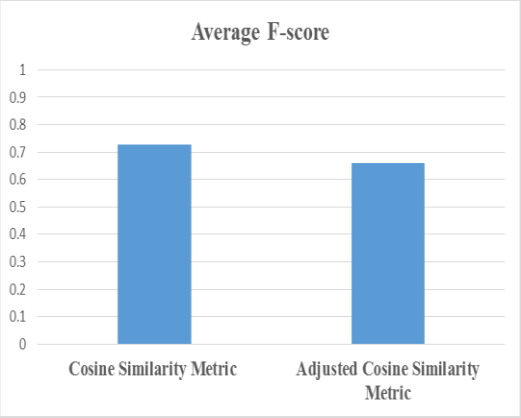

Fig. 3Average f-score for similarity metrics, $\alpha \in 0$

The Fig.4, Fig. 5 shows the effect of the choice of the similarity metric in perspective of the hybrid recommendation approach on the average precision and F-score for the recommendations. The proposed system achieves the precision of 0.83 and the average F-score of 0.82 for cosine similarity metric for $\alpha \in 0.5$. The adjusted cosine similarity metric behaves in the similar way as that of cosine similarity metric with average precision of 0.82 and average F-score of 0.76 for $\alpha \in 0.5$respectively.Therefore, with slight variation in the average precision and average F-score for the cosine similarity metric and the adjusted cosine similarity metric, the cosine similarity metric is chosen as the default similarity metric with $\alpha \in 0.5$.

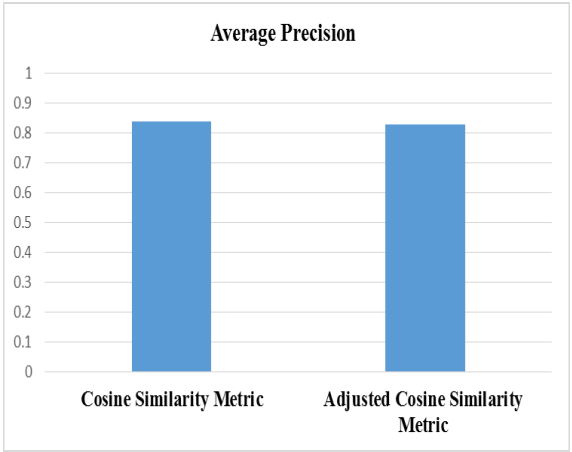

Fig.4 Average precision for similarity metrics, $\alpha \in 0.5$

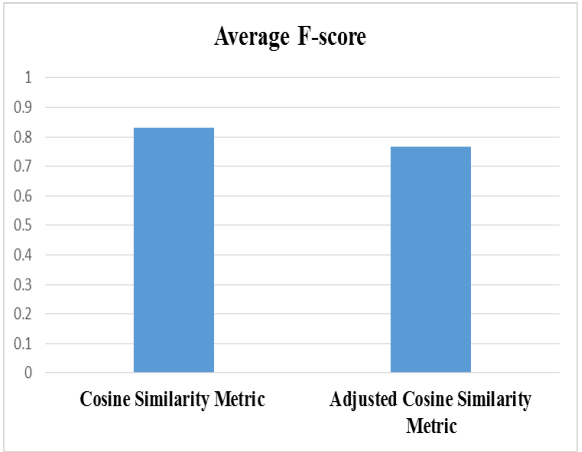

Fig. 5 Average f-score for similarity metrics, $\alpha \in 0.5$

B. Evaluation of Similarity Metrics: Matrix Factorization Based Recommendation Approach

In this set of experiments, the evaluation is made for cosine similarity metric and adjusted cosine similarity metric by using the $\alpha \in 0.5$ in the context of the matrix factorization approach. The Fig.6, Fig. 7 shows that, the effect of the

choice of similarity metric in perspective of values of parameters $\alpha \in 0.5$, $top\ k = 5$, $top\ m = 10$ on the average precision and F-score for the recommendations. This shows that, both the similarity metrics, cosine and adjusted cosine behaves in the same way with average precision of 0.828 and of 0.82 and average f-score is same at the both the points, which is 0.84 and 0.84 respectively.

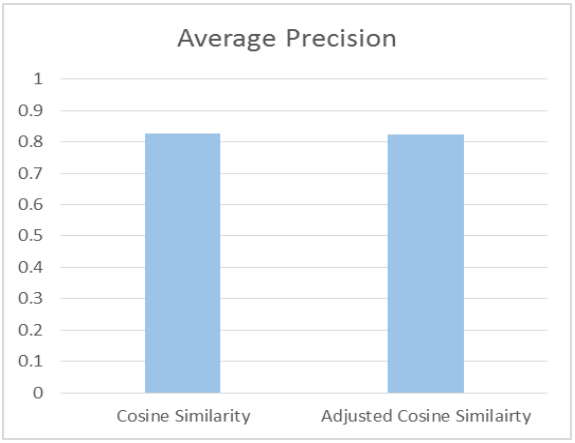

Fig.6 Average precision for similarity metrics, $\alpha \in 0.5$

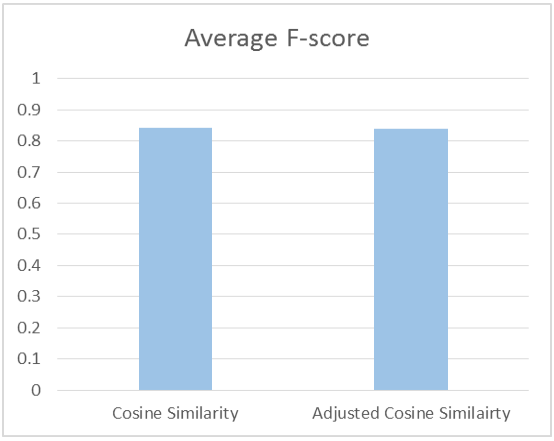

Fig.7 Average f-score for similarity metrics, $\alpha \in 0.5$

C. Evaluation of Similarity Metrics and Discussion

The following observations are identified while evaluating the similarity metrics in the context of the fragment based recommendation approach:

1) With respective of the classical item to item collaborative filtering approach, $\alpha \in 0$the cosine similarity metric and adjusted cosine similarity metrics gives the average precision of [0.76, 0.74] and average F-score of [0.72, 0.65].
2) With respective of the hybrid recommendation approach,$\alpha \in 0.5$the cosine similarity metric and adjusted cosine similarity metrics gives the average precision of [0.83, 0.82] and average F-score of [0.82, 0.76].
3) Therefore, it is observed that, the similarity metrics works better in the hybrid recommendation approach as compared with the classical item to item recommendation approach.The cosine similarity metric gives the better average precision as compared with the adjusted cosine similarity in both the recommendation approaches.

The following observations are identified while evaluating the similarity metrics in the context of the matrix factorization based recommendation approach by using the hybrid recommendation approach$\alpha \in 0.5$.

1) The cosine similarity metric and adjusted cosine similarity metrics gives the average precision of [0.828, 0.82] and average F-score of [0.84, 0.84].
2) Therefore, it is observed that, in matrix factorization recommendation approach both the similarity metrics performs in the similar way.

VI. CONCLUSIONS

The proposed system provides the personalized query recommendation assistant tool through which the database user gets the recommendations of SQL query objects. With the aim of developing the proposed system for the business users, the user needs with regards to the database objects are captured in terms of their static preference profile. By following the path of the web recommendation applications, the proposed system rotates in a variety of angles and generates recommendation in the context of the working style of the users, the logged users and the past users. The recommendation process is followed by viewing of the resultant dataset and analyzing the goodness of recommended query object. The goodness of recommended query in the interpretation of feedback is used to form the knowledge discovery map that represents the estimation of query objects. The item to item collaborative filtering approach is used with their types: 1) classical item to item CF and hybrid CF to identify the similarities among the user sessions in the perspective of query fragments. The fragment to fragment similarities can be identified by using two approaches: 1) fragment based and 2) matrix factorization which generates recommendations very efficiently. The experimental result shows that, the hybrid CF technique works better than that of classical item to item CF technique. The hybrid CF technique enables the system to include the past user sessions with current user session and both the approaches significantly generates efficient recommendations with the use of hybrid CF. The matrix factorization technique is implemented by using the singular value decomposition (SVD) method and the experimental result shows that, it gives better result than that of fragment based approach. This part of the proposed system leads to the consequence towards the upcoming opportunities in the circumstances of use of theSPJ(SELECT, PROJECTION, and JOIN) form of the SQL query objects. By using the SPJ form of a SQL query as the features of it, the matrix factorization techniques will enable to boost the recommendation system to generate the recommendations with respective of SPJ forms. One of the prospects

in the context of query recommendation domain is to use the feedback template to identify the likings of the user in the form of query fragments with regards to the SPJ form of the SQL query.

VII. REFERENCES

[1]Magdalini Eirinaki, Suju Abraham, Neoklis Polyzotis, and Naushine Shaikh, "QueRIE: Collaborative Database Exploration," in IEEE Transaction on Knowledge and Data Engineering, 2013.

[2]Zhiyuan Chen, Tao Li, and Yanan Sun, "A Learning Approach to SQL Query Results Ranking Using Skyline and User's Current Navigational Behavior", in IEEE Transactions on Knowledge And Data Engineering, vol.25, December 2013.

[3]Xiaoyuan Su, 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.

[4]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.

[5]V. Singh, J. Gray, A. Thakar, A. S. Szalay, J. Raddick, B. Boroski, S. Lebedeva, and B. Yanny, " Skyserver traffic report - the first five years," Microsoft Research, Technical Report MSR TR-2006-190, 2006.

[6]H. Field and J. Allan, "Task-Aware Query Recommendation," ACM 978-1-4503-2034-4/13/07, July-August, 2013.

[7]B. M. X. Jin, Y. Zhou, "Task-oriented web user modeling for recommendation," in Proc. of User Modeling 05, 2005.

[8]P. Forbes and M. Zhu, "Content-boosted Matrix Factorization for Recommender Systems: Experiments with Recipe Recommendation," RecSys'11, Oct 23-27 2011.

[9]Ientilucci, E.J., (2003). Using the Singular Value Decomposition". http://www.cis.rit.edu/~ejipci/research.htm

[10]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 (SSDBM '09), 2009.

[11]S. Mittal, J. S. V. Varman, G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "QueRIE: A Recommender System supporting Interactive Database Exploration," in IEEE Intl. Conf. on Data Mining series (ICDM'09) - in ICDM'10 proceedings because of Editor's error, 2009.

[12]J.Akbarnejad, M. Eirinaki, S. KKoshy, D. On, and N. Polyzotis,"SQL QueRIE Recommendations: a query fragment-based approach," 36th Intl. Conf. on Very Large Data Bases, Sept 13-17, 2010.

[13]N. Koudas, C. Li, A. K. H. Tung, and R. Vernica, "Relaxing join and selection queries," in Proc. of the 33nd Intl. Conf. on Very Large DataBases (VLDB '06), 2006, pp. 199–210.

