

RESEARCH ARTICLE



ISSN: 2321-7758

INTELLIGENT QUERY ANALYZER WITH AUTOMATED RECOMMENDATIONS

SINI R¹, LIZMOL STEPHEN²

¹PG Student, Sarabhai Institute of Science and Technology, Thiruvananthapuram, India

²Assistant Professor, Sarabhai Institute of Science and Technology, Thiruvananthapuram, India

Article Received: 31/03/2015

Article Revised on:08/04/2015

Article Accepted on:11/04/2015



ABSTRACT

We propose a new approach for the recommendation that is likely to interest users, the goal is to suggest relevant follow-up queries to a user query. The need for diversification has been studied in the context of query suggestion. The methods in this paper define the objective as generate a relevant suggestion set for given a query. Given a set of suggestions of size N , rerank the suggestions such that the overall utility in top K rank is maximized for all $K \leq N$. In this work, the extraction of relevant details relies on an Entity-Relationship knowledge base. The knowledge base contains an Entity-Relationship graph where nodes are entities and their attributes, and edges are semantic relationships among them. The method builds on a collaborative filtering approach to discover and recommend new suggestions. This includes developing flexible recommender systems which can efficiently compute top- k items within their framework. Recommendation methods try to capture personalized patterns in user feedback data by making assumptions and keeping dense summaries of data. User feedback is typically represented in the form of a sparse matrix that stores existing ratings of users on items. We design efficient top- k algorithms and data structures in order to achieve high scalability.

For this purpose this paper proposes a new approach called fragment based system in which tokenization of string is focused. In this paper, data management systems must provide powerful query management capabilities, from query browsing to automatic query recommendations. Here the session of active user is represented as strings. These recorded strings are used to identify similar query strings by means of string mapping in the previously recorded sessions, which are assembled together to form a complete query for the active user. So the system uses previous information of current user and previous users. Thus this system automates the process of suggesting recommendations. Finally a comparison of fragment based approach to the previously proposed method called tuple based approach is discussed.

Keywords – collaborative filtering, database exploration, personalized and recommended systems

©KY Publications

I. INTRODUCTION

The number of items offered by modern information systems is growing rapidly. Over the past 40 years, database management systems have evolved to provide a sophisticated variety of data management capabilities. At the same time, there is a need for tool for managing queries over the data. People who are not thorough with query making find difficulty to explore which part of database provide useful information. Database exploration is an important task in database mining [3]. Recommender systems help users find their items of interest with little effort, through personalized recommendations. They output the most relevant (top-k) items to the current user by predicting unknown ratings on candidate items. The main idea behind this approach is to calculate a similarity matrix whose entries correspond to pair wise item similarities [11]. Scalability is an important concern, given the ever growing number of users and items in today's recommender systems.

To solve this problem of providing guidance to these users, a QueRIE suggestion system is developed. Here first user queries are fragmented into components for the easiness of calculation of similarities and then reassembles them into simple, understandable meaningful queries. Thus a closed-loop approach is followed [1, 3]. Here also explains personalized query answering in the context of databases. Once item scores are predicted, those with highest predicted scores are served to the user. There has been a great deal of work on efficient algorithms for finding top-k items in various settings. We propose a scalable recommender system based on the fragment based method with an architecture. In the previous work to represent user queries user-based collaborative filtering is used [2]. This paper uses query fragments to represent queries including an overview of tuple based approach. To identify similar components in the previous session similar query strings are used which are reconstructed in interesting queries for active users [1, 2]. A meta query is a query that enables users to locate past queries that match with the previous queries [2]. The users can mine system log to generate recommendations to present queries that match their querying characteristics than to provide with synthetic ones.

The remaining of this paper is organized as follows: Section 2 provides a brief overview of related works of QueRIE system. In section III previous work of query recommendation system is presented, in section IV several major challenges of query recommendation system is presented. Section V discusses an overview of activity diagram of the proposed system in step by step procedure and describes an overview of the abstract framework of the QueRIE system in tuple_based approach and then present the fragment based approach. The experimental evaluation is presented in Section VI and followed by conclusion in section VII.

II. LITERATURE REVIEW

Even though there are recommendations addressed in the Web context, only a few related works exist in the database context. The necessity of a query recommendation framework is emphasized in "A case for a collaborative query management system" [2]. In Context-aware auto completion for SQL, SnipSuggest recommends possible additions to various clauses in the current user's query, and not complete queries [4]. Moreover each query is treated independently of any previous one, even if they belong to the same user session. In "ReDRIVE: Result-driven database exploration through recommendations" a content-based approach is followed [5]. Web databases provide a keyword-based query interface, and suffer from the "empty-answer" and "too-many-answers" problems. In logical foundations of preference queries they show how preferences can be embedded into relational query languages [6] [7].

III. PREVIOUS WORK

In the previous work they developed a tuple_based approach of the framework using user-based similarities to generate recommendations. When the tuple_based instantiation employs approximation techniques (Minhash synopses) to enable real-time calculations, the loss in precision is much greater than that of the fragment-based one [1].

Many relational mining techniques are directly applicable for query mining, especially multi relational mining since queries are likely to be stored in multiple database relations. Related work also is the work on association rule mining. Work that may be adapted for query similarity includes the Context Distance Measure framework designed for computing the distance between two objects

defined across multiple relations. In previous work they discussed the requirements for a collaborative query management system. They outline early system architecture and discuss the many research challenges associated with building such an engine [2].

IV ANALYSIS OF PROBLEM

The traditional DBMS usage patterns are challenging in the area of emerging applications of large-scale scientific data management and industrial data analysis. There is a need for formulating new queries. The traditional trial- and error method is too expensive due to the increasing size of data. A human guidance is costly and unrealistic. Moreover scalability is one of the serious issues for future generation recommendation systems. For the queries posed by the user there is an absence of explicit rating mechanism. There is a problem of how to solve query-equivalence problem.

Challenges:

Managing a collection of queries is more difficult rather than managing ordinary data. To generate personalized efficient query recommendations.

The system must be efficient because it must provide hints and recommendations interactively, as a user types a new query.

Should include an easy-to-use interface and effective visualization methods.

V. PROPOSED WORK

Once a user logs in the system, she is able to select one of the two recommendation engines. The user can author and submit a SQL query. QueRIE sends the request to the database, and presents the user with the results. At the same time, the system records the active user's queries, creating an implicit user profile. This user profile is used as input to the algorithm, along with the predictive model to generate real-time, personalized query recommendations. For each recommended query, the user is able to examine a sample of the results that will be retrieved, in order to decide whether it addresses her needs, prior to actually submitting it to the DBMS.

Taking into consideration the findings of previous work, where developed a tuple_based approach of the framework using user-based similarities to generate recommendations, here decided to follow an item-based approach using query fragments to

represent user sessions. The fragment-based approach can be implemented very efficiently, the space of fragments grows slowly, the summaries are very sparse and, most importantly, the fragment-to-fragment similarities can be computed offline and stored for very fast retrieval when recommendations need to be generated. The dynamic nature of this work is shown below using activity diagram.

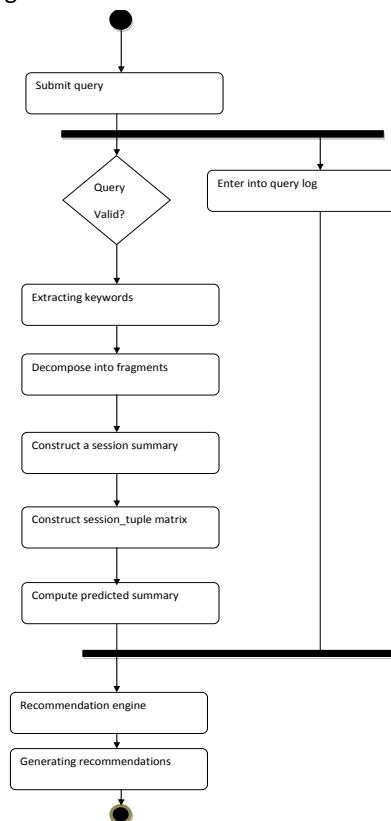


Fig: Activity diagram of query recommendation

METHODOLOGY

The data is divided into two sets, the training set and the test set. The pair-wise fragment similarity is computed against the training set. Each user session in the test set is divided in two parts. One part is treated as the active user's queries, while the second part is treated as future queries. By using the current user's queries from the test set and the previous calculated similarities, QueRIE generates a set of query recommendations. We compare the recommended queries with the unseen queries from the test set

A) Tuple_based recommendation

Here the session summary is represented as a weighted vector, where every coordinate corresponds to a distinct database tuple. The weight of each tuple represents the importance of a given

tuple t in the session of user, and is non-zero only if tuple t is a witness for at least one query in the session. The intuition is that session captures the tuples in the base tables that are touched by the queries in the session.

We assume that the vector represents a single query Q . The value of each element $SQ[t]$ signifies the importance of the tuple t as the witness for Q . We consider two different weighting schemes for setting $S[t]$, a binary scheme and a result-based scheme:

Binary scheme.

In this method two possibilities are there. Either a tuple is present or not. If all tuples are present, they have the same significance.

Result-based scheme.

The idea behind this scheme is similar to the IDF concept from information retrieval: the importance of t is diminished if Q returns many results.

Fragment-based recommendation

The fragment-based approach works similarly as that of tuple-based recommendation, except that the coordinates of the session summaries correspond to fragments of queries instead of witnesses. Here identify as fragments the following syntactical features of the queries in the session: attribute references, table references selection predicates and join. The idea behind this approach is to recommend queries whose syntactical features match the queries of the current user.

Our objective is to identify fragments that co-appear in several queries posed by many users, and guide them by using these fragments in the recommendation process. Formally, session summary S_i is a vector whose cell $S_i[\varphi]$ contains a non-zero weight if the fragment φ appears in at least one query of the session. For a given fragment φ , we define a single query vector cell $SQ[\varphi]$ as a binary variable that represents the presence or absence of φ in a query Q . Then $S_i[\varphi]$ represents the importance of φ in session S_i . We consider two different weighting schemes, a binary scheme and a weighted scheme.

Binary scheme.

In this scheme all fragments have the same weight, regardless of whether they appear in many queries in the session or only one.

Weighted scheme.

In this method each time a fragment appear means it is having higher weight than others.

VI. EXPERIMENTAL RESULT ANALYSIS

In what follows this paper first present an evaluation of the various parameters of the fragment-based approach. This small-scale experimental evaluation, using a subset of the dataset helped me to choose the default values of the system's parameters. Using these, more extensive experiments can be preceded on the entire data set. Then present these results, along with an overview of the results of the tuple based approach [1], in order to compare and discuss the trade-offs of the two instantiations.

VII. CONCLUSION AND FUTURE SCOPE

This paper presents the QueRIE framework that aims to generate useful SQL query recommendations to users of relational databases. Here, we have discussed our vision for the next generation recommender system which has an efficient and scalable item recommendation engine which recommends top- k interesting items, and an efficient package recommendation engine which recommends top- k interesting packages which satisfy all the user specified constraints. Taking into consideration the findings of previous work, a more generic and scalable system is developed. The fragment-based approach can be implemented very efficiently. While the fragment-based approach seems as a straightforward selection, new challenges related to the formulation of session similarity, the synthesis of recommendations and their presentation arise. I would like to measure the impact the query relaxation process has in the quality of recommendations in future.

REFERENCES

- [1] Magdalini Eirinaki, Suju Abraham, Neoklis Polyzotis, and Naushin Shaikh, "QueRIE: Collaborative Database Exploration", IEEE Transactions on knowledge and data engineering, vol. 26, No. 7, July 2014
- [2] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "Collaborative filtering for interactive database exploration," in Proc. 21st Int. Conf. SSDBM, New Orleans, LA, USA, 2009, pp. 3–18.
- [3] S. Mittal, J. S. V. Varman, G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, "QueRIE: A recommender system supporting interactive database exploration," in Proc. IEEE ICDM, Sydney, NSW, Australia, 2010.

-
- [4] N. Khoussainova, Y. Kwon, M. Balazinska, and D. Suciu, "SnipSuggest: Context-aware autocompletion for SQL," *PVLDB*, vol. 4, no. 1, pp. 22–33, 2011.
- [5] M. Drosou and E. Pitoura, "ReDRIVE: Result-driven database exploration through recommendations," in *Proc. CIKM*, Glasgow, U.K., 2011, pp. 1547–1552.
- [6] G. Koutrika, "Personalized DBMS: An elephant in disguise or a chameleon?" *IEEE Data Eng. Bull.*, vol. 34, no. 2, pp. 27–34, Jun. 2011.
- [7] K. Stefanidis, G. Koutrika, and E. Pitoura, "A survey on representation, composition and application of preferences in database systems," *ACM Trans. Database Syst.*, vol. 36, no. 4, Article 19, 2011.
- [8] J. Levandoski, M. Mokbel, and M. E. Khalefa, "FlexPref: A framework for extensible preference evaluation in database systems," in *Proc. IEEE ICDE*, Long Beach, CA, USA, 2010.
- [9] J. Akbarnejad *et al.*, "SQL Query recommendations," *PVLDB*, vol. 3, no. 2, pp. 1597–1600, 2010.
- [10] A. Giacometti, P. Marcel, E. Negre, and A. Soulet, "Query recommendations for OLAP discovery driven analysis," *IJDWM*, vol. 7, no. 2, pp. 1–25, 2011.
- [11] M. Khabbaz and L. Lakshmanan. Toprecs: Top-k algorithms for item-based collaborative filtering. In *EDBT*, 2011.
-