

Survey on Dynamic Query Forms for Database Queries

Deepak Pagaro, Viral Prajapati, Sahil Gambhir, Harish Singh, Mr. Rahul Pawar, Mrs. D. A. Phalke

Department of Computer Engineering

D Y Patil College of Engineering

Akurdi, Pune

Email: deepakpagaro1@gmail.com Email: 8141948373v@gmail.com Email: gambhirsahil20@gmail.com Email: harishbisht29@gmail.com Email: rahulpawar8087@gmail.com, a_dhanashree@yahoo.com

Abstract: The databases used in today's scientific research and web handle huge volumes of data. Such databases contain hundreds or even thousands of complex relations and attributes. The proposed system that implements dynamic query forms for non-relational data. The DQF captures a user's preference and ranks query form components which assists the user in making decisions. Query form generation is an iterative process and initially requires user guidance in the form of feedback. The system automatically generates ranking list of form components, at each iteration and the desired form components are added by the user in Query forms. The ranking of form components depend on the captured user preference. The query results can be viewed at each iteration by the user after filling and submitting the query form. In order to measure the quality of the results generated by the Query form, a probabilistic model has been developed.

Keywords—Key Database, Query Form, Query Execution, User Interaction.

I. INTRODUCTION

Query form is a user interface that is used to query database. Traditional static query forms are created by Database Administrator who has the full knowledge of the underlying database. With rapid development of internet data is becoming very large and complex. Complex databases are created to handle such large data. In social media, such as facebook and twitter, database contains hundreds of entries for user profile. Many web databases like YAGO and Freebase have thousands of data attributes would confuse the user. Therefore, it is very difficult to satisfy user's query needs with a set of static query forms. Existing database development and management tools, such as EasyQuery, have a provision to create customized database queries. However, the creation of it totally depends on user's editing. If a user is not aware of the database schema, those hundreds or thousands of data attributes would confuse the user.

The user interests are obtained through user interactions. User interact with system iteratively. Each iteration has 2 user interactions: Query Form Enrichment and Query Execution. Query Form Enrichment recommends a ranked list of query form component so the user can pick the interested form components. Query Execution user fills and submit current query form and queries, DQF will show result and wait for the feedback from user on the basis of provided query outcome. The form is filled by user and submit to view the query result at each iteration.. So that a query form could be dynamically refined till the user is satisfied with the query results.

This system focuses on creating Dynamic Query Forms which enables users and developers to choose fields of records while querying database even when the underlying structure of the records is not known to user.

The major goal of the system is to enable the user with minimum knowledge of the underlying database of the query syntax and structure of the record to access and modify data present in the records with the help of simple and user friendly query forms. Since, the proposed system is dealing with large volumes of data and a dynamically changing schema, a non-relational database, viz., MongoDB has been used. MongoDB uses documents as a basic storage unit. MongoDB gain its performance mainly by key value design and is easy to scale out.

II. RELATED WORK

Dynamic query form, a query interface which is capable of generating dynamic forms for users. The main objective of DQF is to capture users interest during interaction of user with system and to adapt the query form according to user satisfaction. It begin with basic query form which contain few primary attributes, then it is enriched by an iteration between user and the system until the user is satisfied with the query result [1].

Query form is generally a graphical user interface. Query forms are generated by a data-driven system which finds a set of data attributes which are most likely queried based on data base properties. Then the form is generated based on selected attributes [1].

System automatically generate lots of query form in advance. The user give several keywords to find the query

form from a large number of pre-generated query forms, but it is not appropriate when user does not have keywords to describe the form [5].

DQF provide two level ranked list for providing rank to the generated query form. The first level is rank list for entities. The second level is ranked list for attribute of that particular entity. First describe the rank of each entity's attribute locally and then describe the rank of entities [4].

To choose whether a query form is required or not, not much time is wasted by the user [7]. A cluster of actual instance are represented by each instance in the compressed table. Then to view the comprehensive data instance the user can briefly click through interested cluster. The compressed high level aspect of query results is proposed [8]. Many One pass algorithms have been developed for generating query form [9][10].

The query form are designed according to the users perspective and satisfaction. To measure the quality of query result, two measures are available i.e Recall and Precision [11]. Expected recall and expected precision is used to evaluate the query forms expected performance. Because query forms are capable of generating different query result for different query input and different recall and different precision is evaluated by different queries that produce different query result. The interest of the user is approximated using the users Click-through options on the query results by the query form. Helping non expert users make use of database is a challenging subject. A lot of research work is carried out in designing interfaces which help the users query the databases. QBQ(Query by Query) and Query Form are most promising query interfaces [12]. Currently query forms are commonly used in real world business and scientific systems. Studies and work mainly focus on the process of generating Query forms.

Customized Query Form : Existing database clients provide tools which provide visual interfaces to design query forms like Easy Query, Cold Fusion, SAP and Microsoft Access [3][1]. It helps users to create custom query forms. The draw-back of these aforementioned tools is that they are provided for the professional developers who are familiar with their database and not for the end user[5]. A system which allowed end users to customize the existing query at runtime provided that the end users are fully aware of database schema [6]. If the schema is very huge and complex, which is usually the case it makes it difficult for them to find appropriate entities for the creation of optimized query.

Automatic Static Query : Automatic ways to generate Dynamic Query Forms without user involvement [1][7].

Presented a data driven method [10]. Finding a set of data attributes most likely to be queried on the database, based on the database schema and instances. Forms are thus generated on the selected attributes. System that presents a workload driven approach, it applies clustering algorithm to cluster the historical queries to find the representative queries [4]. The problem with these approaches is, if the schema is complex then user queries could be quite diverse. This means that even if system generate forms prior and in multiple numbers, the users desired queries might not be available in any of those forms. Also the other problem is that if system generate a lots of query forms it will make it difficult for the user to find out the particular form which will give the desired result. solution with the combination of keyword search with query form generation is proposed in [12]. System generates multiple query form beforehand. The user enters the keywords to find matching query forms. It shows good results in databases with rich textual info in data tuples and schema. It does not show good results if the user is not sure about the concrete keywords to describe the query at beginning, especially for numerical attributes. Auto Completion for database queries: New interfaces are developed to help the user to design database queries on the query work load, the distribution and schema of the database [1][6]. Different from our work which concentrates on forms, the queries in their work are SQL keywords.

Query Refinement : It is the technique used by the most information retrieval systems [8]. It suggest new terms about query or changes according to the navigation path of user in search engine. But for the database query form, a database query is a structured and not just a set of terms.

Dynamic Faceted Search : Dynamic faceted search is the type of search engines where relative facets are displayed for the users according to the path traversed in the database [10][3]. Their system is similar to our Dynamic Query Form if it considers only selection components. But apart from selection, there are other query components such as projection. Projection components control the out put of the query and thus are of great importance and cannot be ignored, also selection and projection are mutually influenced.

Database Query Recommendation : Recent work introduced collective approaches to suggest database query components for database exploration [6][9]. Treating SQL queries as items in the collaborative filtering approach. However, it doesn't consider the goodness of query results. A method to recommend an alternative database query based on results of a query[2]. The difference from our work is that, their recommendation is complete query itself whilst our is a component recommendation each iteration.

Dynamic Data Entry Form : An adaptive form system for data entry, which can be changed dynamically according to the previous data input by user[11]. Proposed work is different because it deals with database query forms instead of data-entry forms. Active Feature Probing : It is a probing technique for generating clarification questions to supply appropriate recommendations to users in database search [8]. Different from the work which concentrates on searching the appropriate questions to ask the user ,DQF aims to select query components .

III. SYSTEM ARCHITECTURE

Within the proposed system, first user will fill the query form and this query will be executed by the system and the result is displayed on the screen. Next, the user provide feedback whether the user is satisfied with the query form and if he is not satisfied then there is a option of generating the dynamic query form and ranking is done. At last the user can search the form according to the rank given to the dynamic query form.

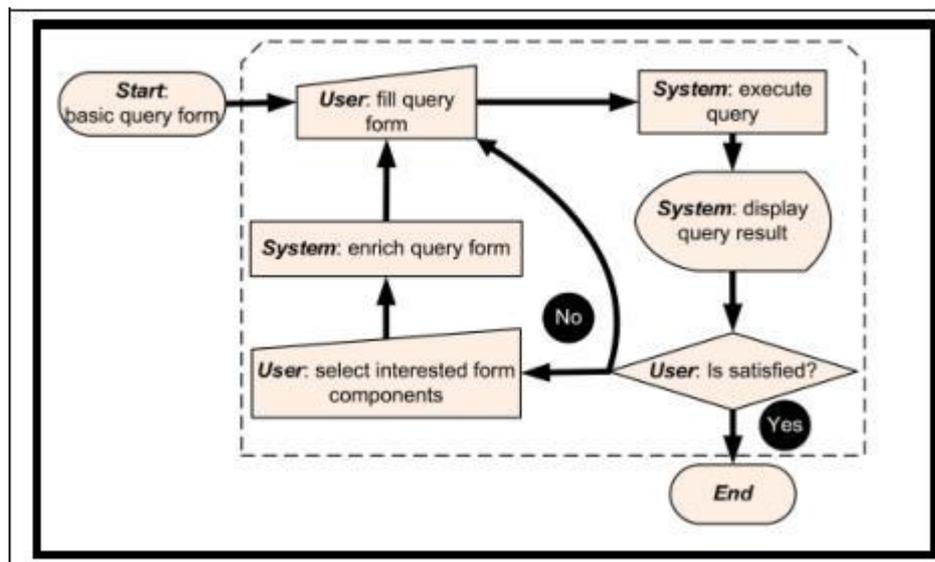


Fig. 1. System Architecture

User and machine directly connect with each other at the time of login and registration for new user. User generate query form and it is executed by the machine and the result is shown to the user. User can generate new dynamic form and then give rank according the user satisfaction.

IV. CONCLUSION

This paper states that DQF system generates query forms at runtime based on the users preference. Based on the number of clicks received by the fields on Query forms and its F-measure score top query form fields are recommended to the user.The NoSQL database can be used to handle large volumes of heterogeneous data along with the DQF system.

ACKNOWLEDGMENT

We express our sincere gratitude to Mrs. M. A. Potey, Head of Department, Computer Engineering, for her assistance, persuasion, and an incent to work better. Our deepest gratitude goes to our project guide, Mrs. D. A. Phalke, for her guidance, ideas, help, encouragement, interpretations and suggestions which helped us in the realization of our

objective and coordinate as a team. We wish to thank Ms. Disha Tiwari and Mr. Rahul Pawar for their constant guidance. We are extremely thankful to our project incharge, Mrs Shanthi Guru for her comments, introspections and support which helped us throughout our work.

REFERENCES

- [1] LiangTang, TaoLi, YexiJiang, and ZhiyuanChen, "Dynamic Query Forms for Database Queries ", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 9, SEPTEMBER 2014.
- [2] M. Jayapandian and H. V. Jagadish ,""Automated creation of a forms-based database query interface". "In Proceedings of the VLDB Endowment, pages 695709, August 2008.
- [3] M. Jayapandian and H. V. Jagadish, "" Automating the design and construction of query forms" ", IEEE TKDE, 21(10):1389 1402,b 2009.
- [4] S. Boriah, V. Chandola, and V. Kumar "", Similarity measures for categorical data: A comparative evaluation" ", SDM, Atlanta, GA, USA, Apr. 2008, pp. 243254.

-
- [5] G. Das and H. Mannila, “[,] ”Context-based similarity measures for categorical databases” ”PKDD, Lyon, France, Sept. 2000, pp, 201210.
 - [6] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, “Combining keyword search and forms for ad hoc querying of databases” ”, ACM SIGMOD, Providence, RI, USA, Jun. 2009, pp. 349360.
 - [7] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum, “[,] Probabilistic information retrieval approach for ranking of database query results ”, ACM Trans. Database Syst. (TODS), 2006.
 - [8] B. Liu and H. V. Jagadish “[,] , Using trees to depict a forest, ”PVLDB, 2009.
 - [9] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, “[,] A framework for clustering evolving data streams ”, In Proceedings of VLDB, 2003.
 - [10]] T. Zhang, R. Ramakrishnan, and M. Livny, “[,] Birch: An efficient data clustering method for very large databases,”in Proceedings of SIGMOD, 1996.
 - [11] G. Salton and M. McGill “[,] , Introduction to Information Retrieval, ”,McGraw-Hill, 1984.
 - [12] M. M. Zloof, “Query-by-example: The invocation and definition of tables and forms, ”in Proc. VLDB, Framingham, MA, USA, Sept 1975, pp. 114