A Survey on Dynamic Query Forms Using Automated Ranking System

Sarita Malunjkar

***Abstract*—Modern databases contains very huge amount of data. Also the databases have exists various complex relationships among them. These databases contains over thousands of relations and their entities. Existing query forms are unable to generate queries dynamically as per the user requirements. Alternative Query Form which takes input in the form of human understandable language and converts it into appropriate SQL query and the query will be fired on relational database to fetch the appropriate data. It is very easy and convenient to ask questions to database in natural language especially to the non-technical users who do not understand the database query language such as SQL. Dynamic query forms are the new database query interface which able to generate query forms dynamically. The primary function of dynamic query forms are- 1) Capture the user preferences and rank those query forms components. 2) Assist user to make their decision.**

Keywords- Query Forms, User Interaction, Query Form Generation, DQF F-measure .

.

I. INTRODUCTION

Usefulness of a database is depends on it's query interface. If a user can not express to the database what the user desire from it, even the richest data store provides little or no value. Static Query forms or predefined query forms are used by the DBA to retrieve the information from the database. But current used databases contain number of attributes and relations. So, retrieving information with static query forms is difficult[1]. Also it is not possible to design static query form with too many attributes to handle. Many database management tools provide mechanisms to design predefined query forms. The process is complex because user must manually edit to design predefined query forms. If a user is unaware of the database schema then handling attributes in the process of designing predefined query forms becomes too complex to handle.

In this paper we proposed query interface approach which generates user interface dynamically. In static query forms getting desired result is one step process but if the database is huge, user end up in getting too many instances of results and hence desired information is unsatisfactory. Proposed approach uses many rounds of actions as input by the user to generate query forms dynamically. Since filtration of results is based on user actions. Process can be extended till satisfactory result or satisfactory number of filtered results can be found. Following figure shows the flowchart of the process. Query forms generated again can be refined according to user feedback and dynamically be changed.

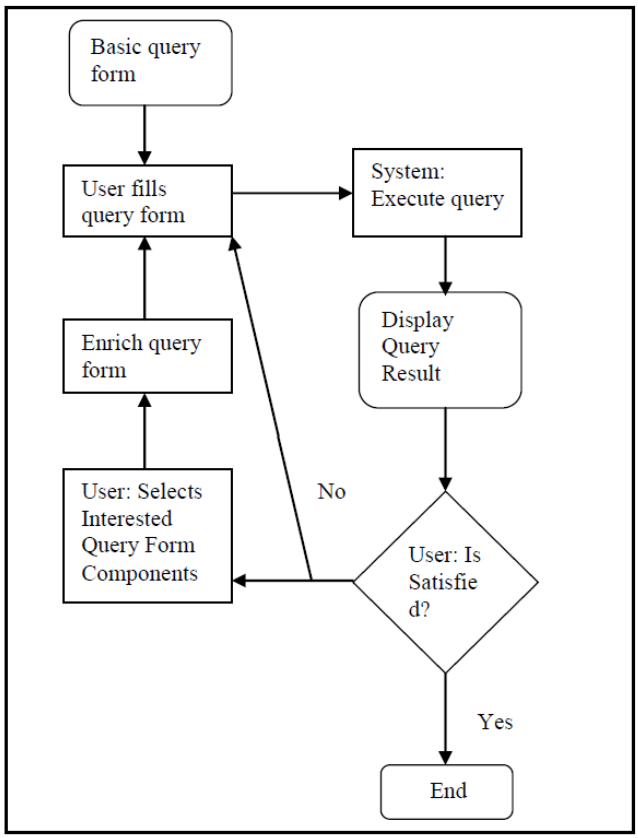

Fig. 1. Flow chart of Dynamic Query Interface

The contribution of this work include :

- Generation of query form for relational as well as for non-relational database.
- Improving dynamic query form by capturing user feedback with a text-box to input some keywords queries.
- Adding personalization so that performance of the system increase.
- Modifying existing system to satisfy queries including or conditions.

II. LITERATURE REVIEW

Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen[1] are propose a Dynamic Query Form system: DQF, a query interface which is capable of dynamically generating query forms for users. Different from traditional document retrieval, users in database retrieval are often willing to perform many

ISSN: 2278 – 7798

rounds of actions before identifying the final candidates The essence of DQF is to capture user interests during user interactions and to adapt the query form iteratively. Each iteration consists of two types of user interactions: Query Form Enrichment and Query Execution.

M. Jayapandian and H. V. Jagadish [2] propose automated creation of a forms-based database query interface The tools provided by the database clients make great eforts to help developers generate the query forms, such as Easy Query [4], Cold Fusion [5] and so on. They provide visual interfaces for developers to create or customize query forms. The problem of those tools is that, they are provided for the professional developers . M. Jayapandian presented a data-driven method . It first finds a set of data attributes, which are most likely queried based on the database schema and data instances. Then, the query forms are generated based on the selected attributes.

M. Jayapandian and H. V. Jagadish [3] presented Automating the design and construction of query forms a workload-driven method .It applies clustering algorithm on historical queries to find the representative queries. The query forms are then generated based on those representative queries. One problem of the aforementioned approaches is that, if generate lots of query forms in advance, there are still user queries that cannot be satisfied by any one of query forms. Another problem is that, when generate a large number of query forms, how to let users find an appropriate query form would be challenging.

K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh, [7] Usher: Improving data quality with dynamic form develops an adaptive forms system for data entry, which can be dynamically changed according to the previous data input by the user. Our work is different as we are dealing with database query forms instead of data-entry forms. demonstrated that probabilistic methodologies can be utilized to outline smart information entry forms that advance high information.

III. EXISTING SYSTEM

With the quick development of net knowledge and scientific databases, modern databases become very large and complex. In natural sciences, like biology and diseases, the databases have over several entities for chemical and biological data resources. many internet databases, like Freebase and DBPedia[3], typically have thousands of structured net entities. Therefore, it's difficult to style a group of static question forms to satisfy varied ad-hoc information queries on those advanced databases. many existing database management and development tools, like easy Query[4], Cold Fusion[5], SAP and Microsoft Access, supply several mechanisms to let users produce customized queries on databases.

A. Limitation of Existing System

However, the creation of customized queries wholly depends on users' manual editing. If a user isn't familiar with the database schema ahead, those a whole bunch or thousands of data attributes would confuse him/her.

IV. IMLEMENTATION DETAILS

A. Proposed Method

To propose a dynamic query form system which generates the query forms according to the users desire.
The main advantage of this system as it provides solution for the query interface in large and complex databases. This paper proposes DQF with NoSQL a novel database query form interface which is able to dynamically generate query forms after each iteration.

The essence of DQF is to capture a users preference at each iteration and rank query form components which allows him/her to make decisions according to the results. The generation of a query form is an iterative process and is guided by the user at runtime. After each iteration system automatically generates ranking lists of form components and the user then adds the desired form components into the query form. Form components are ranked based on the captured user preferences after each iteration.

B. Attribute Ranking metric

DQF provides a two-level ranked list for projection components where first level is the ranked list of entities and second level is the ranked list of attributes in the same entity whereas the selection attributes must be relevant to the current projected entities, if the selection attributes are not related to current projected entities the selection would be meaningless. So DQF system first find out the relevant attributes for creating the selection components. In real world end users are reluctant to provide explicit feedback depend on the results he got.

Now consider given a set of projection attributes A and a universe of selection expressions desired precision and desired recall of a query form F are denoted as PrecisionE(F) and RecallE(F).

Whereas the PrecisionE(F) is defined as the expected number of data instances in the query result that are desired by the user from the total number of instances in the result. And the other parameter RecallE(F) is defined as the expected number of data instances in the query result that are desired by the user from the expected number of instances desired by the user in the whole database. By using this two entities we can calculate the overall performance measure. Expected F-Measure of our system as shown in Equation. The function presented below F-Measure will give the goodness of the query form and thus we can refine the form until it satisfies the user conditions.

$$FScore_E(F) = \frac{(1+\beta^2).Precision_E(F).Recall_E(F)}{\beta^2.Precision_E(F)+Recall_E(F)}$$

Whereas is a constant parameter to control the preference on desired precision or desired recall.

ISSN: 2278 – 7798

After this step FScoreE (Fi+1) is the estimated goodness of the next query form Fi+1. The goal here is to maximize the goodness of the next query form where the form components are ranked in descending order of FScoreE (Fi+1). The score of the next form FscoreE (Fi+1) is obtained by the equation.

$$FScore_E(F) = \frac{(1+\beta^2).Precision_E(F_{i+1}).Recall_E(F_{i+1})}{\beta^2.Precision_E(F_{i+1}) + Recall_E(F_{i+1})}$$

V. CONCLUSION

If database schema is massive and complicated it becomes a tedious work to find attributes and entities. During this case creation of desired query form becomes a busy work. That leads to dynamic query form system. The functioning of this method help user to induce the required results on massive datasets. DQF system generates the query form per user's desire at runtime. Here we tend to use F-measure to estimate the goodness of query form. The metric is acceptable for query form as a result of query forms are designed to help users query the database. Here we are going to capture user preference using both historical queries and run-time feedback such as click through.

REFERENCES

[1] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen "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," Proc. VLDB, vol. 1, no. 1, pp. 695-709, Aug.2008.

[3] M. Jayapandian and H. V. Jagadish, "Automating the design an construction of query forms," IEEE Trans. Know. Data Eng vol. 21, no.10, pp. 1389-1402, Oct. 2009.

[4] Korzh.com. (2005).EasyQuery [Online]. Available: http://devtools.korzh.com/eq/dotnet

[5] Adobe.(1995). Cold Fusion [Online]. Available: http://www.adobe.com/products/coldfusion.

[6] W. B. Frakes and R. A. Baeza-Yates, Information Retrieval: Data Structures and Algorithms. Englewood Cli_s, NJ, USA: Prentice- Hall,1992.

[7] K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh, "Usher:Improving data quality with dynamic forms," in Proc. ICDE, Long Beach, CA, USA, Mar. 2010, pp. 321-332.

[8] S. Zhu, T. Li, Z. Chen, D. Wang, and Y. Gong, "Dynamic active probing of helpdesk databases," Proc. VLDB, vol. 1, no. 1, pp. 748-760, Aug. 2008.

[9] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton, "Combining keyword search and forms for ad hoc querying of databases," in Proc.ACM SIGMOD, Providence, RI, USA, Jun. 2009, pp. 349-360.

[10] G. Salton and M. McGill, Introduction to Modern Information Retrieval. New York, NY, USA:

Sarita Malunjkar received the B.E. Degree in Information Technology in 2013, From BVCOEW , Pune. Now, Pursuing M.E. Degree in Computer Engineering from KJ's College of Engineering & Management Research, Pune in current academic year 2015-16..

ISSN: 2278 – 7798