A Survey Paper on Dynamic Query Forms for A Database Queries

Prof. Swati Nikam
Department of Computer Engineering
DYPIET,Pimpri, Maharashtra, India

Ankita Shaha
Department of Information Technology Engineering
DYPIET,Pimpri,Maharashtra, India

Nikita Khurpe
Department of Information Technology Engineering
DYPIET,Pimpri,Maharashtra, India

Shivani Aher
Department of Information Technology Engineering
DYPIET,Pimpri,Maharashtra, India

Abstract- Database which we are using now days those are web databases and scientific databases which contain heterogeneous and huge data. Such databases contain large no. of relations and attributes. Predefined old question structures are not ready to fulfill different queries from clients on those databases. So to fulfill client requirements new query form generated and that query form is known as Dynamic Query Forms (DQF). DQF is an interface that can create query dynamically. The utilization of DQF is to catch a client's preference and rank query form component, and help the client to make decision. The query form is an iterative process and which is guided by the client. The query form generates the rank list based on the positioning arrangement of structure parts and the client then includes the question structure. Based on the clients preference ranking of form components can be done. Client can fill the question frame and submit query form to see the query result. A query form structure could be powerfully refined till the client fulfills with the question results. In proposed system some DQF techniques are used like Query By Example, Ranking of database, form based interface.

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

I. INTRODUCTION

Query is used to extract data from database in readable format according to user request. Queries divided into two types:

1. Static Query: means a SQL code written in once in development phase and query structure is known.
2. Dynamic Query: In this database structure is dynamic. Many enterprise applications allow user to customize the way data is store and displayed.

Query structure is highly noticeable amongst the most broadly utilized client interfaces for questioning databases. Old question structures are outlined and predefined by designers or database head in different data administration frameworks. The improvement of web data and conventional databases, cutting edge databases turn out to be extensive and complex. Databases have over several elements. Numerous web databases, for example, Freebase and DBPedia [20] regularly have a huge number of organized web elements along these lines, it is hard to plan an arrangement of static question structures to satisfy different database queries on those intricate databases. The database administration like Easy Query, Cold Fusion and advancement instruments for example, SAP and Microsoft Access, give a few components to let clients make redid questions on databases. The production of tweaked questions thoroughly relies on upon client's manual altering. Hundreds and thousands of data attribute will confuse the client if they are not familiar with database. Dynamic Query Form framework a question interface which is prepared to do progressively creating query form for clients. The utilization of DQF is to catch client intrigues amid client connections. An essential query form structure which dynamically create query form. The query form structure is advanced iteratively by means of the collaborations between the client and our framework until the client is fulfilled by the inquiry results.

Techniques used in the DQF are as follows:

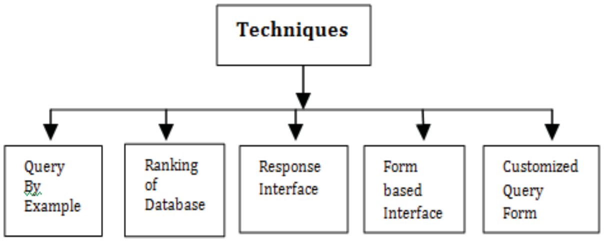

Figure 1: Techniques used in DQF

II. Literature review

This section contains the various literature review work which have been done before and how this research is distinguished from previous research work.

Authors [1] have proposed 'Adaptive technique annotate document'. This system combines content value and querying value for searching. While searching attribute can improve visibility of the document but in this technique not suggest values for identified attributes. In our technique we concentrate on suggesting attribute and values also.

Authors [2] have proposed a system Pay-as-You-Go User Feedback for Data space Systems. This system propose a line of work towards using more expressive queries that leverage annotations is the "pay as you go" is a querying strategy in data space. In data spaces users provide data integration hints at querying time. But this system assumed that data sources already contain structured information and the problem is to match the query attributes with the source attribute. Google Base [12] proposes its own attribute-value pair but these are hard coded pair. But system suggest attribute value pair during form designing. In this integration techniques like PayGo [10] and [2] suggest attribute matching at query time but our system provide at insertion time.

Authors [3] have Usher: Improving Data Quality with Dynamic Forms. This system shows the probabilistic approach can be used to design intelligent data entry forms that promote high data quality. There are multiple steps for data entry: Before entry, ordering of form fields that promotes rapid information capture. During entry, system use the same principle to dynamically adapt the form based on entered values. After entry automatically identify possibly erroneous inputs and re-ask those questions for checking the where actually error is present. The system calculates the error probabilities from whole form and provides how the response provided is erroneous. On the modeling side, our current probabilistic approach assumes that every question is discrete and takes on a series of unrelated values. Such assumptions would make for a richer and potentially more accurate predictive model. This system mainly used for form designing and data quality assurance. Also demonstrate the data quality benefits of each of these components: Question ordering – data entry feedback for question that is it predicates the missing values for incomplete form submission. Re-asking model identifies erroneous responses effectively.

Authors [4] have Automated Creation of a Forms-Based Database Query Interface and Authors [5] have Expressive Query Specification through Form Customization. This system consists of query interface which play main role in determining the usefulness of a database. A form based interface is user-friendly querying method and developed mechanisms to overcome the challenges that limit the usefulness of forms. Also system introduced an algorithm to generate a set of forms automatically given the expected query. A selection query on a SQL database returns all tuples that satisfy the conditions in the query.

Therefore, the following two scenarios are not handled by a SQL system:

a. Empty answers:

When the query is selective then answer may be empty. In this case option of requesting a ranked list of approximately matching tuples without having to specify the ranking function *b. Many answers:*

If query is selective then the result of the query contains many tuples. In such cases the matches automatically ranks more answer tuples and return only the best matches. The result of automated ranking on database is ranking function might be fail because many tuples are tied because of same score. The query forms are generated based on the selected attributes and then clustering algorithm applied on historical queries to find the representative queries. And finally forms are generated based on these representative queries. The problem is that, when user generates a large number of query forms then finding the appropriate and desired query form is challenging. The solution for this problem is that combines keyword search technique with query form generation is proposed in this system. It automatically generates a lot of query forms in advance. The client gives some keywords to find relevant query forms from number of regenerated query forms. This system works when the databases having rich textual information data schemas.

Authors [6] have a system Standing out in a Crowd: Selecting Attributes for Maximum Visibility. Finding ranking function and retrieval of information from database is main part in the query form generation. This system introduced the problem of selecting the best attributes of a new tuple such that tuple will be ranked highly in given a dataset or in query log or both, that is the tuple "stands out in the crowd". Such problems are NP-Complete problem so to solve such problem this system presents extract algorithm based on Integer Programming formulation of the problem also approximation heuristics used in practice. It takes significant amount of time for processing for small workload but system provide optimal and nearest solution.

Authors [7] have a system Random k-LabELsets: An Ensemble Method for Multi label Classification. This system proposes an ensemble method for multi label classification. The Random k-labELsets (RAKEL) algorithm constructs each member of the ensemble as small random subset of labels and learning a single-label classifier are the combination of the subset. The proposed system algorithm aims to take that label correlations using single-label classifiers that are applied on subtasks that manage number of labels and adequate number of examples per label.

This system focuses on effectiveness based on the constraints whose enforcement produces minimum label subset that improve the prediction performance of random selection and it provide theoretical bounds that quantify the probabilities of random selection to produce minimum label subset that meet the proposed construction criteria .The experimental result indicate that the proposed system improve multi label classification accuracy and stability compared with the RAKEL algorithm. Using this we can take into account the correlation between tags for annotations. But in this collaborative annotation is missing.

Authors [8] have a system Towards a Business Continuity Information Network for Rapid Disaster Recovery. This system consist of Crisis Management i.e. the organization that deal with the major event that harm the organization and Disaster Recovery have gained extremely very high importance in the wake of recent man and nature cause a terrible loss. They proposed a solution or model for pre disaster preparation and post disaster business continuity or rapid recovery. In case of disaster system need of rapid information retrieval and sharing of information increases. This system proposed a disaster management model which works well at some extent but it is not considering the effective retrieval of information and system also not support for multimedia elements. Microsoft SharePoint [13] and SAP NetWeaver [14] proposes client to share and search document and also hard coded attributes are also inserted in forms. But CADS improve this using adaptive technique.

A great deal of research works concentrates on database interfaces which help clients to question the social database without SQL. QBE (Query-By-Example) [16] and Query Form are two most generally utilized database questioning interfaces. Current considers and works predominantly concentrate on the most proficient method to create the question shapes.

A. Adjusted Query Form

The devices gave by the database customers make awesome endeavors to offer engineers some assistance with generating the question frames, for example, Easy Query [12], Cold Fusion [11] etc. They give visual interfaces to engineers to make or tweak question frames. The issue of those instruments is that, they are accommodated the expert engineers [13]. [17] Proposed a framework which permits end user to change the current question structure at run time. In the event that the database diagram is vast, it is troublesome for end client to find suitable database substances and characteristics.

B. Computerized Creation of Forms

Authors [13] have exhibited information driven technique. It to begin with finds an arrangement of information properties, which are doubtlessly questioned in light of the database pattern and information examples. The inquiry structures are created taking into account the chose characteristics.

C. Mechanizing the configuration and development of question structures

Authors [18] have introduced a workload-driven strategy. It applies grouping calculation on recorded questions to discover the agent inquiries. The question structures are then created in light of those agent questions. One issue of the previously stated methodologies [13], [18] is that, in the event that we create bunches of inquiry structures ahead of time, there are still client inquiries that can't be fulfilled by any of question structures. Another issue is that, when we produce an extensive number of question shapes, how to let clients locate a suitable question structure would be testing.

D. Joining catchphrase look and shapes

An answer for previously stated methodologies [13], [18] is proposed in [19].It consequently produces a considerable measure of question shapes ahead of time. The client inputs a few catchphrases to discover significant question frames from a substantial number of pre-created question shapes yet it is not suitable when the client does not have solid watchwords to depict the inquiries.

Table 1. Crucial Parameter Table

Parameter	**Dynamic Query**	**Key - value searching**	**Form Interface**	**Data Quality**	**Visibility checking**	**Feedback analysis**
[1]	✓	-	-	-	-	-
[2]	✓	✓	-	-	-	✓
[3]	✓	✓	-	✓	-	✓
[4]	✓	-	✓	-	-	✓
[5]	✓	✓	✓	-	-	✓
[6]	✓	-	-	✓	✓	✓

We found the following issues based on papers:

1. Ranking function might be fail because of basic keyword searching is done.
2. Some Time complexity measures :
   a. Initialization time: Time require after the filling form to the construction of data structure.
   b. Query time: Time requires answering the query.
   c. Insertion time: Time require updating the query form after the feedback given by clients.

III.CONCLUSION

Present day exploratory databases and web databases keep up substantial information. These genuine databases contain over hundreds or even a large number of relations and characteristics. Question structures are not ready to fulfill different impromptu query from clients on those databases. DQF, a novel database question structure interface, used to progressively create inquiry frames. The dynamic inquiry structure era approach which helps clients progressively create question shapes. The key thought is to utilize a probabilistic model to rank structure segments taking into account client inclinations. Positioning of structure parts likewise makes it simpler for clients to modify question structure.

Query interfaces play a essential role in determining The helpfulness of a database. A structure based interface is broadly viewed as the most easy to use questioning technique. In this paper, we have created instruments to succeed the difficulties that point of confinement the handiness of structures, to be specific their prohibitive nature. In this paper we propose an intelligent inquiry structure era approach which makes a difference client to powerfully create question frames. As future work, we will concentrate how our methodology can be stretched out to non relational information. With respect to the future work, we plan to create numerous strategies to catch the client's enthusiasm for the inquiries other than the snap input. Case in point, we can include a content box for clients to enter a few catchphrases questions.

REFERENCES

[1] Eduardo J. Ruiz, Vagelis Hristidis, Panagiotis G. Ipeirotis, “Facilitating Document Annotation using Content and Querying Value”, Vol. 5(6),Issue International Journal of Computer Science and Information Technologies , Pages 8150- 8152 , Year 2014.

[2] S.R. Jeffery, M.J. Franklin, A.Y. Halevy, “Pay-as-You-Go User Feedback for Dataspace Systems”, Issue ACM SIGMOD international conference on Management of data, Pages 847-860, Year 2008.

[3] K. Chen, H. Chen, N. Conway, J.M. Hellerstein, T.S. Parikh, “Usher: Improving Data Quality with Dynamic Forms,” Vol. 23, Issue IEEE Transactions on Knowledge and Data Engineering ,Pages 1138-1153, Year 2010.

[4] M. Jayapandian, H.V. Jagadish , “Automated Creation of a Forms-Based Database Query Interface,” vol. 1, Issue Proceedings of the VLDB Endowment, Page 695-709, Year-2008, http://dx.doi.org/10.1145/1453856.1453932,Aug.

[5] M. Jayapandian, H. Jagadish, “Expressive Query Specification through Form Customization,” Proceedings of the 11th international conference on Extending database technology: Advances in database technology, Page416-427, Year 2008, http://doi.acm.org/10.1145/1353343.1353395.

[6] M. Miah, G. Das, V. Hristidis, H. Mannila, , “Standing out in a Crowd: Selecting Attributes for Maximum Visibility” Issue IEEE 24th International Conference on Data Engineering ,Pages 356-365, Year 2008.

[7] G. Tsoumakas, I. Vlahavas, “Random K-Labelsets: An Ensemble Method for Multilabel Classification,” Issue 18th European Conference on Machine Learning (ECML ’07), page 406-417, Year 2007. http://dx.doi.org/10.1007/978-3-540-74958-5_38.

[8] K. Saleem, S. Luis, Y. Deng, S.-C. Chen, V. Hristidis,T. Li, “Towards a Business Continuity Information Network for Rapid Disaster Recovery,” Issue International Conference Digital Goverment Research , Year 2008.

[9] Microsoft, Microsoft Sharepoint, http://www.microsoft. com/sharepoint/, 2012.

[10] SAP,Sap Content Manager, https://www.sdn.sap.com /irj/sdn/nw-cm, 2011.

[11] ColdFusion. http://www.adobe.com/products/coldfusion

[12] EasyQuery http://devtools.korzh.com/eq/dotnet/.

[13] M. Jayapandian, H. V. Jagadish, , “Automated creation of a formsbased database query interface”, Issue proceedings of the VLDB Endowment, pages 695–709, Year 2008.

[14] S. Agrawal, S. Chaudhuri, G. Das, A. Gionis, “Automated ranking of database query results”. Issue In CIDR, Year 2003.

[15] G. Salton, M. McGill, ,”Introduction to Modern Information Retrieval”, Year 1984.

[16] M. M. Zloof, , “Query-by-example: the invocation and definition of tables and forms”. Issue In Proceedings of VLDB, Framingham, Massachusetts, USA, pages 1–14,Year 1975.

[17] M. Jayapandian, H. V. Jagadish, “Expressive query specification through form customization”, Issue In Proceedings of International Conference on Extending Database Technology (EDBT), pages 416–427, Year 2008.

[18] M. Jayapandian, H. V. Jagadish, ,”Automating the design and construction of query forms”, Vol. 21(10), Issue IEEE Transaction Knowledge And Data Engineering, Page 1389– 1402,Year 2009.

[19] Liang Tang, Tao Li, Yexi Jiang, Zhiyuan Chen, , “Dynamic Query Forms for Database Queries,” Vol. 26, Issue IEEE Transaction On Knowledge and Data Engineering, Page 2166-2178, Year 2013.

[20] DBPedia http://DBPedia.org