Survey Paper on Creation of Dynamic Query Form for Mining Highly Optimized Transactional Databases

Pooja Suthar[1] Ms. Jasmine Jha[2]
[1]M.E Student [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]L. J. Institute of Engineering & Technology, Ahmedabad, Gujarat, India

Abstract— In Scientific databases and web databases maintain large and heterogeneous data. The databases contain over hundreds or thousands of relations and attributes of data. Previously defined old query forms are not able to satisfy various ad-hoc queries from users on those databases. Dynamic query form, a new approach for generating database query form interface is used to dynamically generate query forms. The use of DQF is to capture a user's interests, preferences and rank query form components, assisting user to make decisions without actually typing queries. The dynamic query form is an iterative process and is guided by the user. The system automatically generates dynamic ranking lists of form components and the user then adds the desired form components into the query form. Ranking of form components can be done based on the captured user interests. User can also modify previously generated query forms and by fill the query form and submit queries to see the query result. A query form could be dynamically refined until the user satisfies with the query results.

Key words: Dynamic Query Forms, User Interaction, Query Form Generation

I. Introduction

Query Form is one of the most widely used user interfaces for querying transactional databases. Old query forms are designed and predefined by developers or DBAs in various information management systems. The development of web information and traditional databases lead modern databases become very large and complex. Databases have over hundreds of entities and thousands of tuples. Many web databases, such as Freebase [9] and DBPedia [7] typically have thousands of structured web entities so it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex large databases that are mostly used. The database management and development tools, such as EasyQuery[8], Cold Fusion[6], SAP and Microsoft Access, provide several functionalities to let users create customized queries on databases. The generation of customized queries totally depends on users' manual editing. If a user is not familiar with the database schema those large number of data attributes would confuse the user. In Dynamic Query Form system, a query interface dynamically generates query forms for users. DQF captures user interests during user interactions. The basic query form contains very few primary attributes of the database. The query form is enriched iteratively via refining the interactions between the user and our system until the user is satisfied with the query results.

II. Literature survey

A. *Dynamic Query Forms for Database Queries [1]*

In this paper, Liang Tang et.al presented Dynamic Query Form system (DQF) which is a query interface that is capable of dynamically generating query forms for users. It is Different from traditional document retrieval system where users are often willing to perform number of rounds of operation to get desired result. In this system, DQF captures user interests during user interactions and enrich the query form iteratively.

The main two functions performed in this system was query form enrichment and query execution. DQF first generates basic query form and enrich this form by iteratively via interaction with users. It apply F-measure to find the goodness of a query form that is determined by the query results generated from the query form. Based on this, DQF rank and recommend the potential query form components so that users can refine the query form easily.

The algorithm used for in this paper is:

```
Algorithm: QueryConstruction
–  Data: Q= {Q1, Q2...} is the set of previous queries
   Executed on Fi.
–  Result: Qone is the query of One-Query
–  Begin
–  σone←−0
–  For Q ∈ Q do
–  σone←−σone ∨ σQ
–  Aone←−AFi ∪Ar(Fi)
–  Qone←−GenerateQuery(Aone , σone)
```

It generates the query form and iteratively improves the query form and ranks the components based on suitability to user's interest. It computes the DQF which shows that the execution time grows approximately linearly with respect to the query result size.

B. *Dynamic Query Recommendation for Interactive Database Exploration [2]*

This paper explores techniques that assist users who do not want to use SQL in posing ad hoc structured queries over relational databases. It says that Currently, QF has been utilized in most real-time business or scientific information systems. According to present studies, works mainly focus on how to generate the query forms.

The method is visualizing queries on database instead of querying databases with structure query language. After analyzing and discussing the state of the art for the visualization of temporal intervals and relations, this paper proposes various solutions to the problem of visualizing temporal intervals and their relations for querying databases containing several histories.

This generates the query form where the query mapping is used that shows brief description about the queries that are supported by the form. To build a form for each query template, it used the following standard form components: Label, Drop-down List, Input box, Button.

It does query regeneration if the database schema is large and complex where user queries could be quite diverse. At the end it compares DQF with SQF that suggests that If a query task is covered by one historical queries in history, then SQF built on those historical queries can satisfy that query tasks. The costs of using SQF and DQF to accomplish that task are different.

C. Automating the Design and Construction of Query Forms [3]

In [3] M. Jayapandian et.al explains a form-based query interface, which only requires filling blanks to specify query parameters, is valuable since it helps make data accessible to users with no knowledge of formal query languages or the database schema.

In this paper the form generation technique is to design forms for an entire set of queries and applies form complexity threshold (FCT): a measure of complexity that to control a form's readability.

The algorithm used for generating Query Form :

```
Algorithm GenerateForm
– Input: A queryQ(as an Evaluation Plan)
– Output: A formF
– // Element Construction and Grouping
– Create a new group g for form and add it to the form-tree T;
– Foreach operation o∈Q when traversed top-down do
– Case o is a "selection"
– Create constraint element using the selection predicate;
– Put this constraint-element ing;
– Case o is a "projection"
– Create a result-element using each projected attribute;
– Put these result-elements ing;
– Case o is an "aggregate function"
– Create an aggregate-element using the the group-by attribute, the grouping-basis and the aggregate function;
– Put this aggregate-element ing;
– Case o is a "join"
– Create a join-element using the two (left and right) attributes of the join condition;
– Put this join-element ing;
– Create another new group g'as a child of g in T;
– Set g←g';
– end
– // Element and Group Labeling
– Foreach form-group g∈T do
– Label g relative to its parent group (use absolute path if g is the root);
– Foreach form-element e∈g do
– Labele relative to g;
– End
– end
```

This paper presents a study of system performance using a real query trace, as well as queries from a standard XML benchmark. IT describes an automated self-managing interface-builder will help bring novice users closer to the rich database resources they need to use.

D. Dynamic Query Form With Query Refinement And Database Encryption [4]

In [4], Meenu Joy and ec.al explained "Query by form" that is a simple and intuitive methodology that is frequently used as an entry to database. All attributes from the form along with condition is taken and query is generated, the system executes the generated query by accessing the encrypted database.

By providing the security the usage of the system can be extended to secure database applications. For extending the system security, performance and make the system suitable for various private database applications an encryption is provided to the database. Advanced encryption Standard algorithm (AES) is using here for encryption. In this system the user is provided a keyword search option as an entry to database access.

Main functions explained in this paper are:

1) Customized Query Form:

This allows end-users to customize the existing query form at run time. If the database schema is very large, it is difficult for not-expert users to find appropriate database entities and attributes and to create desired query forms.

2) Auto completion for Database Queries:

Novel user interfaces have been developed to assist the user to focuses on query forms, the queries in their work are in the forms of SQL and keywords.

3) Query Refinement:

Query refinement is a common practical technique that recommends new attributes related to the user interest or modifies the terms according to the navigation path of the user in the search engine.

Experimental results show that the dynamic approach used in this paper leads to secured search, higher success rate and simpler query forms. The ranking of form components also makes it easier for users to customize query forms.

E. Random Query Formulation For Database Queries[5]

In [5], Gopi Krishna Lakkasani and Balakrishna Nayudori explained Random Query Formulation(RQF) that generates query form dynamically and adds a secure captcha to the generated query form to make it more secure. RQF provides two fold securities by using a Security question and captcha by using Object based captcha which is more secure that ordinary text based captcha.

The main functions explained in this paper are Query Generation, CAPTACHA generation, Query Refinement Module, Query Form Ranking Module.

The algorithm used is:

Input: List of documents D[]={d1,d2,………., dn}
Output: Query result documents q[] ={di , …..}

1) For i=1 to n do
2) From the given list of documents, a document di is selected.
3) The selected document contents are splited into tokens(word).
4) The stop words are removed from these tokens and token list is generated as shown below

Ti[] ={ti,,t2,………}

5) The search keyword if compared with the list of tokens.
6) The matching count namely match count is calculated where Match count is the no of matching tokens with the keyword.
7) The match count for each document is stored in a separate array say match[]. A document match count is zero when no single token in the document matches with the keyword.
8) End for
9) From the match[] array remove the documents corresponding to a match count of zero.
10) Based a threshold, the documents are taken from the match count and placed in the Query result array q[]. The threshold can be any integer other than zero. For example if the threshold is 3, all documents in which three tokens are matching will be placed in the resultant query.
11) Finally the document list in the query list is displayed.

The execution results of RQF shows that it provides secured the query form by using security question and CAPTCHA code by asking time locked puzzle security question. It does not submit the query form if the answer is not given within a stipulated time.

III. CONCLUSION

The paper presents a survey on different aspects of the work done till now in the field of Dynamic Query Form generation. The traditional systems focused on static or customized query retrieval techniques for databases. Modern scientific databases and web databases maintain large amount of data. Real-world databases contain over hundreds or even thousands of relations and attributes. Query forms are not able to satisfy various ad-hoc queries from users on those databases. DQF, a novel database query form interface, used to dynamically generate query forms. The dynamic query form generation approach which helps users dynamically generate query forms. This can be achieved by using a probabilistic model to rank form components based on user preferences. Ranking of form components also makes it easier for users to customize query form.

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] Sathu G Rajan, K. Sathya Seelan, ” Dynamic query recommendation for interactive database Exploration”, International Journal of Inventions in Computer Science and Engineering, Volume 1, Issue 3 2014.

[3] Magesh Jayapandian, H. V. Jagadish, ” Automating the Design and Construction of Query Forms”, Proceedings of the 22nd International Conference on Data Engineering (ICDE’06) 8-7695-2570-9/06 2006 IEEE.

[4] Meenu Joy, Bhruguram T M, Shanghai Jiao Tong University, ” Dynamic Query Form with query Refinement and Database encryption” IOSR Journal of Computer Engineering (IOSR-JCE).-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. IV (Sep – Oct. 2014), PP 154-159.

[5] Gopi Krishna Lakkasani, Balakrishna Nayudori, “Random Query Formulation for Database Queries”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 8, August 2014

[6] ColdFusion. http://www.adobe.com/products/coldfusion/. 20/11/2014

[7] DBPedia. http://DBPedia.org. 16/11/2014

[8] EasyQuery. http://devtools.korzh.com/eq/dotnet/. 18/11/2014

[9] Freebase. http://www.freebase.com 8/11/2014