

Dynamic Query Forms for Database Queries

Ramesh Poliseti

M.Tech Student,

Nova College of Engineering and Technology.

Chitti Babulu Sape (B.Tech,M.Tech,Ph.D)

Associate Prof,

Nova College of Engineering and Technology.

Abstract:

Modern scientific databases and web databases maintain large and heterogeneous data. These real-world databases contain over hundreds or even thousands of relations and attributes. Traditional predefined query forms are not able to satisfy various ad-hoc queries from users on those databases. This paper proposes DQF, a novel database query form interface, which is able to dynamically generate query forms. The essence of DQF is to capture a user's preference and rank query form components, assisting him/her to make decisions.

The generation of a query form is an iterative process and is guided by the user. At each iteration, the system automatically generates ranking lists of form components and the user then adds the desired form components into the query form. The ranking of form components is based on the captured user preference. A user can also fill the query form and submit queries to view the query result at each iteration.

In this way, a query form could be dynamically refined till the user satisfies with the query results. We utilize the expected F-measure for measuring the goodness of a query form. A probabilistic model is developed for estimating the goodness of a query form in DQF. Our experimental evaluation and user study demonstrate the effectiveness and efficiency of the system.

Index Terms : Query Form, User Interaction, Query Form Generation,

Introduction:

Query form is one of the most widely used user interfaces for querying databases. Traditional query forms are designed and predefined by developers or DBA in various information management systems. With the rapid development of web information and scientific databases, modern databases become very large and complex.

In natural sciences, such as genomics and diseases, the databases have over hundreds of entities for chemical and biological data resources. Many web databases, such as Freebase and DBpedia, typically have thousands of structured web entities. Therefore, it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex databases.

Many existing database management and development tools, such as Easy Query, Cold Fusion, SAP and Microsoft Access, provide several mechanisms to let users create customized queries on databases. However, the creation of customized queries totally depends on users' manual editing. If a user is not familiar with the database schema in advance, those hundreds or thousands of data attributes would confuse him/her.

Existing System:

Traditional query forms are designed and pre-defined by developers or DBA in various information management systems. With the rapid development of web information and scientific databases, modern databases become very large and complex. Therefore, it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex databases.

Limitations:

- Query forms are designed and pre-defined by developers in information management systems.
- Difficult to design a set of static query forms to satisfy various ad-hoc database queries on complex databases.

Proposed System:

We propose a dynamic query form system which generates the query forms according to the user's desire at run time. The system provides a solution for the query interface in large and complex databases. This paper proposes DQF, a novel database query form interface, which is able to dynamically generate query forms.

The essence of DQF is to capture a user's preference and rank query form components, assisting him/her to make decisions. The generation of a query form is an iterative process and is guided by the user. At each iteration, the system automatically generates ranking lists of form components and the user then adds the desired form components into the query form. The ranking of form components is based on the captured user preference. A user can also fill the query form and submit queries to view the query result at each iteration. In this way, a query form could be dynamically refined till the user satisfies with the query results.

Advantages:

- We propose a dynamic query form generation approach which helps users dynamically generate query forms.
- The dynamic approach often leads to higher success rate and simpler query forms compared with a static approach.
- The ranking of form components also makes it easier for users to customize query forms.

MODULE DESCRIPTION:

The system is proposed to have the following modules along with functional requirements.

1. Query Form Enrichment.
2. Query Execution .
3. Customized Query Form.
4. Database Query Recommendation.

1. Query Form Enrichment:

- 1) DQF recommends a ranked list of query form components to the user.
- 2) The user selects the desired form components into the current query form.

2. Query execution:

- 1) The user fills out the current query form and submits a query.
- 2) DQF executes the query and shows the results.

- 3) The user provides the feedback about the query results.

3. Customized Query Form:

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 who are familiar with their databases, not for end-users. If proposed a system which allows end-users to customize the existing query form at run time. However, an end-user may not be familiar with the database. If the database schema is very large, it is difficult for them to find appropriate database entities and attributes and to create desired query forms.

4. Database Query Recommendation:

Recent studies introduce collaborative approaches to recommend database query components for database exploration. They treat SQL queries as items in the collaborative filtering approach, and recommend similar queries to related users.

Software Requirements:

Technologies : Asp .Net and C#.Net
 Database : MS-QLServer 2005/2008
 IDE : Visual Studio 2008

Hardware Requirements:

Processor : Pentium IV
 RAM : 1GB

Data Dictionary :

A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them. A first step in analyzing a system of objects with which users interact is to identify each object and its relationship to other objects. This process is called data modeling and results in a picture of object relationships. After each data object or item is given a descriptive name, its relationship is described (or it becomes part of some structure that implicitly describes relationship), the type of data (such as text or image or binary value) is described, possible

predefined values are listed, and a brief textual description is provided. This collection can be organized for reference into a book called a data dictionary.

CONCLUSION:

In this paper we propose a dynamic query form generation approach which helps users dynamically generate query forms. The key idea is to use a probabilistic model to rank form components based on user preferences. We capture user preference using both historical queries and run-time feedback such as click through.

Experimental results show that the dynamic approach often leads to higher success rate and simpler query forms compared with a static approach. The ranking of form components also makes it easier for users to customize query forms. As future work, we will study how our approach can be extended to non relational data.

REFERENCES:

- [1] Cold Fusion. <http://www.adobe.com/products/cold-fusion/>.
- [2] DBPedia. <http://DBPedia.org>.
- [3] EasyQuery. <http://devtools.korzh.com/eq/dotnet/>.
- [4] Freebase. <http://www.freebase.com>.
- [5] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In Proceedings of VLDB, pages 81–92, Berlin, Germany, September 2003.
- [6] R. Agrawal, S. Gollapudi, A. Halverson, and S. Jeong. Diversifying search results. In Proceedings of WSDM, pages 5–14, Barcelona, Spain, February 2009.
- [7] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In CIDR, 2003.