International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Optimization Technique for Efficient Dynamic Query Forms with NoSQL

Kavita Ozarkar[1], Rakesh Rajani[2]

[1] PG Student, Department of Computer Engineering,
Alard college of engineering and management, Sivitribai Phule Pune University, Pune, India

[2] Professor, Department of Computer Engineering,
Alard college of engineering and management, Sivitribai Phule Pune University, Pune India

Abstract: ***At present real world databases maintain large and heterogeneous data which contain thousands of attributes and relations. Accessing the information from this corpulent database is a nontrivial task and is an exploring area of interest. Queries are using for database access, but it is a difficult task to the end user those who are not well familiar with query language and illiteracy of underlying schema. This paper presents new system which uses the allowance of query by form methodology that is dynamic query forms (DQF). In which the user is granted with a query form that will help the user to iteratively search the database records and query enhancement is provided to the user by means of feedback or response. Ad_hoc queries can also gratified by using this proposed system with the use of document oriented NoSQL database like MONGODB. MONGODB support dynamic queries that do not require predefined map reduce function. Query enhancement by response is provided by ranking the attribute used in the form. Negative feedback from user can be eliminated to optimize the query form technique.***

Keywords: Dynamic query forms, NoSQL database, ranking query form attributes, negative feedback, F-measure

1. Introduction

Query by form is a simple methodology that is frequently used as an entry to database. Query by form have an advantage that the user need not to worry about how the data is organized in the database and no expertise in query language like sql. Traditional query forms that are designed by developers and DBA's of information management systems. There are various existing systems like Static query forms (SQF) and customized query forms (CQF) have lot of drawbacks as it does not have query refinement and no dynamic nature to solve ad_hoc queries.

The queries which are required to use and express relational database are high level of SQL. But this solution will not works well for many applications, to fully satisfying way of finding the required data. For unknown users these systems are difficult to use and understand, and they require a long time period to understand these systems. So clearly there is a need for easy to use, quick and powerful query methods for accessing database.

In this paper a query interface (DQF with NoSQL) is proposed which is capable of dynamically generating query forms for users. The importance of DQF is to capture user interests during user interactions and to adapt the query form iteratively. Dynamic query form systems were introduced to generate the query forms according to the user's desire at run time. Modern databases become very large and complex and therefore it is very hard to manage using traditional relational database management systems. NoSQL technology is used to solve user problem for those users which are unknown to the database sysetm. NoSQL databases like MongoDB are often highly optimized key–value stores intended for simple retrieval and appending operations.

1.1 SQL Vs NoSQL

From many developers facing problems with relational database, but application developers are increasingly turning to NoSQL databases to meet new challenges. Relational and NoSQL data models [3] are very different. Relational database is a model consist of multiple table which related to each other. Each table consist of row-column structure where row can be a model which represent the multiple information of model like a car. Columns can represents attributes associated with that model for example wheels of car or color of car are the attributes. Relational model uses foreign key concept to refer data from relational table. As compare to this NoSQL databases has a very different model. NoSQL database has a document oriented structure which takes the data and store it into documents in the JSON[4] format. So that one can use JSON document as an object in their application, JSON has key-value structure where using key-value database can be easily accessible.

1.2 NoSQL database

A NoSQL database provides us a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations. Motivation for this approach include that modern databases are very large and complex and needs simplicity of design.

NoSQL and RDBMS are also differ in data structure, NoSQL has key-value, graph or document as a structure while RDBMS has relational tables therefore some operations faster in RDBMS and some in NoSQL. Particular suitability of a given NoSQL DB depends on the problem it must solve.

1.3 MongoDB

MongoDB is an open-source cross platform document oriented database, and leading NoSQL database. This document consists of set key-value pair structure. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and collection documents may hold different types of data.

MongoDB consist of multiple collections, each collection consist of different types of object. Object can be called as document and document can represent in JSON structure. JSON object consist key-value pair structure. The important function of MongoDB is that it supports ad_hoc queries. It also supports search by field value, range queries, regular expression searches. In MongoDB queries can be used to return specific fields of documents and also include user-defined JavaScript functions.

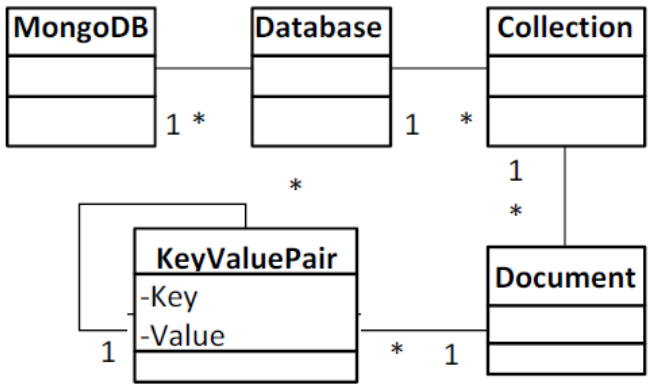

Figure: Model of the MongoDB system

1.4 Negative Feedback

Sometimes user also provides negative feedback [9] which will consists of collection of instances that are not desired by user and instances which are already desired by user in previous iteration query form generation system. Positive feedback is always more informative than negative feedback.

2. Motivation

Existing system uses customized query form where using different mechanism user can create queries and accesses database. Examples of exiting database management and development tools, such as EasyQuery [5], Cold Fusion [6], SAP and Microsoft Access. However, the creation of customized queries totally depends on users' manual editing. If a user is not familiar with the database schema in advance, then user can get confused in thousands of data attributes.

Database systems support a simple Boolean query retrieval model, where a selection query on a SQL database returns all tuples that satisfy the conditions in the query. This often leads to the many answers. When the query is not very selective, too many tuples may be in the answer. In this paper, we formally define the Many-Answers Problem in ranking database query results, and also outline a general architecture of our solution.

The problem found here is how an efficient query form can be designed to boost the user satisfaction in information retrieval. For that purpose NoSql database is used and problem can be solved by designing a good query form with where user can iteratively search to the database until he or she get satisfied with the result at runtime. In this case even unknown user can find the result from the database. Also the query form is provided with a query refinement by means of ranking the attributes of encrypted database. Ranking is done by precision and recall which are measures used for performance evaluation of information retrieval [9].

3. Literature Survey

Existing database clients and tools help developers design and generate the query forms [1]. They provide forms to developers where developer can interact with application to create customize query forms. It provides an interface to user where user can express the desired queries to get desired result. Expressions can be created by filling forms therefore, existing forms are used to do some modification in the expression. End user may not be aware of database then it will be very difficult to form desired query form, as end user may not know the appropriate attributes.

A mechanism to overcome the challenges that limit the usefulness of forms, namely their restrictive nature and the tedious manual effort required to construct them is proposed. An algorithm to generate a set of forms automatically given the expected query workload is introduced. This is workload-driven method [2]. Helps to bring novice users closer to the rich database resources they need to use, and maximize their efficiency with a sizably reduced learning curve. Limitations of this automated form generation is that with large database as there are multiple queries then finding an appropriate query form will be a challenging task. And also user queries could be quite diverse if the database schema is large and complex.

Query recommendation for interactive database exploration [7], here introduce a collaborative approach to recommend database query components for database exploration. They treat SQL queries as item in the collaborative filtering approach and recommend similar queries to related users. One of the problems is that they do not consider the goodness of query result. In proposed system recommendation of user interaction is a query component for each iteration.

Keyword Searching for Querying of Database generates a lot of query forms in advance. In this technique user has to input several keywords using which query can be formed from large number of pre-generated query forms. The system proposes to take as input a target database and then generate and index a set of query forms offline [2]. At query time, a user with a question to be answered issues standard keyword search queries; but instead of returning tuples of result, the system returns forms which are relevant to the keyword entered for search by user.

User can build query using some of above form and submit the query it may give result but each has some drawbacks.

But it is not appropriate when the user is unaware of concrete keywords to describe the queries at the beginning and it can be happen result may not be appropriate and satisfies after using all of the existing query forms.

4. Problem Formulation

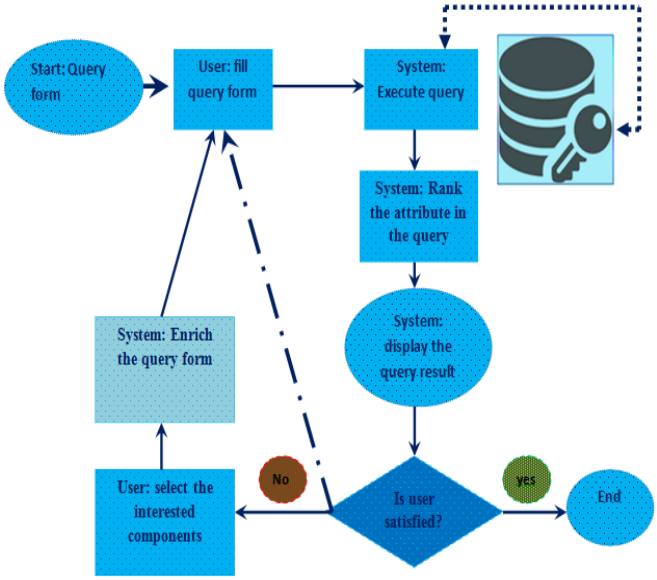

Figure: Architecture of Dynamic query form with NoSQL

Above figure shows the Architecture of dynamic query forms. Dynamic query form (DQF) which generates query form according to the user's desire at run time. The system uses NoSQL and provides a solution for query interface in large and complex databases. F-measure is a metric can be apply to estimate the goodness of a query form. F-measure is a metric in which recall and precision are used to evaluate query result. The metric is also appropriate for query form because query forms are designed to help users to form query for the database. The goodness of a query form is depend on desired query results generated from the query form and time required to generate. Based on this, we can rank and recommend the potential query form with more components. Here efficiency is important because dynamic query form with NoSQL (DQF) is an online system where user often expects quick responses.

Each and every query form represent to SQL query template. Query form provides user form where user will fill parameters to generate different query results. In this section, main focus is on the projection and selection components of a query form. Whatever Ad-hoc join will be done is not handled by the dynamic query form because join is not a part of the query form and that will be invisible for users. Many database queries may output many answers; result will contain many data instances. To avoid many-answer problem, output is a compressed view of result table which will show high level view to get desired query result. Compressed view of result table contains instances which are clusters of actual data instances. Using these instances user can get desired result by click through event in interested instances or clusters.

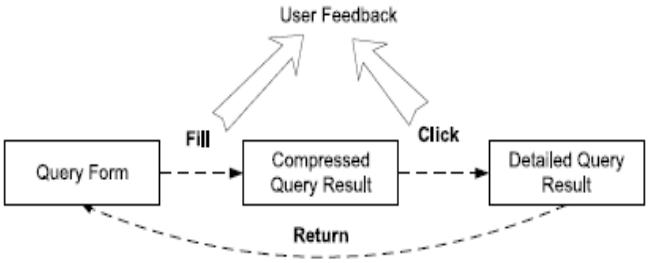

Figure: User Actions

5. Methodology

Our proposed system consist of various modules described as follows

5.1 Attribute Ranking Metric

The generation of dynamic query form is an iterative process and is guided by the user. After every iteration, the system automatically generates ranking lists of form components and the user adds the components into the query form according to the desired result. In this way, a query form could be dynamically refined till the user satisfies with the query results. The form components here refer to the selection and projection components. DQF with NoSQL has two-level ranked list for projection components [8][9]. The first level consist of ranked list of entities and the second level consist of ranked list of attributes from the same entity. The selection attributes must be relevant to the current projected entities; otherwise that selection would be meaningless. Therefore, the system first has to find out the relevant attributes for creating the rank list of selection components. It is necessary to understand how to select relevant attributes and then understand a simple method and a more efficient one-query method to rank selection components.

'A' is given as a set of projection attribute and a universe of selection expressions σ, the desired precision and desired recall of a query form F are denoted as PrecisionE(F) and RecallE(F).

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. RecallE(F) is called 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. From these above two measures, overall performance measure, and expected F-Measure can be calculated using following Equation1. This F-Measure will give the goodness of the query form and thus we can refine the form until it satisfies the user conditions.

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

Equation-1

β is a constant parameter to control the preference on desired precision or desired recall. FScoreE (Fi+1) is the estimated

goodness of the next query form Fi+1. The aim is to maximize the goodness of the next query form, the form components are ranked according to the descending order of FScoreE (Fi+1). So that user will get best ranked component for selection purpose using which using which user can achieve desired query result. FScoreE (Fi+1) is obtained as follows.

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

Equation-2

5.2 Quality Metric

The quality of a particular query result can be depend on precision and recall results. Using query form one can produce different input and different inputs can produce different outputs of query results, it means query form can achieve different precision and recall. So using expected recall and expected precision expected performance of query result can be calculated.
Above probabilistic models are used to find expected recall and expected precision. Using both recall and precision metrics F-measure can be calculated, ultimately performance of query form can be evaluated.

5.3 Query Processing

This system supports to captures user interests during user interactions. User interests can be iteratively to form desired result of user. Each iteration has two stages to interact with user. In the first stage, user can interact to enrich query form. Second stage user can interact to execute query. Dynamic query form generates a ranked list of query form components to the user. Then user selects the desired form components from that list to find desired result. Then user can submit the query by submitting the current query form. This displays the desired query results and based on selected components. On this displayed results user can provide feedback to the system about the query results retrieved.

6. Conclusion and Future Work

We propose dynamic query form with NoSQL which will help users which are unknown to the database. In this system query form can generate results dynamically, based on user preference, historical queries and runtime feedback. Probabilistic models are used to rank query form components. The ranking of form components makes it easier for users to find the query results.

As for future work, we can develop multiple methods to capture the user's interest for the queries besides the click feedback. We can add a text-box for users to input some keywords of queries. For ranking purpose relevance score between the keywords and the query form can be incorporated each step. Here will use a NoSQL database such as MONGODB [4]. The proposed system will consist of optimization technique to optimize the performance of system by eliminating negative feedback from user.

Where the our main focused is on the displaying this improved results in the form itself with the help of graph tools to show the differences in the generated results for each and every single users of the tool, where the every user can judge the effectiveness of our proposed tool.

References

[1] M. Jayapandian and H. V. Jagadish. "Expressive query specification through for customization". In Proceedings of International Conference on Extending Database Technology (EDBT), pages 416–427, Nantes, France, March 2008.
[2] E. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton. "Combining keyword search and forms for ad hoc querying of Databases". In Proceedings of ACM SIGMOD Conference, pages349–360, Providence, Rhode Island, USA, June 2009.
[3] Arto Selminen "Introduction to NoSQL" NoSQL seminar 2012 at TUT
[4] Prof. Stefan keller "MongoDB An introduction and performance analysis" Master of Science in Engineering,Major Software and Systems, HSR Hochschule für Technik Rapperswil
[5] Korzh.com. (2005). EasyQuery http://devtools.korzh.com/eq/dotnet/
[6] Cold Fusion. http://www.adobe.com/products/coldfusion/.
[7] G. Chatzopoulou, M. Eirinaki and N. Polyzotis. "Query recommendations for interactive database exploration" In proceedings of SSDBM, pages 3-18, New Orleans, LA, USA, June 2009.
[8] S.BhaskaraNaik, B.VijayaBhaskar Reddy "Dynamic Query Forms for Non-Relational Database Queries" International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume - 3 Issue -8 August, 2014
[9] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen "Dynamic Query Forms for Database Queries" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING VOL:PP NO:99 YEAR 2013