Optimization Technique for Efficient Dynamic Query Forms with Keyword Search and NoSQL

Kavita Ozarkar[1]
PG Student, Dept of Computer Engineering,
Alard college of engineering and management,
Savitribai Phule Pune University, Pune

Sonali Patil[2]
Professor, Dept of Computer Engineering,
Alard college of engineering and management,
Savitribai Phule Pune University, Pune

Abstract: Modern web database as well as scientific database maintains tremendous and heterogeneous that is unstructured data. In order to mine this data traditional data mining technologies cannot work properly. These real word databases may contain hundreds or even thousands of relations and attributes. Latest trends like Big data and cloud computing that leads to the adoption of NoSQL which simply means Not Only SQL. In current scenario Most of the web applications are hosted in cloud and that available through internet. This create explosion in number of concurrent users. So the technique to handle unstructured data is proposed in our work named as Dynamic query forms with nosql.

This system presents dynamic query form interface for database exploration of an organization. In this system document oriented NoSQL database is used for that purpose MONGODB is used which support dynamic queries that do not require predefined map reduce function. And further the process generation of a query form is an iterative process guided by user. At each step system automatically generate ranking list of form components and user adds the desired form component into query form and submit queries to view query result. Two traditional measures to evaluate the quality of query result i.e, precision and recall is presented. Quality measures can be derived using overall performance measure as Fscore

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

*****

I. Introduction

Query by form is a simple methodology that is frequently used as an entry to database. Dynamic 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. 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. These traditional systems are difficult to use and understand for new users. So they require a long training period to understand these systems. Which certainly leads to the creation of easy to use and quick and powerful query methods for accessing the unstructured datasets.

In this paper a query interface (DQF with NoSQL) is proposed which is capable of dynamically generating query forms for users also provide more options to user for searching purpose. DQF with nosql is used to capture user interests during user interactions and to adapt the query form iteratively. Dynamic queries are a novel approach to information seeking that may enable users to cope with information overload.

NoSQL databases like MongoDB are often highly optimized key–value stores intended for simple retrieval and appending operations.

A collaborative approach[10] to recommend db query form for db exploration is introduced here. Users employ a query interface (typically, a web-based client) to find a series of SQL queries that aim to analyze the data and mine it for interesting information.

A. SQL Vs NoSQL

From many developers facing problems with relational database, but application developers are increasingly turning to NoSQL databases to meet new challenges. Unstructured data is growing faster than structured data. Unstructured data is everywhere, for example, satellite images, like Google Earth picture, scientific data, photographs and videos, Radar or Sonar data. Some text files such as survey results emails, social media data such as YouTube data, Facebook, twitter data, linked-In and Flicker data, Mobile data.

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. 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.

B. MongoDB

MongoDB is an open-source cross platform document oriented database, and it is also a leading NoSQL database. This document consist 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.

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.

C. Negative Feedback

Sometimes user may 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. To avoid negative feedback more options are provided to the user as a hint to select proper condition. Positive feedback is always more informative than negative feedback.

II. Motivation

Existing system uses customized query form where using different mechanism user can create queries and accesses database. Examples of existing database management and development tools, such as EasyQuery [5], Cold Fusion [6], SAP and Microsoft Access. Creation of customized queries totally depends on manual editing. It may possible user is unaware with the database schema in advance, then user can get confused in thousands of data attributes.

Database system simply retrieve tulples from dabase according to where condition. 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.

Here, at the start of the work-flow of the system the user is given with a textbox area where the user can enter the keyword for formalizing the database querying by combining the keyword search and forms. The user can then select the desired form, fill in the form and can submit the form for further processing.

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. Precision and recall are measures used for performance evaluation of information retrieval [9] for ranking purpose.

III. 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.

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.

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. This system take as input a target database and then it generate an index a set of query forms offline [2]. This system returns forms which are relevant to the keywords which are 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. It is not appropriate when the user is new and 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.

IV. Problem Formulation

Below 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. A collaborative approach [10] to recommend db query form for db exploration is introduced which provides more options to user.

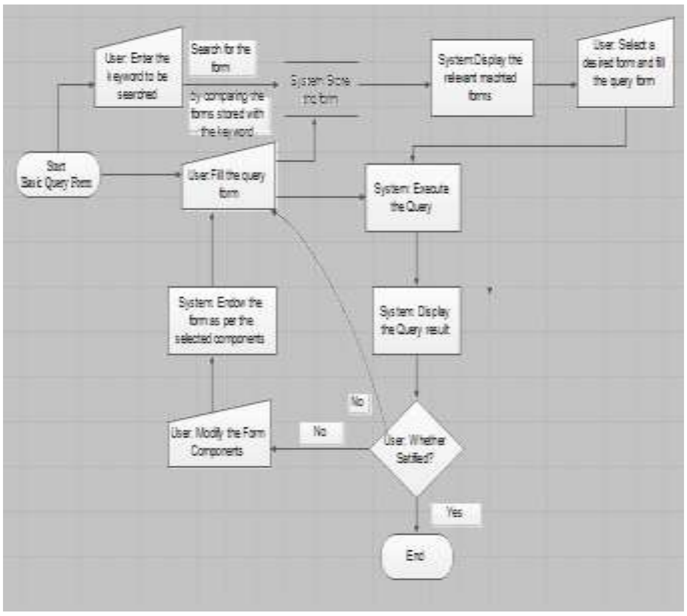

User: Enter the keyword to be searched
Search for the form
by comparing the forms stored with the keyword
System: Store the form
System:Display the relevant machted forms
User: Select a desired form and fill the query form
Start Basic Query Form
User: Fill the query form
System: Execute the Query
System: Endow the forms as per the selected components
System: Display the Query result
No
User: Modify the Form Components
No
User: Whether Satisfied?
Yes
End

Fig- Architecture of Dynamic query form with NoSQL

V. METHODOLOGY

Our proposed system consist of various modules described as follows

A. 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 particular entity. Selection attributes are always 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 set which 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 of Fi that is 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

B. Quality metric

Precision and recall results are used to estimate quality of query result. Query forms are used produce different outputs according to the inputs provided to the query results, it means query form can achieve different precision and recall. Expected performance of query result can be calculated using expected precision and recall.

This probabilistic models are used to calculate expected recall and expected precision. Using both metrics that are recall and precision, expected F-measure can be calculated, using this performance of query form can be evaluated.

C. Collaborative Search

In our proposed work we have added a collaborative approach to recommend db query form for db exploration is introduced. The work-flow starts with a basic query form which contains very few primary attributes of the database. In a collaborative search Here, at the start of the work-flow of the system the user is given with a projection and selection textbox area where the user can enter the required keywords he wants to search. We match these keywords with the history forms and gives user a list of forms which best matches the keywords. When user selects any particular form, form gets open. User can then enter the values and results are displayed to user. At the start of system we have also given user more forms. These forms which are having best fscores. User can directly select the form out of it and fill the values, results are displayed by executing these query with mongodb.

D. 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. Which 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.

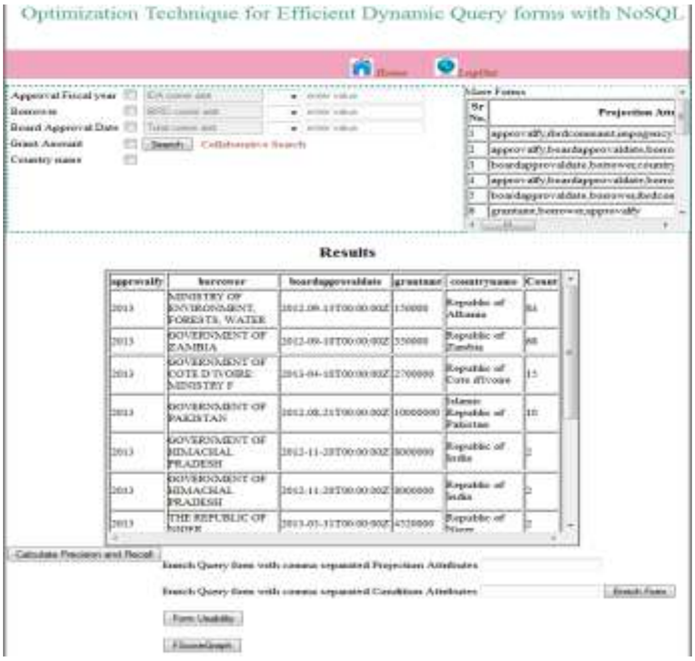
VI. User Study and Experimental Setup

A. Usability matrix:

In this system we have also calculated a widely used matrix which is used for measuring the usability of a system that is FNratio. In a web page based system, FNmax is the total number of UI components in web pages explored by the user and FN is the total number of UI functions used by user. FNratio can be calculated by using following formula.

$$FNratio = (FN/FNmax) * 100$$

We have developed the dynamic query forms as a web application system using JDK 1.8 and Advanced Java Technology such as JSP and Servlets. The database used for our system is MongoDB which is a document oriented NoSQL database. Project Experiments were run using a machine with Intel Core i3 CPU running on Windows 7. Zip code dataset and World Bank JSON Datasets are used in this system that can be used in mongodb database.



Optimization Technique for Efficient Dynamic Query forms with NoSQL

Sr No.	Projection Att
1	approvalfy,lendcommant,impagency
2	approvalfy,boardapprovaldate,borro
3	boardapprovaldate,borrower,country
4	approvalfy,boardapprovaldate,borro
5	boardapprovaldate,borrower,lendcom
6	grantamt,borrower,approvalfy

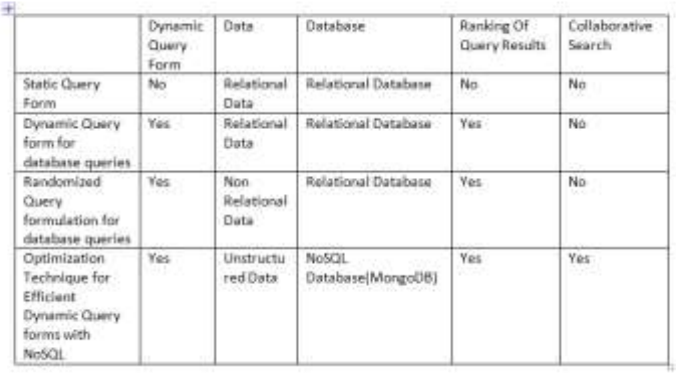
Results

approvalfy	borrower	boardapprovaldate	grantamt	countryname	Coun
2013	MINISTRY OF ENVIRONMENT, FORESTS, WATER	2012-09-11T00:00:00Z	150000	Republic of Albania	84
2013	GOVERNMENT OF ZAMBIA	2012-09-18T00:00:00Z	350000	Republic of Zambia	68
2013	GOVERNMENT OF COTE D'IVOIRE MINISTRY F	2013-04-18T00:00:00Z	2700000	Republic of Cote d'Ivoire	15
2013	GOVERNMENT OF PAKISTAN	2012-08-21T00:00:00Z	10000000	Islamic Republic of Pakistan	10
2013	GOVERNMENT OF HIMACHAL PRADESH	2012-11-28T00:00:00Z	8000000	Republic of India	2
2013	GOVERNMENT OF HIMACHAL PRADESH	2012-11-28T00:00:00Z	8000000	Republic of India	2
2013	THE REPUBLIC OF NIGER	2013-03-31T00:00:00Z	4520000	Republic of Niger	2

Calculate Precision and Recall
Search Query form with comma separated Projection Attributes
Search Query form with comma separated Condition Attributes
Search Form
Form Usability
FScoreGraph

VII. Results

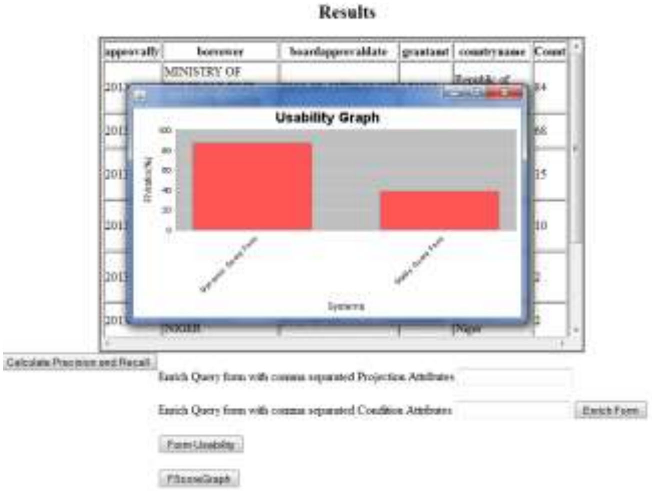
The comparison table of our proposed system with existing system is shown in below table. The table shown below is the comparison of various dynamic query forms with our proposed work. It clearly shows that our system is better as compared to previous systems, As our system can work with unstructured data and also we have added collaborative search mechanism which improves the usability of form.



	Dynamic Query Form	Data	Database	Ranking Of Query Results	Collaborative Search
Static Query Form	No	Relational Data	Relational Database	No	No
Dynamic Query form for database queries	Yes	Relational Data	Relational Database	Yes	No
Randomized Query formulation for database queries	Yes	Non Relational Data	Relational Database	Yes	No
Optimization Technique for Efficient Dynamic Query forms with NoSQL	Yes	Unstructured Data	NoSQL Database(MongoDB)	Yes	Yes

In order to evaluate our system we have calculated usability matrix and compare the results with static Query forms. The graph representing the comparison of Static and dynamic query form is shown in the figure below.

Usability of form is one of the component which is use to evaluate the query form that we have calculated in the form of FNratio. The graph drawn is in between static and dynamic query form against FNratio that is the usability matrix, higher the value of FNratio greater the usability of form. The results clearly shows that the usability of dynamic query form is higher as compared to static query form

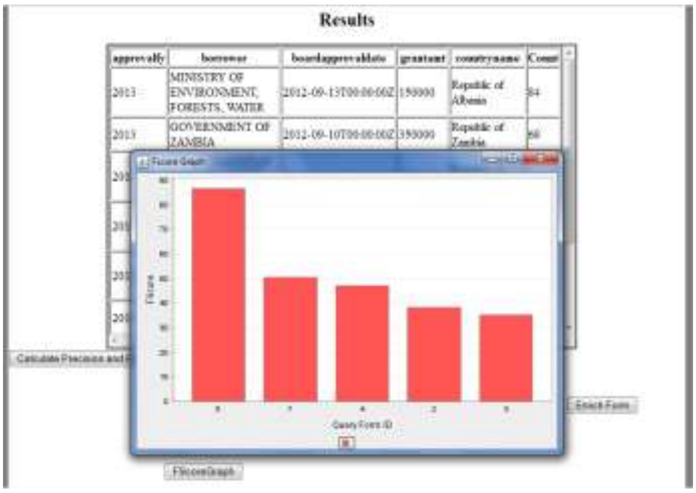


Results

approvalfy	borrower	boardapprovaldate	grantamt	countryname	Count
201	MINISTRY OF			Republic of	84
201					68
201					15
201					10
201					2
201	NIGER			Niger	2

Enrich Query form with comma separated Projection Attributes

Enrich Query form with comma separated Condition Attributes



Results

approvalfy	borrower	boardapprovaldate	grantamt	countryname	Count
2013	MINISTRY OF ENVIRONMENT, FORESTS, WATER	2012-09-13T00:00:00Z	150000	Republic of Albania	84
2013	GOVERNMENT OF ZAMBIA	2012-09-10T00:00:00Z	350000	Republic of Zambia	68

Another graph presenting the fscores of various query forms of our proposed work show below

VIII. 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 also consist of keyword search using collaborative approach to provide more option for user.

IX. 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,

[9] 2014

[10] 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

[11] Remya Raj, Rosna P. Haroon "Dynamic Query Processing and Optimization in relational Complex Databases with Keyword Search" IOSR Journal of Computer Engineering (IOSR-JCE) Volume 16, Issue 5, Ver. III (Sep – Oct. 2014)