

An Alternative For Database Queries:Auto Query Forms

Rohidas B. Sangore¹, Pratik P.Bhavsar², Manish S.Patil³, Tushar M.Chaudhari⁴

Assistant Prof. Department of Information Technology, SSBT's COET, Jalgaon, India¹

Student (Researcher) Department of Information Technology, SSBT's COET, Jalgaon, India²

Abstract: Modern databases contains very huge amount of data. Also the databases have exists various complex relationships among them. These databases contains over thousands of relations and their entities. Existing query forms are unable to generate queries dynamically as per the user requirements.

Alternative Query Form which takes input in the form of human understandable language and converts it into appropriate SQL query and the query will be fired on relational database to fetch the appropriate data. It is very easy and convenient to ask questions to database in natural language especially to the non-technical users who do not understand the database query language such as SQL.

Our system will provide easy access to complex databases using NLP which perform lexicon processing, query formulation, and dynamic generation of SQL queries which can be easily reused or modified.

Keywords: : Database, Natural Language Processing (NLP), Alternative Query Form(AQF), SQL, Query formulation, Ranking method,Data Source Name(DSN),Local Area Network(LAN) etc.

I. INTRODUCTION

Query forms are used to ease the work of database users. Earlier the query forms are designed by developers or DBA using predefined SQL statements. But as development had done the databases becomes so complex that the DBA can't predict all the queries while developing query form. A query forms is popular approach to query databases, as they allow users who have no knowledge of the database query language or the schema to build structured queries.

Natural language processing (NLP) is the ability of a computer to understand what a human is saying to it. By using NLP it is possible to allow users to retrieve data from databases using human understandable language such as English. NLP is also used to create effective communication between end users and computer database without predefined complex SQL commands and queries. Symbolic languages i.e. as Mathematical language are capable of interpreted processing of queries by sentences. The most sophisticated ones provide a simple text box for a query which takes Natural Language (NL) queries as input. Suppose if we want to view information of a particular student from STUD table then we are supposed to use the following query: `SELECT ADDRESS FROM STUD WHERE name ='ABC'`; But if an user doesn't know SQL, he/she will not be able to access the database unless he/she knows the syntax and semantics of firing a query to the database. But using Natural language, this task of accessing the database will be very easy. So the for above query we will write using Natural language such as: What is the address of student "ABC".

II. RELATED WORK

Query forms are designed to return the user's desired result. There are two traditional measures to evaluate the quality of the query results: precision and recall [2]. Query forms are able to produce different queries by different

inputs, and different queries can output different query results and achieve different precisions and recalls, so we use expected precision and expected recall to evaluate the expected performance of the query form. Intuitively, expected precision is the expected proportion of the query results which are interested by the current user. Expected recall is the expected proportion of user interested data instances which are returned by the current query form. The user interest is estimated based on the user's clickthrough on query results displayed by the query form. For example, if some data instances are clicked by the user, these data instances must have high user interests. Then, the query form components which can capture these data instances should be ranked higher than other components. Next we introduce some notations and then define expected precision and recall [1].

Dynamic Data Entry Form: Develops an adaptive forms system for data entry, which can be dynamically changed according to the previous data input by the user. Our work is different as we are dealing with database query forms instead of data-entry forms [3].

Evolutionary algorithms are biologically inspired methods to solve problems in Computer Science, but do not aim to hold a 1:1 relationship with nature and biology. However, the methodology assumes a cycle in which some key concepts in Darwinian evolution are retained. De Jong (2006) sums up the key concepts [5]:

- One or more populations of individuals competing for limited resources
- The notion of dynamically changing populations due To the birth and death of individuals
- A concept of fitness which reflects the ability of an Individual to survive and reproduce
- A concept of variational inheritance: offspring closely resemble their parents, but are not identical De Jong goes on to note that although there is no Consensus on this, a

system would not be described As a Darwinian one if one of these components were Missing. Even if a simple algorithm as this is no plausible model of nature and evolution that takes every biological complexity into account, biology-inspired algorithms can still be used to solve real-world problems. Evolutionary algorithms have been successfully employed in computer science to search, optimization and adaptation to changing environments, as well as being able to model complex behaviour in nature. The author points out that the goal and purpose of evolution remains unclear. Where nature’s evolution at large is headed is largely a philosophical question; however, the goal of employing an optimization or search algorithm is well-defined. Evolutionary algorithms can be continually expanded and refined through the use of more knowledge from biological systems to close the gap between inspiration source and application, but possibly without fully accounting for every aspect of nature. Despite the discrepancy between algorithms and inspiration, evolutionary algorithms can be used to solve the abovementioned challenges [4].

III. EXISTING SYSTEM

In traditional system people access the information using SQL but in modern databases it is very difficult and complex. So information retrieval is very difficult. Suppose there are two tables in database stud and company_details both tables are correlated to each other and manager wants to access the information about their employee’s past education at that time manager fires the query in SQL syntax and to understand this syntax is very difficult for the non-technical user.

IV. DRAWBACKS OF EXISTING SYSTEM

In existing system the user has to create queries in SQL which is totally depend on user so if the user doesn’t familiar with the schema of database in earlier stage then those large number of attributes would like to confuse the user to create SQL queries. Information retrieval is very tedious process which consumes lots of time and user intelligence.

V. OUR APPROACH

In this paper, we propose An alternative for database queries: Auto query forms: AQF, the query form which is very user friendly that means we can easily select tables then it will automatically shows the attributes present in that table after selecting the required attributes.

When we have to enter the query in natural languages then our system can prepare the SQL query by using natural language sentence and it will show the result.

Suppose if the user will be satisfied with result according to his requirement then it gives the ranking to the result like good, poor, average etc and its stored in our system database that means in feature user will be enter the same sentence or related to same sentence at that time system will be search firstly our database ranking and show the good ranking result.

TABLE 1
 Interactions between Users and AQF

Alternative Query for Enrichment	1) AQF recommends a ranked of query form to the user. 2) The user selects the particular form components into the current query form.
Query Execution	1) The user fill up the current query form and enter natural sentence. 2) AQF prepares the query and displays the results. 3) The user provides the opinion about the query results.

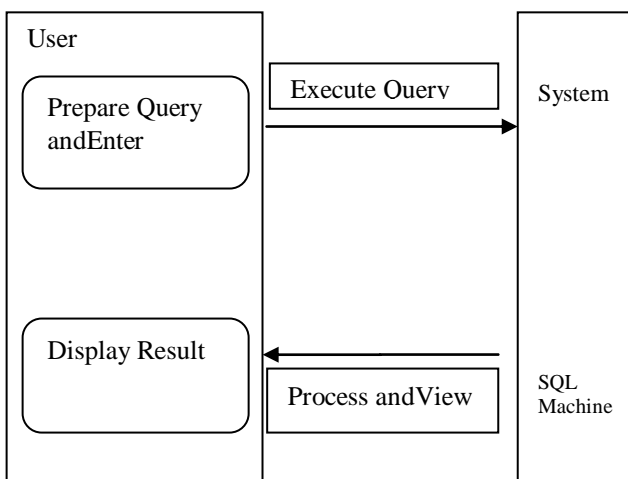


Fig. 1: Preparation of SQL machine

By using the ranking system our system performance will be increase day by day because system show the result accurately and user will be satisfied. T

he system is provide the one feature like as web based that means system firstly prepare DSN of the network and connect to the server.

So the user will be access the server database any system in the LAN or network.

V. SYSTEM ARCHITECTURE

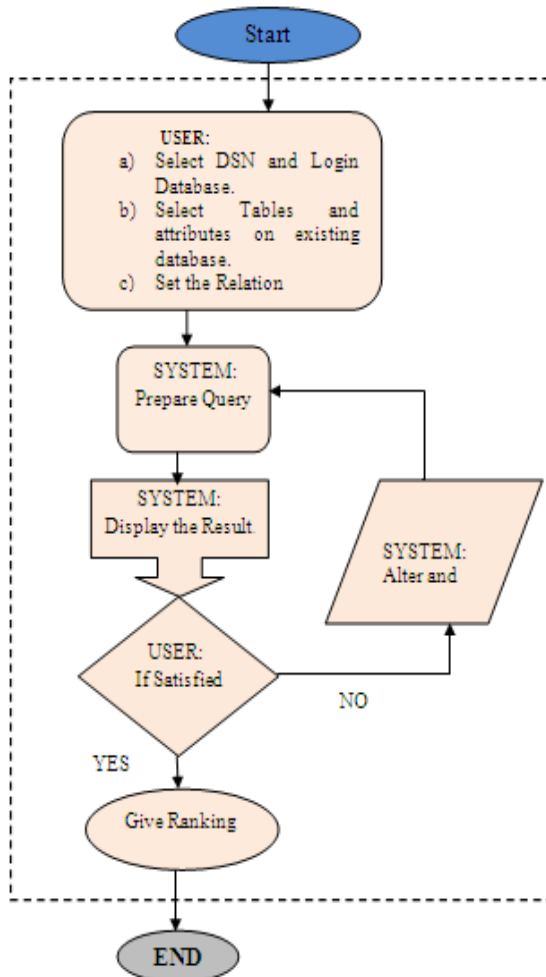


Fig.2: Working of AQF

Our contributions can be summarized as follows:

- We propose an alternative query form system which generates the query forms as per the user's requirement at run time. This system can provide solution for very complex queries applied on a very large database.
- We apply a measure to test the quality of our query form. This measure will help us to evaluate the results generated by our system. If the queries generated by the query forms are as per the user requirement then we can say that our query form is quality based. From this, we can rank the queries according to their importance.
- According to the proposed measure, we develop some algorithms to test the quality of the projection and selection of form components.
- Fig. 2 shows the working of our system. After starting the system first the user has to select a DSN and login database. DSN provides the connectivity to the database and the login database will authenticate user. After logging in the user has to select the tables and particularly the attributes on which he/she has to work. Then user sets the relations between those tables and their attributes.

- After taking all the input from user the system starts its working. The query entered by the user is in the form of natural language so then the system divides that sentence in the form of tokens and then matches those tokens with the existing keyword dictionary. After that the query will be formed.
- According to the query generated by system the appropriate result will be displayed to the user.
- User checks the result and if he/she is satisfied with the result then they will give ranking to the query for further enhancement, otherwise they chose alternate option which shows another query.
- This process continues until the user satisfies with result.

VI. ALGORITHM

- 1) T_0 generate initial population of m individuals
- 2) Set generational counter $k = 1$
- 3) Evaluate T_0 for fitness
- 4) Begin iteration until termination (number of generations or termination criteria reached)
 - Select parents T_{par} T_{k-1} .
 - Get offspring T_{offsp} . By recombining parents.
 - Mutate some offspring.
 - Select population to survive unto next generation $T_k = T_{k-1} \cup T_{offsp}$.
 - Iterate generation counter is $k = k + 1$

VII. CONCLUSION

- In this paper we propose an alternative query form which helps the user to generate the query form at run time. The concept behind this is that the form components can be ranked according to user preference. The user preference can be captured by previous opinions given by user and also his rankings given at run time.

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:PP NO:99 YEAR 2013
- [2] G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1984.
- [3] K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh. Usher: Improving data quality with dynamic forms. In Proceedings of ICDE conference, pages 321-332, Long Beach, California, USA, March 2010.
- [4] Lars Bungum, Bjørn Gambäck Department of Computer & Information Science Norwegian University of Science and Technology SemSælands v. 7-9, 7491 Trondheim, Norway "Evolutionary Algorithms in Natural Language Processing", Norwegian Artificial Intelligence Symposium, Gjøvik, 22 November 2010.
- [5] K. A. De Jong. Evolutionary Computation: A Unified Approach. MIT Press, 2006.