INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue4)
Available online at www.ijariit.com

An Interactive Interface for Natural Language Query Processing to Database Using Semantic Grammar

Soumya M. D
KLE Society's Dr. M. S. Sheshgiri College of Engineering and Technology, Karnataka
soumyamdindur@gmail.com

B. A. Patil
KLE Society's Dr. M. S. Sheshgiri College of Engineering and Technology, Karnataka
b_a_patil@yahoo.com

***Abstract:** Above and beyond the year's field of natural language query processing (NLQP) has seen humongous changes in research as well as methodology. Trending issue within the database management is to provide a high-level interface for the layman. A major source of data which is generated by many organizations and companies are stored in a database which plays important role in IT field. To access these data one needs to know SQL language but since layman doesn't have any knowledge about formal language i.e. SQL a proper interface is required. So an interface for NLQP to the database by using semantic grammar deals with the design and developing a system which understands the natural language like English and converts the English language to SQL queries. So this paper presents the insight of NLQP system on how to enable communication between human and computers without memorization of complex commands and procedures and also NLQP system is able to intelligently process the user request in reasonable human useable format.*

***Keywords:** Database Management System (DBMS), Structured Query Language (SQL), Natural Language (NL), Natural Language Interface for Databases (NLIDB), Natural Language Query Processing (NLQP)*

I. INTRODUCTION

A major source of data which is generated by many organizations and companies are stored in a database which plays important role in IT field. The individuals familiar with query methods can access these data directly. It requires very professional knowledge for dealing with the database by using query languages and also it is complex problem. However, for now, professional user, an opportunity can be created where in user types a query in natural language and it is converted to the SQL like query and finally drives result by Natural language query processing (NLQP).Some software's do not have the ability to generate a complex query. To address this issue, an interface called NLQP using semantic grammar is represented which execute lexical parse and also semantic analysis on natural language query with natural language processing (NLP) and later transform it to the SQL query language commands by which the user's data can be fetched from the database. Questions formed in natural language are a convenient and easy method for accessing the data, especially for not professional users who do not have any knowledge about database query languages like SQL.

An interface for NLQP to the database by using semantic grammar deals with the design and developing a system which understands the natural language like English and converts the English language to SQL queries. The queries executed in DBMS will respond back in the English language. This makes very easy access to the database by no professional user and hence reduces its complexity for dealing the database. The main goal of our system is to design and represent software that will analyze, understand and generate language such that humans understand it or which they use naturally. As in we are addressing the computer just like we address other people.

II. LITERATURE SURVEY

In [1] the author describes the PRECISE. PRECISE is NLIDB which is database independent and it is more or less like a human which allows interaction with the system. Example if a user fires a query "What is the name of the author of the book C-Programming?" and later if the user asks yet another query like "Programming with C?" then automatically second query was being taken as "What is the name of the author of book Programming with C?". This was flexible enough, but also having one disadvantage such that initially, it cannot identify any error before the SQL query is fired to the database. Despite having a disadvantage, PRECISE handles very large amount of natural language questions perfectly. PRECISE having 93.8% accuracy on benchmark ATIS data set.

In [2] the author explains about the natural language interface i.e. NALIX. NALIX acts as an interface to XML database. NALIX which is the interface takes input as English language sentences that may also include many operations such as joins, nesting, aggregation and much more. After the input is received which is the English query being translated after reformulation into XQuery i.e. to be evaluated against XML database. NALIX maintains query history and provides feedback messages if the query is being irrelevant. The disadvantage from the [1] is being solved here by prior detection of the error before the query is being fired to the database.

In [3] the author introduced the concept of semantic templates which is used for translating English queries into SQL queries. The system built is actually a semantic template based system having run time module and a pre processor. It also uses WordNet i.e. a dictionary which includes application programming interface (API) for different kind of languages. By using WordNet one can use hyponyms, synonyms of keywords in the queries. In [4] The system which behaves just like a human in question and answering the natural language queries back in natural language. The author here used machine learning approach to fire any query for QA i.e. not specifically restricted to any of the domain. This approach mainly consists of two steps: firstly, the question which is incoming uses a classifier to find respective tables and columns which are appropriate in the structured database and then retrieve the answers.

III. SYSTEM DESIGN

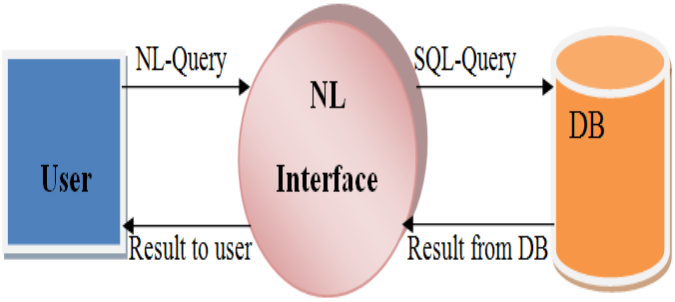

Fig 1: Block diagram of NLQP System

1. The user gives the input to the NL interface as a sentence that will be an English statement to access the database.
2. In NL interface, synonymous words are identified and later input statements are parsed. The statements are converted into an internal representation based on the syntactic and semantic knowledge of the English statement and then converted into SQL queries using NL interface.
3. The Database allows the user to fetch the results based on the SQL query generated by the NL interface and returns the result back to the user in the fastest way with higher quality.

A. Architecture

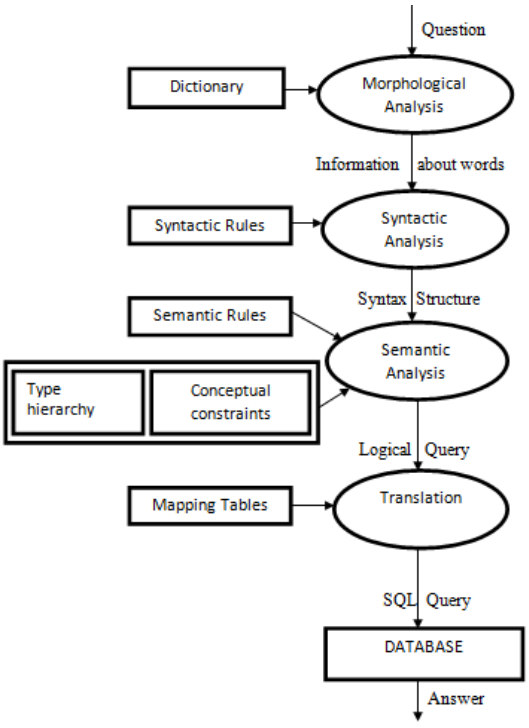

Fig 2: Architecture of NLQP Using Semantic Grammar

The major functions of this system are explained below:

- Analyzer Module
  In this module, the query which is in the English language is taken as input and separates the query into keys such as names, keywords, constants, operators, and variables.
- Parser Module
  From the previous Analyzer module, we get the keys to the Parser module. This module mainly concerned with grouping the tokens into larger syntactic classes such as statements, procedure, and expressions.
- Query Builder Module
  In this module, the data obtained from parsing will be next used to construct the SQL query based on the availability of information.
- Code Optimizer Module
  Code Optimizer module is treated as the top level module for this NLQP operation. This is the one being responsible for retrieving the data from the database depending on the query that we have initially provided.

B. NLQP Algorithm

Step 1: Split query and extract patterns.
Step 2: Tokenization
- Split the query in tokens.
- Give order number to each token identified.

Step 3: Replace synonyms with proper attribute names.
Step 4: Map value for identified attribute and corresponding table.
Step 5: Get an intermediate form of query.
Step 6: Transform it into SQL.
Step 7: Display the results.

C. Sequence Diagram

The Sequence diagram describes the interaction and exchange of messages between the user/administrator and the database through the NLQP engine. Figure 4 display the sequence of working of the system. The NLQP interface initially displays the login page where in the user or administrator needs to validate themselves. If the user is registered user then a user is redirected to next main interface or else invalid message is displayed. Once the user enters English query the NLQP engine generates SQL query and display it back to the user. Later the NLQP engine establishes a connection to the database and executes the SQL query. Once the query is executed the results is fetched and displayed back to the user.

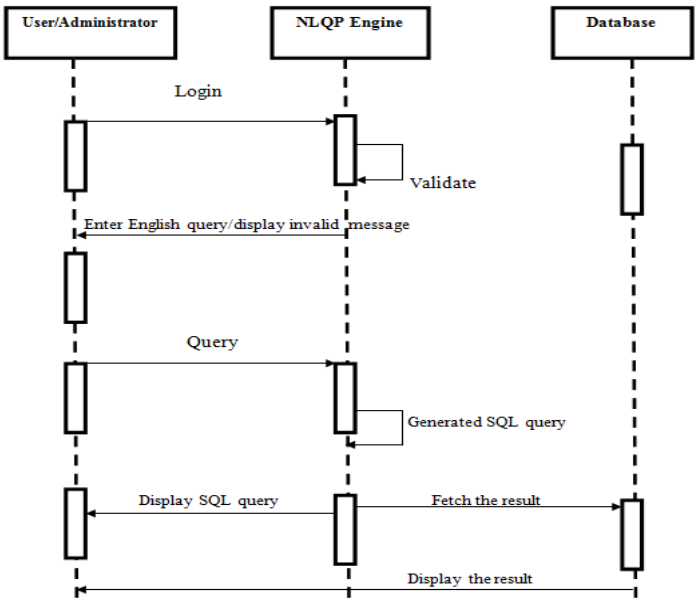

Fig 4: Sequence Diagram for NLQP Using Semantic Grammar

IV.INTERPRETATION OF RESULTS

The user has to first login and then connect to the database. A database setting is a very important requirement for accessing the database. After getting the information about database name, user name and the password, the connection is made. Once the database is setup, Natural Language Query is translated to SQL with the capability of searching multiple tables and multiple fields in the respective database.
In this project there are 2 ways of querying:

1) Using the mouse, that is the top three list in the frontend where we can select the database, then the table and then particular column and the result will be displayed in the text area.

2) There is choice named an example where we can select any of the examples from the drop down menu and then click on Do query button to get the result displayed in the below text area.

The Main interface of the NLQP using semantic grammar is shown in figure 5 which consists of three portions in the upper part. First consist of the Database name, second consist of the table name and last consist of column name with respect to the table name. It also consists of query field in which the natural language query is written and the examples are the drop down menu in which some of the example queries are written.Generated SQL is the one which displays the SQL generated query after the Natural language query is processed. Finally, the result part displays the generated results.

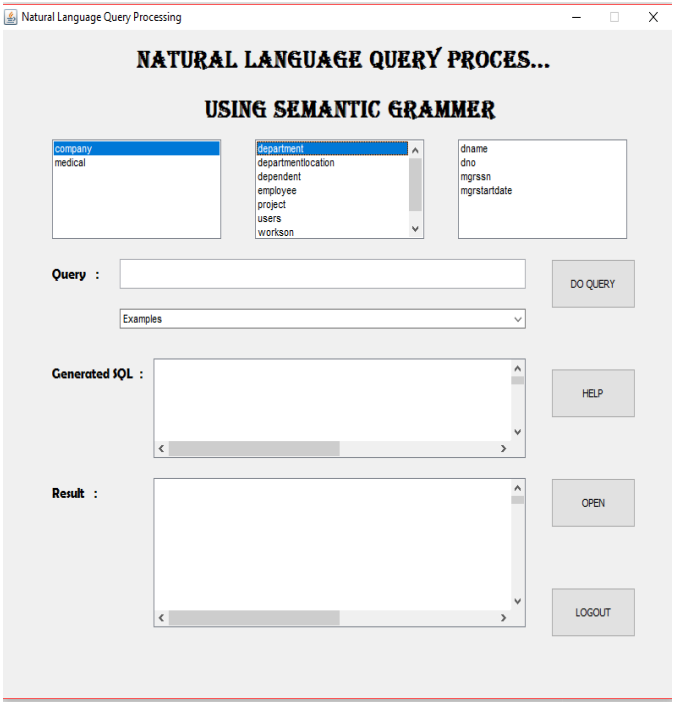

Fig 5: Main Interface which displays the company database and its tables and column names

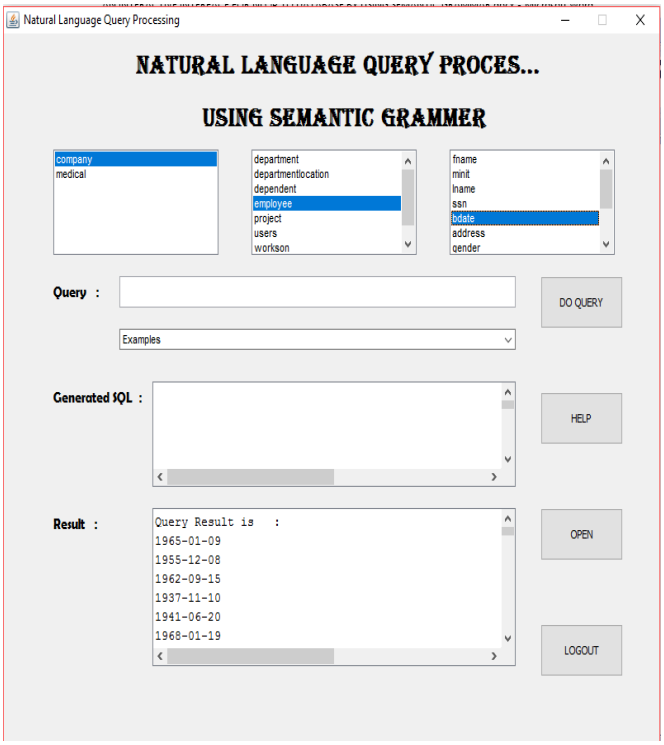

Fig 6: Executing the first type of query by selecting company database along with its respective fields

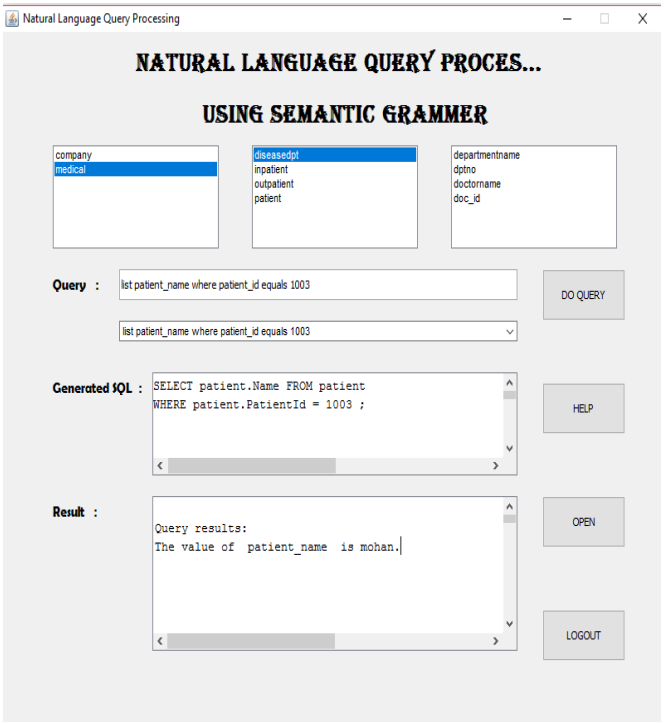

Fig 7: Executing the second type of query by writing natural query for medical database

FUTURE SCOPE

This project can be extended for large scale and more complex queries can be included in GROUP, LOGICAL clauses and EXIST, NOT EXIST clause. Further this project can be extended for voice recognition and can be implemented not only in English but also for other natural languages.

CONCLUSION

An interactive interface for natural language query processing acts as powerful enhancements for any computer program interface. The proposed system is capable of handling simple queries along with some of the standard join conditions. Many complex queries require further development. The system helps in retrieving the data from the database for developing applications in the natural language itself, no need of writing the query in a high-level language. It provides simple and yet easy interface for writing queries in natural language. Our system is able to intelligently process the user request in reasonable human useable format.

ACKNOWLEDGEMENT

I have been bestowed the privilege of expressing my gratitude to everyone who helped me in the successful completion of my dissertation work. I express my sincere gratitude to my guide Prof. B. A. Patil for his valuable guidance and another teaching faculty of Computer Science &Department.

REFERENCES

[1]. Popescu, A. Armanasu, O. Etzioni, D. Ko, A. Yates, “PRECISE on ATIS: Semantic Tractability and experimental results”, Proceedings of the National Conference on Artificial Intelligence- AAAI, pp. 10261027, 2004

[2]. Y. Li, H. Yang, H.V. Jagadish, “NALIX: an interactive natural language interface for querying XML”, Proceedings of the International Conference on Management of Data, pp. 900-902, 2005.

[3]. N. Stratica, L. Kosseim, B. C. Desai, “Using semantic templates for a natural language interface to the CINDI virtual library”, Data and Knowledge Engineering Journal, 55 (1), pp. 4-19, 2004.

[4] M. Khalid, V. Jijkoun, M. Rijke, “Machine Learning for Question Answering from Tabular Data”, IEEE, Database and Expert Systems Applications, 2007. DEXA '07. 18th International Workshop, pp. 392396, 2007.

[5] H. Li, Y. Shi, “A WordNet-Based Natural Language Interface to Relational Databases”, IEEE, Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference, pp. 514-518, 2010.

[6] M. Llopis, A. Ferrandez, “How to make a natural language interface to query databases accessible to everyone: An example”, Computer Standards and Interacts, Elsevier, pp. 470-481, 2012.

[7] A. Shah, J. Pareek, H. Patel, N. Panchal, "NLKBIDB – Natural Language and Keyword Based Interface to Database", IEEE, Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference, pp. 1569-1576, 2013.

[8] S. Srirampur, R. Chandibhamar, A. Palakurthi, R. Mamidi, "Concepts Identification of an NL Query in NLIDB Systems", IEEE, Asian Language Processing (IALP), 2014 International Conference, pp. 230233, 2014

