SQL GENERATION AND EXECUTION FROM NATURAL LANGUAGE PROCESSING

*Saravjeet Kaur, **Rashmeet Singh Bali
*Student, MMU, Mullana
saravjeet.kaur12@gmail.com

ABSTRACT
This paper proposes a method of querying with the databases by means of a natural language interface. This is hot issue in the area of database management is to provide a high level interface for nontechnical users. Normal users are not aware with the formal language like SQL. Then the problem is how they interact with the database system. A normal user may find him/her self handicapped to deal with the database system. The paper presents an interface module that converts user's query given in natural language into a corresponding SQL command. .Asking questions to databases in natural language like English is a very convenient and easy method of data access from database system, especially for normal users who do not understand complicated database query languages such as SQL. This paper proposes the architecture for translating English Query into SQL.
Keywords: Databases, Database Management System (DBMS), Structured Query Language (SQL), Natural Language Interface for Databases (NLIDB), Natural Language Processing (NLP)

1. INTRODUCTION

Database applications play an important role in today's commercial system. Most of the businesses need these types of applications by using the SQL language. Natural language processing (NLP) is becoming one of the most active techniques used in Human-computer Interaction. It is a branch of AI which is used for Information Retrieval, Machine Translation and Language Analysis. The main goal of NLP is to enable communication between human and computers without memorization of complex Commands and procedures. In other words, NLP is the techniques that can make the computer to understand the natural languages used by humans. Today's requirement of commercial system is to extracting data from a DataBase Management System such as MS Access, Oracle and others. A person without knowledge of SQL may find himself/herself handicapped in corresponding with the database. Therefore in this work the development of system for people to interact with the database in simple English language is implemented. This enables a user to input their queries in simple English and get the answer in same language. This is known as a Natural Language Interface to a Database (NLIDB).

2. RELATED WORK

LUNAR (1972) [1] is a system that answers questions about samples of rocks brought back from the moon. In this System the two Databases are used, the chemical analyses and the

literature references. This system uses an Augmented Transition Network (ATN) parser and Wood's Procedural Semantics. **RENDEZVOUS (1977) [2]** in this system, users was able to access databases via natural language. In this Codd's system, special emphasis on query paraphrasing and in engaging users in clarification dialogs when there is difficulty in parsing user input. **PHILIQA (1977)** this was known as Philips Question Answering System **[3],** uses a syntactic parser which runs as a separate pass from the semantic understanding passes. This system is mainly involved with problems of semantics and has three separate layers of semantic understanding. The layers are called "English Formal Language", "World Model Language", and "Data Base Language" and appear to correspond roughly to the "external", "conceptual", and "internal" views of data. **LIFER/LADDER (1978) [4]** was one of the first good database NLP systems. It was designed as a natural language interface to a database of information about US Navy ships. **CHAT-80 (1980) [5]** is one of the best known NLIDB of the early eighties. CHAT-80 is developed in Prolog language. In which English text is transferred into prolog expressions, which were evaluated against the Prolog database? The code of CHAT-80 was circulated widely and formed the basis of several other experimental NLIDB's. **ASK (1983) [5]** allowed end-users to teach the system new words and concepts at any point during the interaction. ASK was actually a complete information management system providing its own built-in database and ability to interact with multiple external databases electronic mail program and other computer applications. All the applications connected to ASK were accessible to the end-user through natural language request. The users started his/her request in English and ASK transparently generated suitable requests to the appropriate underlying system. **EUFID (1983) t**his system consists of three major modules, analyzer module, mapper module and translator module **[6]**. **DATALOG (1984)** is an English database query system based on Cascaded ATN grammar and provides separate representation schemes for linguistic knowledge, general world knowledge, and application domain knowledge, DATALOG was highly portable and extendable **[7]**. **TEAM (1987)** large part of the research of that time was devoted to portability issues. TEAM was designed to be easily configurable by database administrators with no knowledge of NLIDBs **[3, 8] JANUS (1989)** it was similar to multiple underlying systems (databases, expert systems, graphics devices, etc). All the underlying systems could participate in the evaluation of a natural language request, without the user ever becoming aware of the heterogeneity of the overall system. JANUS system also supports temporal questions **[9]**.

3. WORK DONE

This paper work uses **NLP** for interfacing with the Database using natural language. In this work only English language is used as a mean for providing inputs. In this system we consider a database ORACLE and a default table is used which is properly normalized. A system is developed that eliminates the problem of normal user to interact with database with rigid language SQL. The users are able to access information's by issuing query in simple English language. The system is developed in JAVA language. The methodology adopted for preparing this software can be classified through following steps **[10]**.

- **Morphological Analysis:** Individual words are analyzed into their components and non word tokens such as punctuation are separated from the words.
- **Syntactic Analysis:** Linear sequences of words are transformed into structures that show how the words relate to each other.
- **Semantic Analysis:** The structures created by the syntactic analyzer are assigned meanings.
- **Discourse integration:** The meaning of an individual sentence may depend on the sentences that precede it and may influence the meanings of the sentences that follow it.
- **Pragmatic Analysis:** The structure representing what was said is reinterpreted to determine what was actually meant.

The methodologies will be used for solving the above problem.

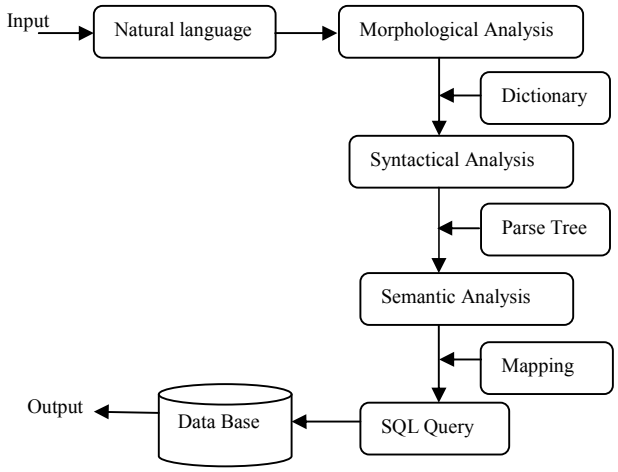

Fig. 1 - Structure of the System

Firstly, an input query is passed in natural language like English. e.g. English query: What is the Saravjeet's RollNumber?

Then the Morphological analysis are identified each world like what, is, the, Saravjeet's, RollNumber. Individual words are analyzed into their components, and it separated noun and adjective in the sentences. Morphological analysis must pull apart the word "Saravjeet's" into the proper noun "Saravjeet" and the possessive suffix "'s". A limited data dictionary is also used to store all related words about the system. After this, Syntactic rules checks the grammatical mistakes of a sentence and Semantic analysis must map individual words into appropriate objects in the knowledge base or database and the meanings of the individual words combine with each other and find out the meaning of simple English query. Example:

Meaning of query: RollNumber of student with name Saravjeet.

Then the translator will change the above sentence with SQL query and with the help of SQL query, we will able to find out the results.

SQL Query: Select RollNumber from student where student name = "Saravjeet".

4. RESULTS

The user has to first login and then connect to the database. Once the database is setup, a user can input his/her query in simple English language. The system can be use by following these steps:-

- Firstly user has to login and enter to the home page of the system.
- Type a query in English language on the text box given in the home page.
- Click on "Execute" button, it will display the results in new page.
- Click on the SQL button, it will display SQL query in new page.
- The system will ask the user for the expected meaning. In case of ambiguities the user has to select the desired query.
- Change Password Wizard to be used by Database Administrator to change the password.
- Add New Topic Wizard to be used to add a new table and the columns of the Table in to the Database.

Outputs

The obtained results of the system software are displayed below:

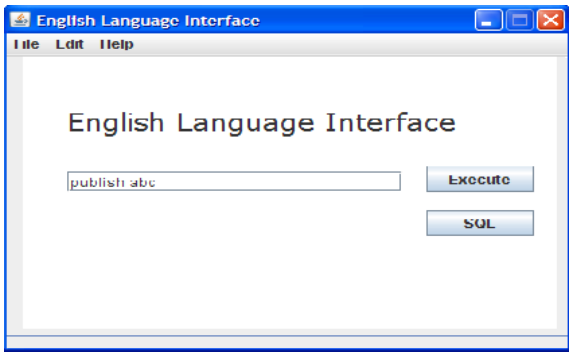

Fig. 2 Example of Query for Selection of Whole Table input by the user.

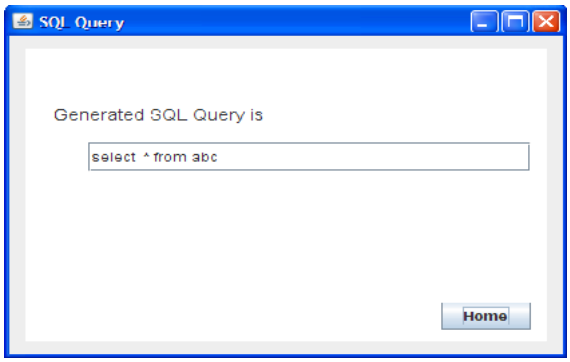

Fig. 3 SQL Query generated for Selection of Whole Table.

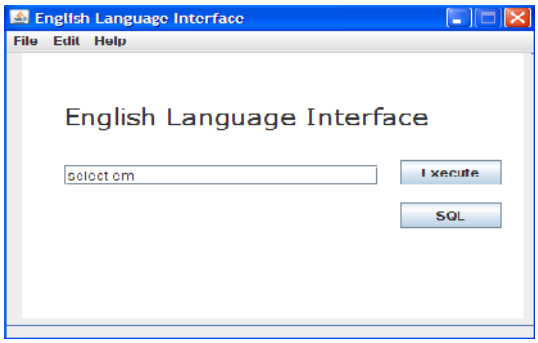

Fig. 4 Execution of the select Query using NLP Language statement when there is ambiguity.

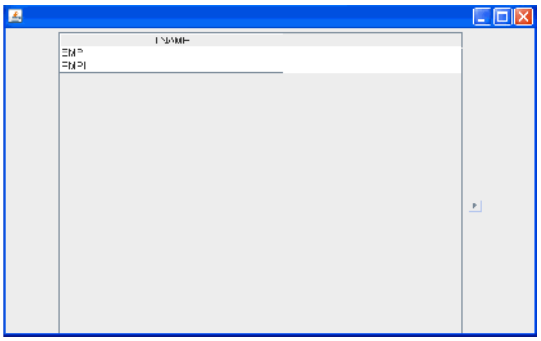

Fig. 5 Ambiguity resolution by displaying all the possible relations in the Query of Figure 4

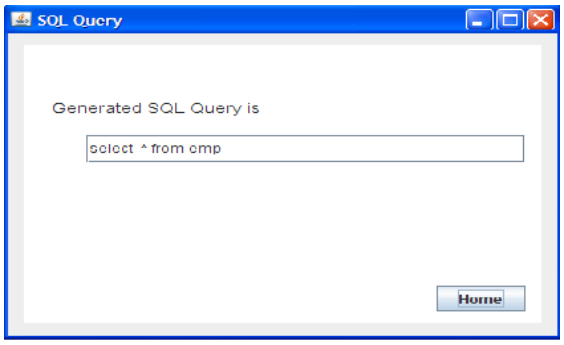

Fig. 6 SQL query is generated Corresponding to fig. 4 query.

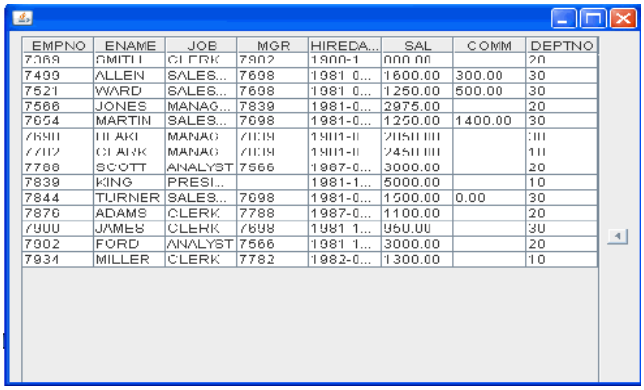

Fig.7 Results after Executing Query for Selection

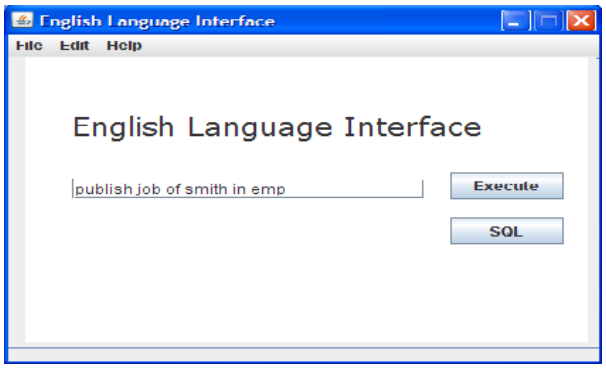

Fig. 8 Executing SELECT Query with ‘where’ Condition.

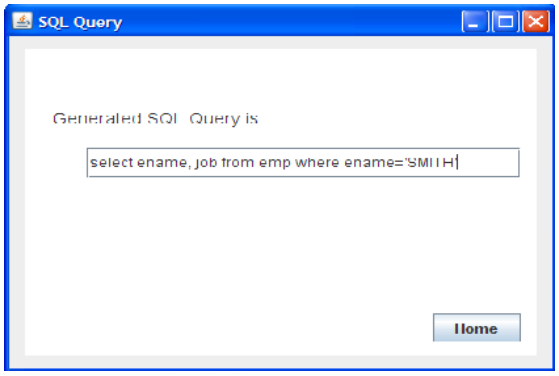

Fig. 9 SQL Query Shown with ‘where’ Condition

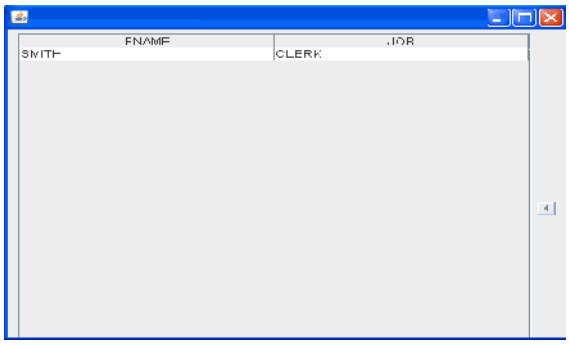

Fig. 10 Results after Executing Query with 'where' Condition.

Fig. 11 Executing SELECT Query with 'where' Condition and relational operator.

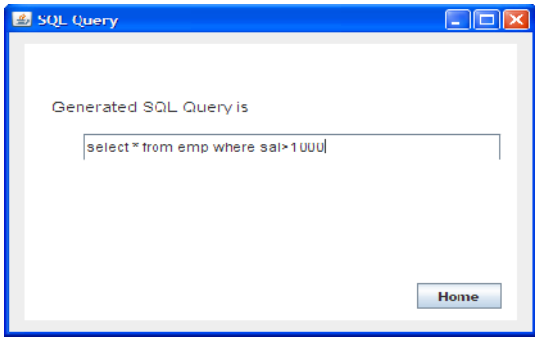

Fig. 12 SQL Query Shown with 'where' Condition and relational operator.

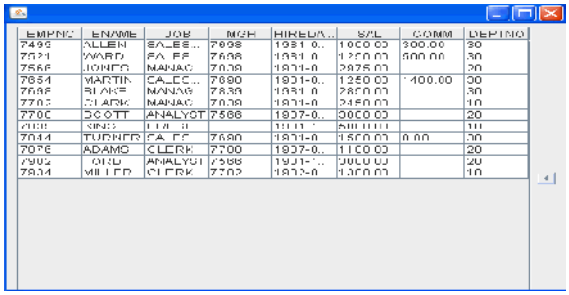

Fig. 13 Results after Executing Query with 'where' Condition and relational operator.

5. CONCLUSION

In this work a system is developed that is able to execute both DDL and DML quires, input by the user in his/her natural language (English). The system is developed in JAVA

Programming language and various tools of java are used to build the system. An oracle database is used to store the information's. Input given by the user is not required in the form of questions (who- form like what, who, where, etc). A limited Data Dictionary is used where all possible words related to a particular system are included. The Data Dictionary of the system must be regularly updated with words that are specific to the particular system. Ambiguity among the words will be taken care of while processing the natural language. The results show that our software is correct and handles the SQL Queries without any problem.

References

[1] W.A. Woods, R.M. Kaplan, and B.N. Webber, "The Lunar Sciences Natural Language Information System", Final Report, vol. BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, 1972.

[2] E.F. Codd, "Seven steps to rendezvous with the casual user", In IFIP orking Conference Data Base Management, pp 179–200, 1974.

[3] R.J.H., Scha, "Philips Question Answering System PHILIQA1", In SIGART Newsletter, no.61. ACM, New York, February 1977.

[4] Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., Slocum, J. "Developing a natural language interface to complex data", ACM Transactions on database systems, 3(2), pp. 105-147, 1978.

[5] Androutsopoulos, G.D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases – An Introduction", Journal of Natural Language Engineering 1 Part 1, 29–81, 1995.

[6] M. Templeton and J. Burger, "Problems in Natural Language Interface to DBMS with Examples from EUFID", In Proceedings of the 1st Conference on Applied Natural Language Processing, Santa Monica, California, pp 3–16, 1983.

[7] C.D. Hafner, "Interaction of Knowledge Sources in a Portable Natural Language Interface", In Proceedings of the 22nd Annual Meeting of ACL, Stanford, California, pp 57–60, 1984.

[8] B.J. Grosz, "TEAM: A Transportable Natural-Language Interface System", In Proceedings of the 1st Conference on Applied Natural Language Processing, Santa Monica, California, pp 39–45, 1983.

[9] P. Resnik, "Access to Multiple Underlying Systems in JANUS", BBN report 7142, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, September 1989.

[10] Gauri Rao et al, "Natural language query Processing using semantic Grammar", International Journal on Computer Science and Engineering, Vol. 02, 219-223, 2010.