NATURAL LANGUAGE QUERY PROCESSING BASED ON PROBABILISTIC CONTEXT FREE GRAMMAR

Gauri Rao[1],S.H.Patil[2]

[1] *Lecturer department o f Computer Engineering Bharati Vidyapeeth College of Engineering, Pune , India*

[2] *Professor and Head department o f Computer Engineering Bharati Vidyapeeth College of Engineering, Pune , India*

E-mail: grrao@bvucoep.edu.in, shpatil@bvucoep.edu.in,

Abstract:

The field of natural language processing (NLP) has seen a dramatic shift in both research direction and methodology in the past several years. In the past, most work in computational linguistics tended to focus on purely symbolic methods. Recently, more and more work is shifting toward hybrid methods that combine new empirical corpus-based methods, including the use of probabilistic and information theoretic techniques, with traditional symbolic methods.

The main purpose of ***Natural Language Query Processing*** *is for an English sentence to be interpreted by the computer and appropriate action taken. Asking questions to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL.*

This paper proposes the architecture of a new NLDBI system including its probabilistic context free grammar, the inside and outside probabilities which can be used to construct the parse tree and the usage of dependency structures and verb sub categorization in analyzing the parse tree.

Keywords:

Natural language database interface; context free grammars, Morphological analysis ;LR Parsing; Verb sub categorization; Dependency Structure.

1.0 INTRODUCTION

Natural language processing is becoming one of the most active areas in Human-computer Interaction. The goal of NLP is to enable communication between people and computers without resorting to memorization of complex commands and procedures. In other words, NLP is a technique which can make the computer understand the languages naturally used by humans, but not by artificial or man-made language such as a programming language.

how to link those concepts together in a meaningful way. While natural language may be the easiest symbol system for people to learn and use, it has proved to be the hardest for a computer to master.

Despite the challenges, natural language processing, or NLP, is widely regarded as a promising and critically important endeavor in the field of computer research. The general goal for most computational linguists is to imbue the computer with the ability to understand and generate natural language so that eventually people can address their computers through text as though they were addressing another person. The applications that will be possible when

NLP capabilities are fully realized are impressive computers would be able to process natural language, translating languages accurately and in real time, or extracting and summarizing information from a variety of data sources, depending on the users' requests.

2.0 RELATED WORK

The very first attempts at NLP database interfaces are just as old as any other NLP research. In fact database NLP may be one of the most important successes in NLP since it began. Asking questions to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL. The success in this area is partly because of the real-world benefits that can come from database NLP systems, and partly because NLP works very well in a single-database domain. Databases usually provide small enough domains that ambiguity problems in natural language can be resolved successfully.
LUNAR (Woods, 1973) involved a system that answered questions about rock samples brought back from the moon. Two databases were used, the chemical analyses and the literature references. The program used an Augmented Transition Network (ATN) parser and Woods' Procedural Semantics. The system was informally demonstrated at the Second Annual Lunar Science Conference in 1971.[1]
LIFER/LADDER 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. This system, as described in a paper by Hendrix (1978), used a semantic grammar to parse questions and query a distributed database. The LIFER/LADDER system could only support simple one-table queries or multiple table queries with easy join conditions.[4]

3.0 SYSTEM DESCRIPTION

A brief description of the project is as follows: Suppose we consider a database say ORACLE. Within this oracle database we have placed certain tables, which are properly normalized. Now if the user wishes to access the data from the table, he/she has to be technically proficient in the SQL language to make a query for the ORACLE database. Our project eliminates this part and enables the end user to access the tables in his/her language.

Let us take an example:
Suppose if we want to view information of a particular employee from EMP table then we are suppose to use the following query:
SELECT * FROM EMP WHERE e_name ='ABC';

But a person, who doesn't know ORACLE, 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 NLP, this task of accessing the database will be much simpler. So the above query will be rewritten using NLP as:
Give the information of employee whose name is ABC.
Both the SQL statement and NLP statement to access the EMP table would result in the same output the only difference being, a normal person who doesn't know anything about SQL can easily access the ORACLE database.

4.0 SCOPE OF THE SYSTEM

- To work with any RDBMS one should know the syntax of the commands of that particular database software (Microsoft SQL, Oracle, etc.).
- Here the Natural language processing is done on English i.e. the input statements have to be in English.
- Input from the user is taken in the form of questions (wh- form like what, who, where, etc).
- A limited Data Dictionary is used where all possible words related to a particular system will be 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.
- All the names in the input natural language statement have to be in double quotes.
- Data Dictionary used will be: - EMP, DEPT and PROJECT

5.0 SYSTEM ARCHITECTURE

The system includes the following components :

- Designing the front end or the user interface where the user will enter the query in Natural Language.
- Parsing: Derives the Semantics of the Natural Query given by the user and parses it in its technical form.
- Query Generation: After the successful parsing of the statement given by the user, the system generates a query against the user statement in SQL and further gives it to the back end database.
- Data Collection: This module collects the output of the SQL statement and places it in the User Interface Screen as a result form.

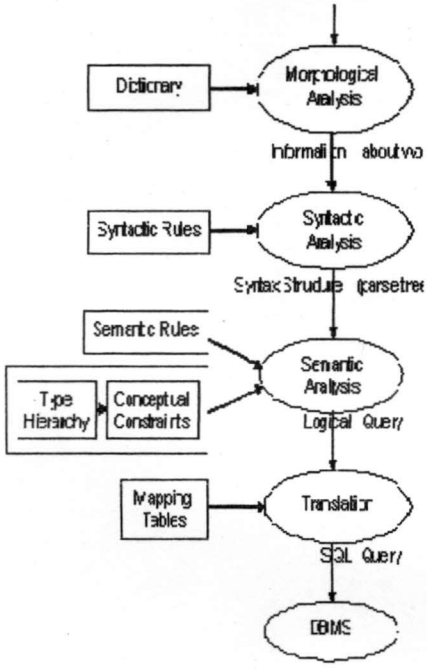

Fig 1.0 Stages In Natural Language Interpretation Process

6.0 OVERVIEW OF PROCESS

The first stage is the token generation, or lexical analysis, by which the input character stream is split into meaningful symbols defined by a grammar of regular expressions. For example, a calculator program would look at an input such as "12*(3+4)^2" and split it into the tokens 12, *, (, 3, +, 4,), ^, and 2, each of which is a meaningful symbol in the context of an arithmetic expression. The parser would contain rules to tell it that the characters *, +, ^, (and) mark the start of a new token, so meaningless tokens like "12*" or "(3" will not be generated.

The next stage is parsing or syntactic analysis, which is checking that the tokens form an allowable expression. This is usually done with reference to a context-free grammar which recursively defines components that can make up an expression and the order in which they must appear. However, not all rules defining programming languages can be expressed by context-free grammars alone, for example type validity and proper declaration of identifiers. These rules can be formally expressed with attribute grammars.

The final phase is semantic parsing or analysis, which is working out the implications of the expression just validated and taking the appropriate action. In the case of a calculator or interpreter, the action is to evaluate the expression or program; a compiler, on the other hand, would generate some kind of code. Attribute grammars can also be used to define these actions.

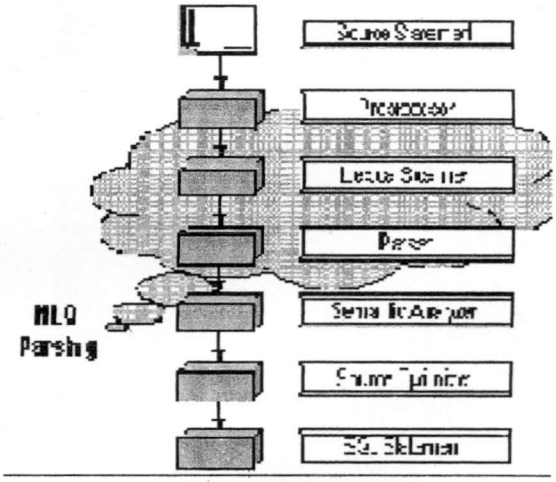

Fig 2.0 **Query Generation process**

7.0 SQL TRANSLATION

After obtaining the parsed grammar tree, the next step is translating the leaves of the tree to the corresponding SQL. Actually the process is collecting information from the

parsed tree. Two techniques may be used to collect the information: dependency structure and verb sub categorization. A dependency structure is used to capture the inherent relations occurs in the corpus texts that may be critical in real-world applications, and it is usually described by typed dependencies. A typed dependency represents grammatical relations between individual words with dependency labels, such as subject or indirect object.[9]

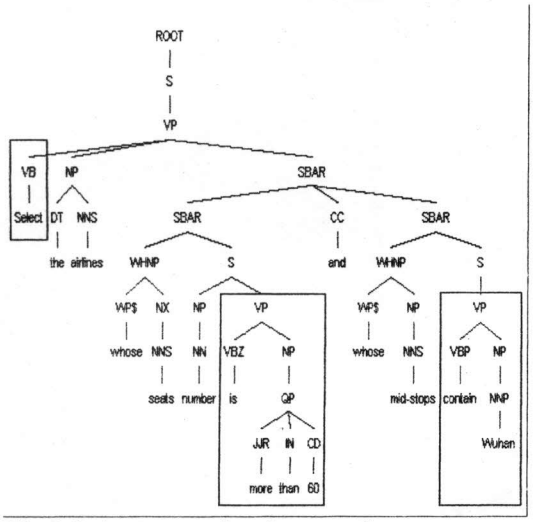

Fig 3.0 An example of parsed tree

Verb sub categorization is the other technique. If we know the subcategorization frame of the verb, we can find the objects of this verb easily, and the target of the query can be found easily. Usually this process can be done after scanning the parsed tree
The SQL translator scanned the parsed tree and recognized the phrases which may be the targets of the query which are sub-trees in the rectangles. . [1]

8.0 CONCLUSION

Natural Language Processing can bring powerful enhancements to virtually any computer program interface, because language is so natural and easy to use for humans. The Intelligent Tutoring System, NLDBI is no exception. Alternatives for integrating a database NLP component into the NLDBI were considered and assessed.

The Project Management database NLP system is currently capable of handling simple queries, standard join conditions. Because not all forms of SQL queries are supported, further development would be required before the system can be used within NLDBI.

References

[1] Huangi,Guiang Zangi, Phillip C-Y Sheu "A Natural Language database Interface based on probabilistic context free grammar", IEEE International workshop on Semantic Computing and Systems 2008

[2] Akama, S. (Ed.) *Logic, language and computation,* Kulwer Academic publishers, pp. 7-11, 1997.

[3] ELF Software CO. *Natural-Language Database Interfaces from ELF Software Co,* cited November 1999, available from Internet: http://hometown.aol.com/elfsoft/

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

[5] Joseph, S.W., Aleliunas, R. "A knowledge-based subsystem for a natural language interface to a database that predicts and explains query failures", in *IEEE CH,* pp. 80-87, 1991.

[6] Mitrovic, A. *A knowledge-based teaching system for SQL,* University of Canterbury, 1998. Moore, J.D. "Discourse generation for instructional applications: making computer tutors more like humans", in *Proceedings AI-ED,* pp.36-42, 1995.

[7] Suh, K.S., Perkins, W.C., "The effects of a system echo in a restricted natural language database interface for novice users", in *IEEE System sciences,* 4, pp. 594-599, 1994.

[8] Whenhua, W., Dilts, D.M. "Integrating diverse CIM data bases: the role of natural language interface", in *IEEE Transactions on systems, man, and cybernetics,* 22(6), pp. 1331-1347, 1992.

[9] Dan Klein, Christopher D. Manning: Corpus-Based
Induction of Syntactic Structure: Models of

Dependency and Constituency. ACL 2004:
478-485.

Biography `