NATURAL LANGUAGE QUERY PROCESSING USING PROBABILISTIC CONTEXT FREE GRAMMAR

Arati K. Deshpande[1] and Prakash. R. Devale[2]
[1]Student and [2]Professor & Head,
Department of Information Technology, Bharati Vidyapeeth Deemed University, Pune, India

ABSTRACT
Databases have become ubiquitous. Almost all IT applications are storing into and retrieving information from databases. Retrieving information from the database requires knowledge of technical languages such as Structured Query Language (SQL). However majority of the users who interact with the databases do not have a technical background and are intimidated by the idea of using languages such as SQL. This has led to the development of a few Natural Language Database Interfaces (NLDBIs). A NLDBI allows the user to query the database in a natural language. This paper highlights on architecture of new NLDBI system using Probabilistic Context Free Grammar (PCFG), its implementation and discusses on results obtained. In most of the typical NLDBI systems the natural language statement is converted into an internal representation based on the syntactic and semantic knowledge of the natural language. This representation is then converted into queries using a representation converter. A natural language query is translated to an equivalent SQL query after processing through various stages. The work has been experimented on primitive database queries with certain constraints.

KEYWORDS: *Natural language database interface, representation converter, syntactic and semantic knowledge, PCFG.*

I. 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. 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 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 instill 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.
This paper describes a natural language database interface that wires complex queries based on a probabilistic context free grammar (PCFG) to relational database. First we summarize some classic NLDBI systems. Consequently we discuss the overall system architecture of the natural language database interface, some implementation details and experimental results.

II. RELATED WORK

The very first attempts at Natural language 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 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. Here are some examples of database NLP systems:

2.1 LUNAR

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]

2.2 LIFER/LADDER

It 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 LIFERILADDER system could only support simple one-table queries or multiple table queries with easy join conditions. [4]

2.3 CHAT-80

The system CHAT-80 [5] is one of the most referenced NLP systems in the eighties. The system was implemented in Prolog. The CHAT-80 was an quite impressive, efficient and sophisticated system. The database of CHAT-80 consists of facts (i. e. oceans, major seas, major rivers and major cities) about 150 of the countries world and a small set of English language vocabulary that are enough for querying the database.

III. SYSTEM DESCRIPTION

A brief description of the system is given. Let us consider a database say ORACLE. Within this oracle database we have stored some tables which are properly normalized. Now if the user wishes to access the data from the table, he/she has to be technically strong in the SQL language to make a query for the ORACLE database. This system cuts this part and enables the end user to access the tables in his/her language.

3.1 SYSTEM SCOPE

- Input to this system is in natural language, here it is in English.
- A limited data dictionary is used in which all possible related to a particular system is stored.
- 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.

3.2 SYSTEM ARCHITECTURE

The system admits the following elements.
Designing the front end or the user interface where the user will enter the query in Natural Language.
1. Manage corpus: A data dictionary is used in which all possible word related to a particular system is loaded
2. PCFG rule: a Probabilistic Context Free Grammar(PCFG) rule is entered with probability.
3. Parsing: Derives the Semantics of the Natural Query given by the user and parses it in its technical form.
4. PCFG rules used: After the successful parsing of the statement given by the user, the system generates CYK chart against the user statement.
5. SQL query: This module generates the corresponding SQL statement and places it in the User Interface Screen as a result form.

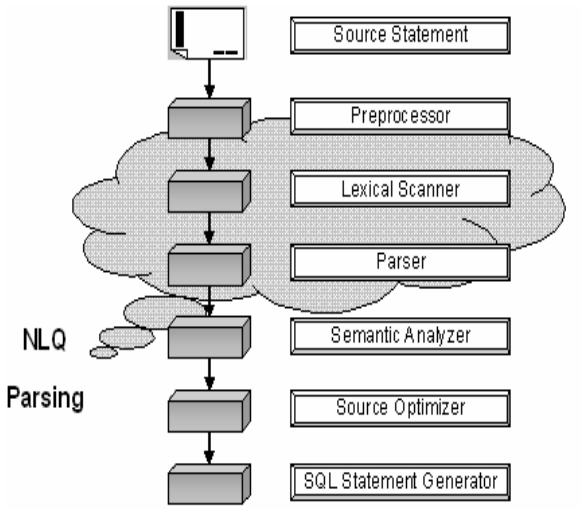


Figure. 1 Architecture of NLDBI System.

The system architecture of natural language database interface developed is given in Figure. 1, which depicts the layout of the processes included in converting Natural Language query into a syntactical SQL query to be fired on the RDBMS.

To process a query, the first step is speech tagging followed by word tagging. The second step is parsing the tagged sentence by a grammar. The grammar parser analyzes the query sentence according to the tag of each word and generates the grammar tree/s. Finally, the SQL translator processes the grammar tree to obtain the SQL query.

SQL translator is used 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 [13, 14, 15]. These techniques are also used in disambiguation, since a PCFG is context-free and ancestor-free, any information in the context of a node in the parsed tree needs not be taken into account when constructing the parsed tree. 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. Verb sub categorization is the other technique [16]. If we know the sub categorization frame of the verb, we can find the objects of this verb easily, and the target of the query can be found easily.

This paper is based on a concept of processing user natural language into a technical form so as to access the data from higher end data storage. NLDBI is a system that allows users to access a database in natural language and has been a popular field of study. Suppose we consider a properly normalized database. Now if the user wishes to access the data from the table, he/she accesses the tables in his/her language.

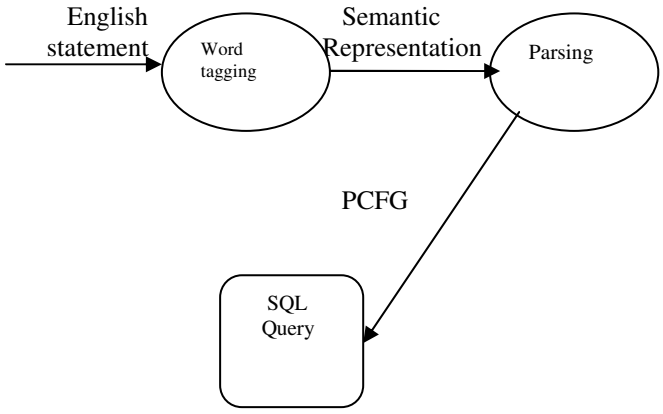


Figure. 2 Generation of SQL query from English Statement.

IV. RESULTS

The user interface of or NLDBI system is shown in Figure 3. At first the user clicks on Manage Corpus button so that all the data dictionary is loaded. When the user clicks on the pcfg Rules setting the user has to enter Probabilistic Context Free Grammar as shown in fig 4.After this the user has to enter his query in natural language. Parser option will generate the parse tree. For example, for the query "Select the airlines whose seats number is more than 60 and whose mid-stops contain Wuhan", the system generated the parsed tree shown in Figure 6 using the Stanford Lexicalized Parser v1.6 that, in turn, uses the Penn Treebank.

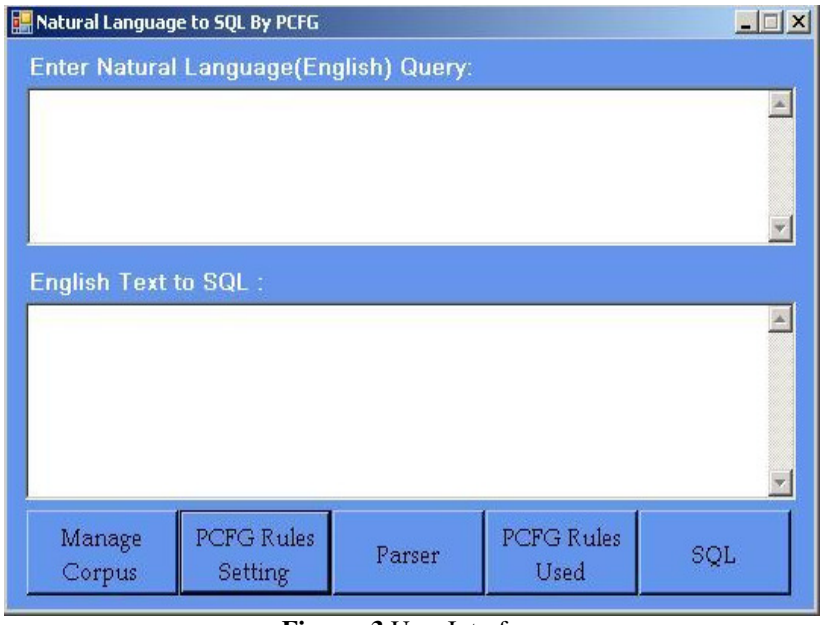
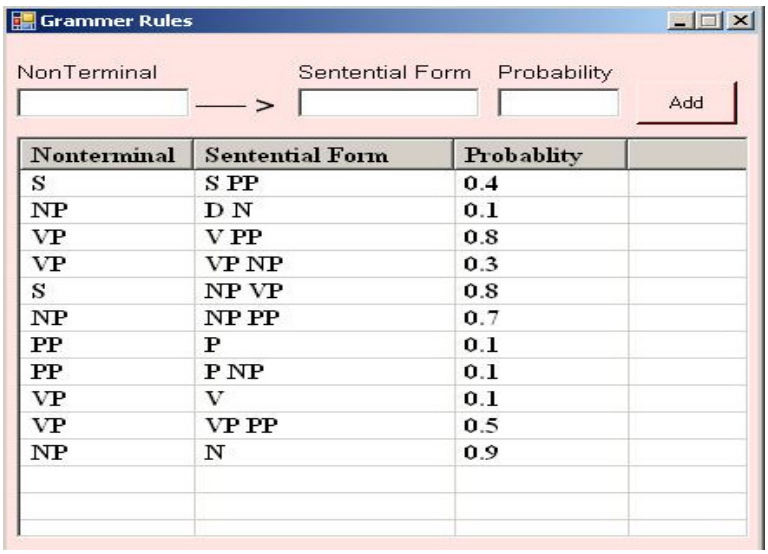


Figure. 3 User Interface

After the successful parsing of the statement given by the user, the system generates CYK chart against the user statement. The SQL module generates the corresponding SQL statement and places it in the User Interface Screen as a result form as shown in the Fig 5.For example if the user enters the Natural Language "find all student name.". the corresponding SQL query displayed will be " select * from STUDENT".



Nonterminal	Sentential Form	Probablity
S	S PP	0.4
NP	D N	0.1
VP	V PP	0.8
VP	VP NP	0.3
S	NP VP	0.8
NP	NP PP	0.7
PP	P	0.1
PP	P NP	0.1
VP	V	0.1
VP	VP PP	0.5
NP	N	0.9

Figure.4 PCFG Rules

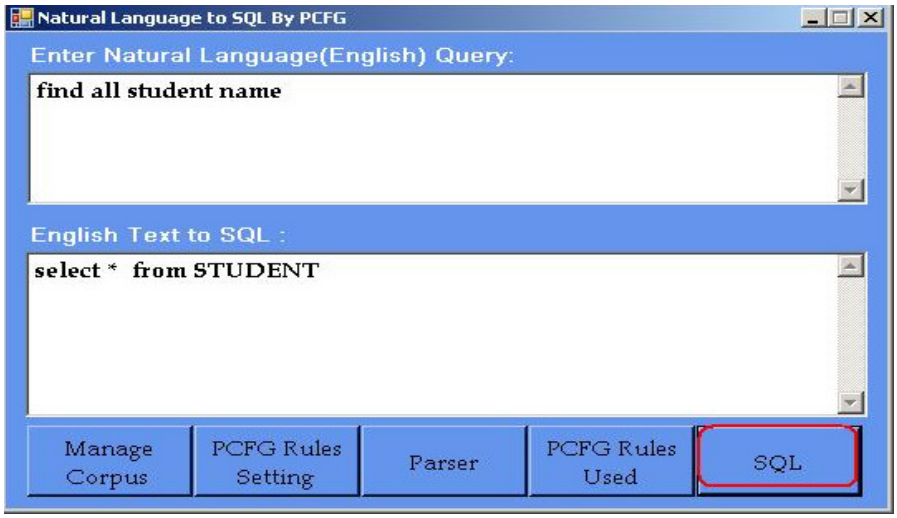


Figure 5. Final Output after query conversion

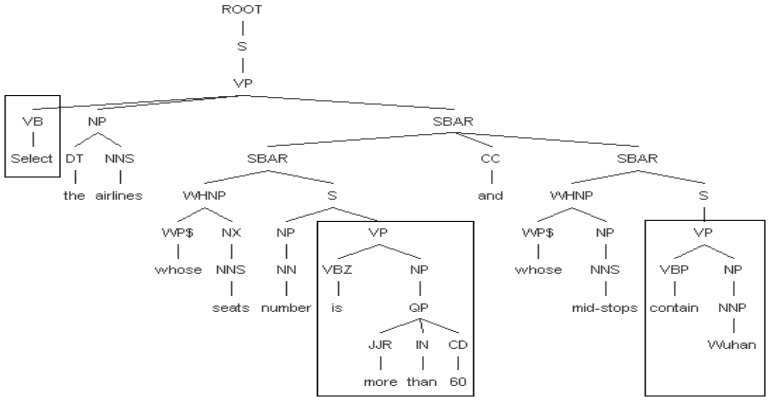


Figure 6. An example parsed tree

V. CONCLUSIONS AND FUTURE WORKS

The system accepts an English language requests that is interpreted and translated into SQL command using semantic grammar technique. In addition, the system requires a knowledge base that consists of a database and its schema. The result of the number of experiments in the form of trials in a user friendly environment had been very successful and satisfactory. To improve the system performance in natural language processing various issues like enriching the knowledge sources of the system in order to increase the system efficiency and researching methods to improve the coherence and the fluency of output texts must be considered. From the experiments we can seen it is possible to translate a natural language query to SQL, and a probabilistic approach may be promising. So far, our NLDBI system considers *selection* and a few simple aggregations. The next aim of our research is to optimize the PCFG, to accommodate more and more complex queries.

ACKNOWLEDGEMENTS

I express my deep sense of gratitude and my research guide Prof. P. R. Devale for his continuous inspiration and valuable guidance in throughout my dissertation work.

REFERENCES

[1] Woods, W., Kaplan, R. “Lunar rocks in natural English: Explorations in natural language question answering”. Linguistic Structures Processing. In Fundamental Studies in Computer Science, 5:521-569, 1977.
[2] Androutsopoulos, I., Richie, G.D., Thanisch, P.“Natural Language Interface to Database – An Introduction”. Journal of Natural Language Engineering, Cambridge University Press. 1(1), 29-81, 1995.

[3] Linguistic Technology. English Wizard – Dictionary Administrator's Guide. Linguistic Technology Corp.Littleton, MA, USA, 1997.
[4] Hendrix, G. (1977). The LIFER manual A guide to building practical natural language interfaces. SRI Artificial Intelligence Center, Menlo Park, Calif. Tech. Note 138.
[5] Warren, D., Pereira, F. (1982). An efficient and easily adaptable system for interpreting natural language queries in Computational Linguistics. Volume 8 pages 3 – 4.
[6] I. Androutsopoulos, G.D. Ritchie, and P. Thanisch, Natural Language Interfaces to Databases – An Introduction, Journal of Natural Language Engineering 1 Part 1 (1995), 29–81.
[7] 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.
[8] ELF Software CO. *Natural-Language Database Interfaces from ELF Software Co,* cited November 1999, available from Internet:
[9] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates, Modern Natural Language Interfaces to Databases:Composing Statistical Parsing with Semantic Tractability, COLING (2004).
[10] Natural language Interface for Database: A Brief review, Mrs. Neelu Nihalani, Dr. Sanjay Silakari, Dr. Mahesh Motwani. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011 ISSN (Online): 1694-0814
[11] Generic Interactive Natural Language Interface to Databases (GINLIDB) Faraj A. El-Mouadib, Zakaria S. Zubi, Ahmed A. Almagrous, and Irdess S. El-Feghi. International Journal of Computers Issue 3, Volume 3, 2009.
[12] Miikkulainen R., "Natural language processing with subsymbolic neural networks", Neural Network Perspectives on Cognition and Adaptive Robotics.(2011)
[13] Dan Klein, Christopher D. Manning: Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency. ACL 2004: 478-485.
[14] M-C.de Marneffe, B. MacCartney, and C. D. Manning. "Generating Typed Dependency Parses From Phrase Structure Parses". In Proceedings of the IEEE /ACL 2006 Workshop on Spoken Language Technology. The Stanford Natural Language Processing Group. 2006.
[15] Dan Klein and Christopher D. Manning. 2003. Fast Exact Inference with a Factored Model for Natural Language Parsing. In Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, MA: MIT Press, pp. 3-10.
[16]Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.
[17] Phillip McCarthy, Applied Natural language Processing: Identification, Investigation and Resolution.(2011).

Biography:

Arati K. Deshpande, PG Scholar in information Technology at Bharati Vidyapeeth Deemed University, Pune. Her fields of interest are Natural Language Processing, Operating system and Computer Networking.

Prakash Devale, presently working as a professor and Head department of Information Technology at Bharati Vidyapeeth Deemed University College of Engineering, Pune. He received his ME from Bharati Vidyapeeth University and pursuing Ph.D degree in natural language processing.