

ACCESSING DATABASE USING NLP

Pooja A.Dhomne¹, Sheetal R.Gajbhiye², Tejaswini S.Warambhe³, Vaishali B.Bhagat⁴

¹ Student, Computer Science and Engineering, SRMCEW, Maharashtra, India, poojadhohne@yahoo.com

² Student, Computer Science and Engineering, SRMCEW, Maharashtra, India, rsheetalgajbhiye@gmail.com

³ Student, Computer Science and Engineering, SRMCEW, Maharashtra, India, tejaswiniwarambhe@gmail.com

⁴ Lecturer, Information Technology, SRMCEW, Maharashtra, India, bhagat.vaishali14@yahoo.in

Abstract

Generally, computer system is handled by the English language only. But the person who is unaware of the English language and structure of query language cannot handle the system. This paper proposed a new approach for accessing the database easily without knowing English. So, the database is accessed with the help of natural languages such as Hindi, Marathi etc. Natural language processing (NLP) is the study of mathematical and computational modeling of various aspects of language and the development of a wide range of systems. Natural Language Processing holds great promise for making computer interfaces that are easier to use for people, since people will be able to talk to the computer in their own language, rather than learn a specialized language of computer commands.

Keywords: NLP, Mathematical modeling, Computational modeling.

1. INTRODUCTION

Natural language processing is a branch of artificial intelligence which includes Information Retrieval, Machine Translation and Language Analysis. The goal of accessing database by natural language processor is to make dataset access easier for the common people. While natural language may be the easiest symbol system to learn and use, it has proved to be the hardest to a computer to master. To access database a user must have the knowledge of Structured Query Language (SQL). Only those users who have the knowledge of these languages can access data or information. So in order to access the information, a graphical user interface is used which requires some basic training for using this system. In India, there are many people who know English but are not fluent enough to formulate queries in it. With the help of this interface an end user can query the system in natural languages like English, Hindi and Marathi etc., and can see the result in the same language.

We are also adding Hindi thesaurus with this application. Thesaurus is such a tool which is important to the country like India where a very large fraction of population is not convenient with language like English and consequently does not have access to the vast store of information that is available. More over Hindi is the official language of India. For Hindi language, the alphabet set is very large. Most importantly this alphabet set and shape characteristics are utilized in the development.

2. EXISTING SYSTEM

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 successes in NLP

since it began. Asking question to databases in natural languages is very convenient and easy method of data access, especially for the users who does not have any knowledge of database queries such as SQL. The success in this area is partly because of the real-world benefits that can come from only NLP system 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.

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.

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 a 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.

3. OBJECTIVE AND AIM

We are going to develop an application that will take the Database queries in the form natural language and then processes it and gives the result. This includes many sub components like Language Analyzer, Query Builder and Viewer. The system will first parses the query in natural language and finds the major parts in the string. Then first it will look for the table name and then it parses the string for the where clause and then for the order by clause. After parsing it will construct the query string based on the data available. The generated SQL query is posted to the database to fetch the results.

4. PROPOSED SYSEM

4.1 Natural Language Interface To Database

The interesting area of Natural Language Processing (NLP) is the development of a natural language interface to database systems (NLIDB). In the last few decades many NLIDB systems have been developed. Through these systems, users can interact with database in a more convenient and flexible way. Because of this, this application of NLP is still very and widely used today. Natural Language Interface has been a very interesting area of research since past times. The aim of Natural language Interface to Database is to provide an interface where user can interact with database more easily using their natural language and access or retrieve their information using the same. We can also say that NLIDB is a system that converts the query in native language into SQL and vice-versa.

4.2 Sub components of NLIDB

There are two sub components of NLIDB

- Linguist components
- Database components

Linguistic Component

It is responsible for translating natural language input into a formal query and generating a natural language response based on the results from the database search.

Database component

It performs traditional Database Management functions. A lexicon is a table that is used to map the words of the natural input onto the formal objects (relation names, attribute names, *etc.*) of the database. Both parser and semantic interpreter make use of the lexicon. A natural language generator takes the formal response as its input, and inspects the parse tree in order to generate adequate natural language response. Natural language database systems make use of syntactic knowledge and knowledge about the actual database in order to properly relate natural language input to the structure and contents of that database. Syntactic

knowledge usually resides in the linguistic component of the system, in particular in the syntax analyzer whereas knowledge about the actual database resides to some extent in the semantic data model used. Questions entered in natural language translated into a statement in a formal query language. Once the statement unambiguously formed, the query is processed by the database management system in order to produce the required data. These data then passed back to the natural language component where generation routines produce a surface language version of the response.

4.3 Architecture of NLIDB

The architecture of NLIDB system is as follows which uses the both semantic and syntactic grammar system architecture.

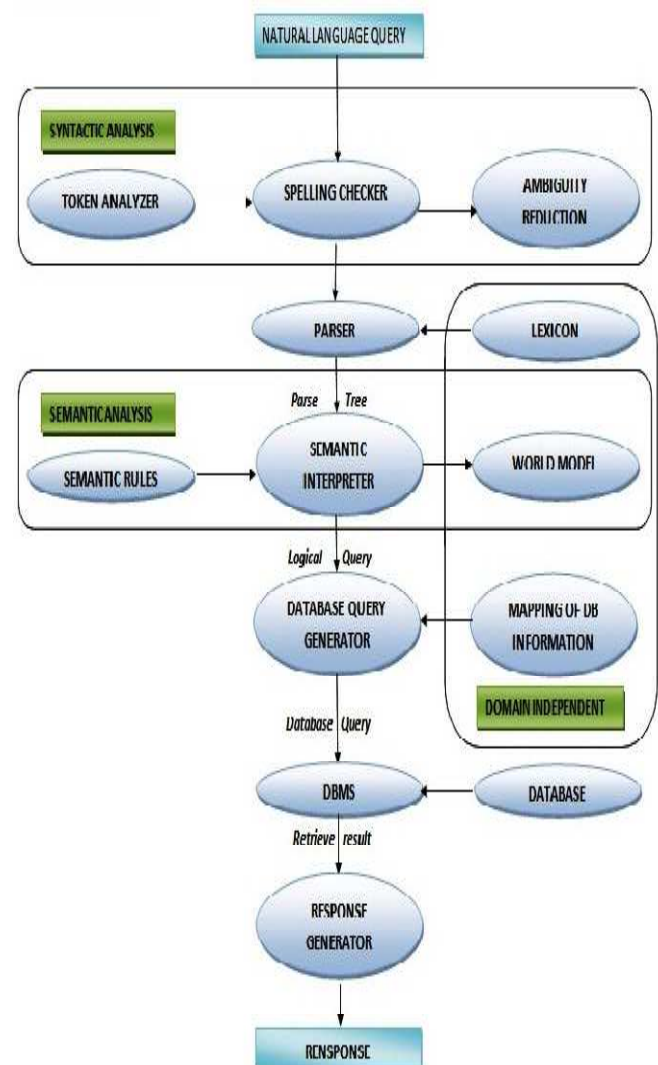


Fig.1: Architecture of NLIDB

The architecture is discussed as follows:

- **Syntactic Analysis:** The objective of the syntactic analysis is to find the syntactic structure of the sentence. This splits the sentence into the simpler elements called Tokens.
- **Token Analyzer:** It split the input string into a sequence of primitive units called tokens that is treated as a single logical unit.

- **Spelling Checker:** The Spelling Checker module makes sure that each token is in the system's dictionary (lexicon) and if this is not the case then the spelling correction is performed or new words are added to the system's vocabulary.
- **Ambiguity Reduction:** This module reduces the ambiguity in a sentence and simplifies the task of the parser.

Parse Tree:

Parse tree is the output obtained from syntactic analysis which represents the syntactic structure of sentence according to some formal grammar. A Parse Tree is a collection of nodes and branches (root node, branch node, leaf node). In a parse tree, an interior node is a phrase and is called a non-terminal or non-leaf node of the grammar, while a leaf node is a word and is called a terminal of the grammar.

For e.g., "List me all employees", the parse tree for this query is shown in figure 2

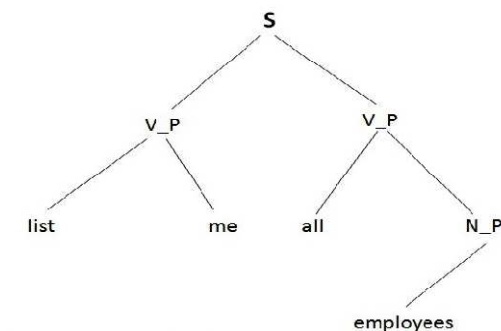


Fig.2: Parse tree

Here, S_ sentence, V_P_ verb phrase, N_P_ noun phrase.

- **Semantic Analysis:** Semantic Analysis is related to create the representations for meaning of linguistics inputs. It deals with how to determine the meaning of the sentence from the meaning of its parts. So, it generates a logical query which is the input of Database Query Generator.
- **Database Query Generator:** The task of the Database Query Generator is to map the elements of the logical query to the corresponding elements of the used databases. The query generator uses four routines, each of which manipulates only one specific part of the query. The first routine selects the part query that corresponds to the appropriate DML command with the attribute's names (*i.e.* SELECT * clause). The second routine selects the part of the query that would mapped to a table's name or a group of tables names to construct the FROM clause. The third routine selects the part of the query that would be mapped to the WHERE clause (condition). The fourth routine selects the part of the natural language query that corresponds to the order of displaying the result (ORDER BY clause with the name of the column).

- **Database Management System:** The purpose of this system is to get the correct result from the database. It executes the query on the database and produces the results required by the user. In Microsoft Word you can look up a word quickly if you right click anywhere in your document, and then to find a synonyms for a specific word, either type the word in the task pane search field or highlight it in your document. Then list of all possible synonyms appear in the context menu. Likewise Hindi Thesaurus work for you.

For example if user select and right clicked on word "Beautiful" then resultant words are shown in the popup menu and synonyms as well as antonyms are listed as shown below in the Fig. 3

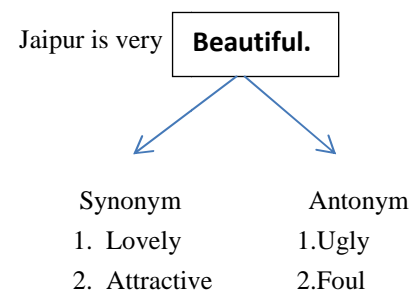


Fig.3: Illustration of English thesaurus example

4.3 User Interface

Following figures shows the user interfaces for accessing database by natural language processor.

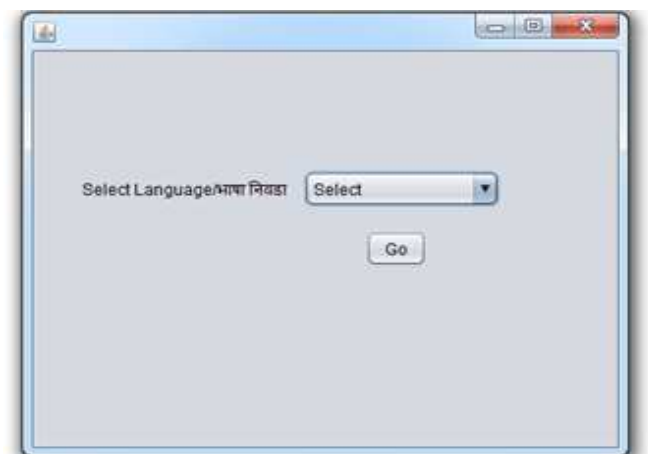
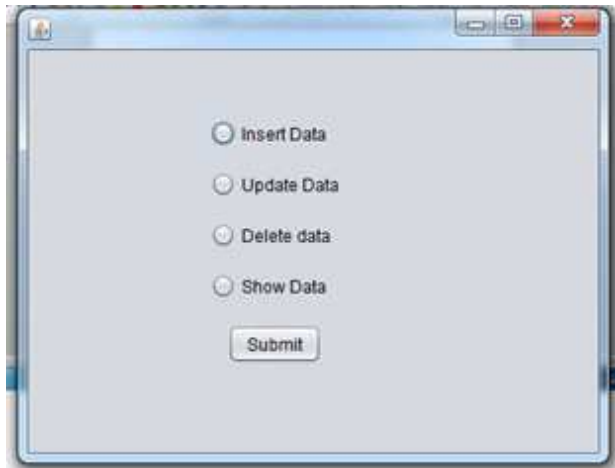
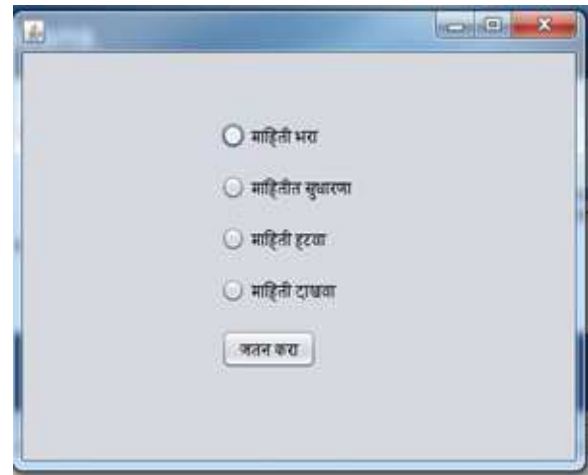


Fig.4: Language selection form



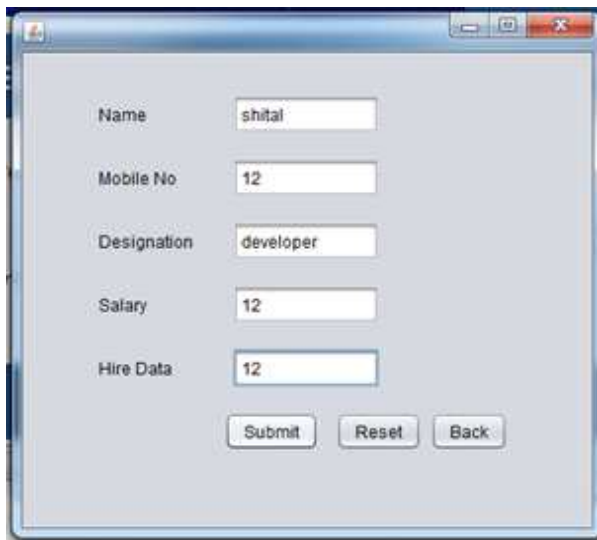
A Java Swing window titled "Operation selection form" with a light blue background. It contains four radio buttons stacked vertically: "Insert Data", "Update Data", "Delete data", and "Show Data". The "Insert Data" radio button is selected. Below the radio buttons is a "Submit" button.

Fig.5: Operation selection form



A Java Swing window titled "Operation selection form in Marathi" with a light blue background. It contains five radio buttons stacked vertically: "माहिती भरा" (Enter Information), "माहितीत सुधारणा" (Modification in Information), "माहिती हटवा" (Delete Information), "माहिती दाखवा" (Show Information), and "नवीन करा" (New). The "माहिती भरा" radio button is selected. Below the radio buttons is a "जतन करा" (Save) button.

Fig.8: Operation selection form in Marathi



A Java Swing window titled "Insert Form" with a light blue background. It contains six text input fields: "Name" (shital), "Mobile No" (12), "Designation" (developer), "Salary" (12), and "Hire Data" (12). Below the input fields are three buttons: "Submit", "Reset", and "Back".

Fig.6: Insert Form



A Java Swing window titled "Insertion form in Marathi" with a light blue background. It contains six text input fields: "नाव" (Name) (पुष्प), "दूरधनी क्रमांक" (Mobile No) (12), "पद" (Designation) (विकास), "पगार" (Salary) (12), and "भरती तारीख" (Hire Date) (12). Below the input fields are four buttons: "जतन करा" (Save), "रिसेट" (Reset), "मराठी अड्डे" (Marathi Address), and "मागे" (Back).

Fig.9: Insertion form in Marathi



A Java Swing window titled "Display form" with a light blue background. It contains six text input fields: "ID" (16), "Name" (anup), "Contact No" (123), "Designation" (java), "Salary" (123), and "Hire Date" (123). There is a "Get Data" button next to the "ID" field. Below the input fields are three buttons: "Submit", "Reset", and "Back".

Fig.7: Display form



A Java Swing window titled "Searching form in Marathi" with a light blue background. It contains one text input field labeled "आयडी" (ID) with the value 49. Below the input field are two buttons: "हटवा" (Delete) and "मागे" (Back).

Fig.10: Searching form in Marathi

5. METHODOLOGY

5.1 Techniques used to develop the Natural Language Interface to Database

There are number of techniques that are used for the development of natural language interface to Database like Pattern Matching System, Syntax Based System, Semantic Grammar System and Intermediate Representation Language.

These techniques are discussed below:

- **Pattern-Matching Systems:** Many NLIDBs were based on pattern-matching techniques to answer the user's questions. The main advantage of the pattern-matching approach is its simplicity *i.e.* no elaborate parsing and interpretation modules are needed, and the systems are easy to implement. These systems cope up even when the query is out of range of sentences in which patterns were design to handle and provide some reasonable answers to them. As a simple illustration of pattern matching technique, consider the following database:

Table 1: Sample database table

Country	Capital	Language
France	Paris	French
Italy	Rome	Italian
India	Delhi	Hindi

If the user asked "What is the capital of India?", using the first pattern rule the system would report "Delhi". The system would also use the same rule to handle question such as "print the capital of India: "could you please tell me what is the capital of India?"etc. ELIZA is among the few systems that plays the role in the above style. ELIZA functions by processing users, by these responses to the scripts. It typically says differently and rephrased the statements of the users as questions and replies the answers of those questions. Mr. Joseph Weizenbaum programmed ELIZA nearly from 1964 to 1966.

- **Syntax-Based Systems:** In syntax-based systems the user's question is parsed (*i.e.* analyzed syntactically) and the resulting parse tree is directly mapped to an expression in some database query language. Syntax-based systems use a language system that explains the feasible syntactic structures of the user's query]. Syntax-based NLIDBs usually interface to application specific database systems that provide database query languages, carefully designed to facilitate the mapping from the parse tree to the database query. Generally, it is hard to design mapping rules that will map the parse tree into some expression directly in a real-life database query language *e.g.* SQL.

- **Semantic Grammar Systems:** In semantic grammar systems, the question-answering is still done by parsing the input and mapping the parse tree to a database query. The difference, in this case, is that the grammar's categories do not necessarily correspond to syntactic concepts. Semantic information about the knowledge domain is hard-wired into the semantic grammar due to this systems based on this approach are very difficult to port to other knowledge domains. For an NLIDB, configured for a new language domain, a fresh semantic grammar has to be written. Semantic grammar categories are usually chosen to enforce semantic constraints. Much of the systems developed till now like LUNAR, LADDER, use this approach of semantic grammar.

- **Intermediate Representation Languages:** Due to the difficulties of directly translating a sentence into general query language using a syntax based approach, the intermediate representation systems were proposed. The logic is to map a sentence into a logic query language followed by the translation of the logical query into a general database query, such as SQL. There can be several intermediate meaning representation languages in the process. Figure 4 shows architecture of an intermediate representation language system.

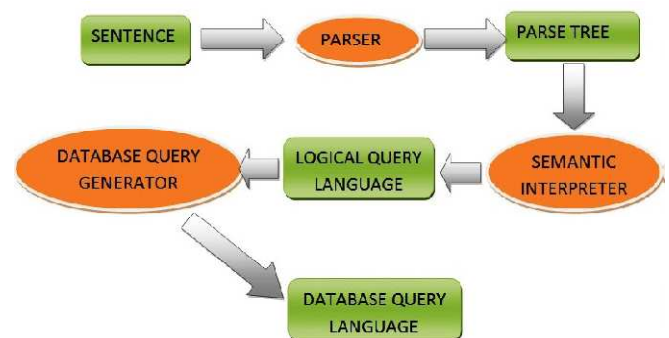


Fig.11: Intermediate Representation Language Architecture

6. ALGORITHM

- Tokenization (scanning)
 - Split the Query in tokens
 - Give order number to each token identified
- Split Query and extract patterns
 - Look for sentence connectors/criteria words
 - Break Query on the basis of connector/criteria tokens.
 - Use criteria tokens to specify condition in query.
 - Find attributes and values after criteria token
- Map value for identified attribute and corresponding table
- Replace synonyms with proper attribute names

- Get intermediate form of Query
- Transform it into SQL

7. ADVANTAGES

The main advantage of NLP is that it relieves the burden of learning syntax. It means there is no need to learn the database languages like SQL and no requirement of any special training before working with the NLP system.

8. CONCLUSIONS

Natural Language Processing can bring powerful enhancements to virtually any computer program interface. This system is currently capable of handling simple queries with standard join conditions. Because not all forms of SQL queries are supported, further development would be required before the system can be used within NLIDB. Alternatives for integrating a database NLP component into the NLIDB were considered and assessed. The system accept an English Language and translate it into SQL .The next aim of our research is to accommodate more and more query. From this research we can seen it is possible to translate a natural language query into SQL. We are also adding Hindi thesaurus with this application for better understanding of user.

ACKNOWLEDGEMENT

We express our deep sense of gratitude and our research guide Vaishali Bhagat for her continuous inspiration and valuable guidance in throughout our dissertation work.

REFERENCES

- [1]. Huang, 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 2010.
- [2]. Read paper from www.elixirpublishers.com ELF software co., "Natural Language Database Interfaces" from ELF software co. November 1999, [online] Available: <http://hometown.aol.com/elfsoft/>.
- [3]. Nlp Algorithms to detect phrases and keywords from text-stack is taken from www.stackoverflow.com.
- [4]. IEEE paper Template in A4(v1)-IJERT from www.ijert.org/browse/volume-2-2013/march-2013 edition.
- [5]. 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.

BIOGRAPHIES



Pooja Dhomne , is pursuing B.E Degree from SRMCEW in Computer Science and Engineering from RTMNU, Maharashtra, India. Her field of interest is JAVA.



Sheetal Gajbhiye , is pursuing B.E Degree from SRMCEW in Computer Science and Engineering from RTMNU, Maharashtra, India. Her field of interest is VB.Net.



Tejaswini Warambhe, is pursuing B.E Degree from SRMCEW in Computer Science and Engineering from RTMNU, Maharashtra, India.. Her field of interest is Computer Networking.



Vaishali Bhagat has received the B.E. degree in Information Technology from RTMNU, Maharashtra, India in 2008 & pursuing M. Tech in CSE from RTMNU. Since 2010, she is working in the department of IT as a lecturer in SRMCEW, Nagpur, Maharashtra, India.