International Journal For Technological Research In Engineering
Volume 5, Issue 2, October-2017 *ISSN (Online): 2347 - 4718*

NATURAL LANGUAGE INTERFACE TO DATABASES

Madhuri Kuthadi
Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad, India

ABSTRACT: In the world of computing, information plays an important role in our lives. One of the major sources of information is database. Database and Database technology are having major impact on the growing use of computers. Almost all IT applications are storing and retrieving the information or data from the database. Database Management Systems (DBMS) have been widely used for storing and retrieving data. However, databases are often hard to use since their interface is quite rigid in co-operating with users. For storing and retrieving the information from database requires the knowledge of database language like SQL. Structured Query Language (SQL) is an ANSI standard for accessing and manipulating the information stored in database. However, everyone may not be able to write the SQL query as they may not be aware of the syntax and structure of SQL and database respectively. The purpose of Natural Language Interface is to allow users to compose questions in Natural Language and receive the response also in Natural Language. The idea of using Natural Language instead of SQL has promoted the development of new type of processing called Natural Language Interface to Database(NLIDB). This paper discuss about an introduction of Intelligent Database System, Natural Language Processing and Natural Language Interface to Database. It also gives a brief overview of subcomponent of NLIDB, techniques used to development of NLIDB along with its architecture.

I. INTRODUCTION

Database Management System is a collection of interrelated data and set of programs to access those data. Database systems are designed to manage large bodies of information. To access this information, we should have the knowledge of Structured Query Language (SQL). Only those users who have the knowledge of these languages can access the data or information [2]. An end user normally doesn't know SQL. So in order to access the information, a graphical user interface has been used. This graphical user interface requires some basic training for using the system. With the help of this interface, the end user can query the system in natural language like English, Hindi, Telgu, etc., and can see the result in same language. This gives the idea of Natural Language Interface to Database (NLIDB). With the help of this interface, the end user can query the system in natural language like English, Hindi, Telgu, etc., and can see the result in same language [3]. NLIDB system is proposed as a solution to the problem for accessing information in a simple way, allowing ideally any type of users, mainly inexperienced ones; to retrieve information from a database (DB) using natural language (NL) [5]. It is a type of computer human interface. Natural Language interface to database is basically a field of natural language processing which has been discussed further. This is the user-friendly interface through which users can interact with the database [9].

II. NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Natural language understanding is the major challenge for any NLP that is enabling the computers to extracts meanings from natural language query. The foundation of NLP lies in number of disciplines like computer and information science, linguistics, mathematics, electrical and electronics engineering, artificial intelligence, robotics psychology, agriculture, weather forecasting, etc [10].

2.1 Major Task of NLP

NLP is a technique which makes the computer understands the languages naturally used by humans [8]. The following is a list of some of the most commonly researched tasks in NLP [10].

Speech segmentation: Given a sound clip of a person or people speaking, separate it into words.

Topic segmentation and recognition: Given a chunk of text, separate it into segments each of which is devoted to a topic, and identify the topic of the segment.

Word segmentation: Separate a chunk of continuous text into separate words. For a language like English, this is fairly trivial, since words are usually separated by spaces.

Speech Recognition: This is a process of mapping acoustic speech signals to a set of words. The difficulties arise due to wide variations in the pronunciations of words.

Natural language generation: It converts the information from the databases into readable human native language.

Machine translation: It translates text from one native language to another automatically.

Natural Language Interfaces to Databases: It allows querying a database using natural language sentences. It is a system that allows the user to access information stored in a database by typing requests expressed in some natural language.

 Copyright 2017.All rights reserved.

Information Retrieval (IR): This is concerned with identifying documents relevant to user's query. Indexing, word sense disambiguation, query modification and knowledge base have also been used in IR system to enhance performance.

Information Extraction: This captures and outputs factual information contained within a document. Similar to IR system, it responds to user's information need. The information need is not expressed as a keyword query instead it is specified as pre-defined database schemas or templates.

2.2 Advantages and disadvantages of NLP

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. We have seen that NLP is very beneficial, but it has some disadvantages also. The disadvantage of NLP is that it requires clarification dialogues with the user to make it more understandable. Sometimes it requires more keystrokes also to perform a task and it may even not show some required context (the circumstances surrounding an event or statement or idea).

III. NATURAL LANGUAGE INTERFACE TO DATABASE

One of most wide and 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 [4]. 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 [5]. We can also say that NLIDB is a system that converts the query in native language into SQL and vice-versa.

3.1 Sub Components of NLIDB

Computing scientists have divided the problem of natural language access to a database into two sub-components:

- Linguistic component
- Database component

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.

3.2 NLIDB's Comparison with Other GUI

The use of NLIDB, however, is much less widespread than it was once predicted, mainly because of the development of alternative graphic and form-based database interfaces. But these alternative interfaces are less natural to interact with and queries that involve quantification, or that require multiple database tables to be consulted are very difficult to formulate with graphic or form-based interfaces, whereas they can be expressed easily in natural language

3.3 Advantages of NLIDB

The NLIDB systems allow the people to communicate with database in much the same way they communicate with each other. The main advantages of Natural language Interface to Database are given follows [7].

No requirement of Artificial Language: One advantage of NLIDBs is that the user is not forced to learn an artificial communication language. Formal query languages like SQL are difficult to learn and master, at least by non-computer specialists.

No need of Training: No special training is required before using the natural language interface. It is highly user friendly and easy to use by the non-expert end users.

Simple and easy to use: The natural language interface in very simple and easy to use because the end users write the query in its native language.

Better for some question: It has been argued that there are some kind of questions (e.g. questions involving negation, or quantification) that can be easily expressed in natural language, but that seem difficult (or at least tedious) to express using graphical or form-based interfaces.

Easy to use for multiple database tables: Queries that involve multiple database tables like are difficult to form in graphical user interface as compared to natural language interface.

3.4 Disadvantages of NLIDB

Many NLIDB systems have been developed so far for business purpose use but use of NLIDB system is not broad-

 Copyright 2017.All rights reserved.

spread and it is not the primary choice for interfacing to database. This lack of acceptance is mainly due to the large numbers of deficiencies which are given below:

Linguistics coverage is not obvious: Currently all NLIDB systems can understand some subsets of a natural language but it is quite difficult to define these subsets. Even some NLIDB systems can't handle certain query belong to their own subsets. This is not the case of formal language like SQL. Because the formal language coverage is obvious and provide the corresponding answers of any statements that follow the given rules [4].

Linguistics vs. conceptual failure: When NLIDB system fails, the system does not give any explanation of what causes the system to fail. Some user try to rephrase the question or just leave the question unanswered [4].

Inappropriate Medium: It has been argued that natural language is not an appropriate medium for communicating with a computer system. Natural language is claimed to be too verbose or too ambiguous for human-computer interaction [7]. NLIDB users have to type long questions, while in form-based interfaces only fields have to be filled in, and in graphical interfaces most of the work can be done by mouse-clicking. In natural language interface user has to type full sentence with all the connecters (articles, prepositions, etc) but in graphical or form based interfaces it is not required [7].

IV. TECHNIQUES USED TO DEVELOPED 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 was 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

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

A primitive patter-matching system according to [6] may use riles as:

Pattern: … "Capital" … <country>

Action: Report CAPITAL of row where COUNTRY = <country>

Pattern: … "Capital" … <country>

Action: Report CAPITAL and COUNTRY = <country> of each row

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 [7]. 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 [4]. 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 [4]. 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 [7].

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 [7]. 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 [7]. 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 [4]. 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 5.1 shows a possible architecture of an intermediate representation language system.

 Copyright 2017.All rights reserved.

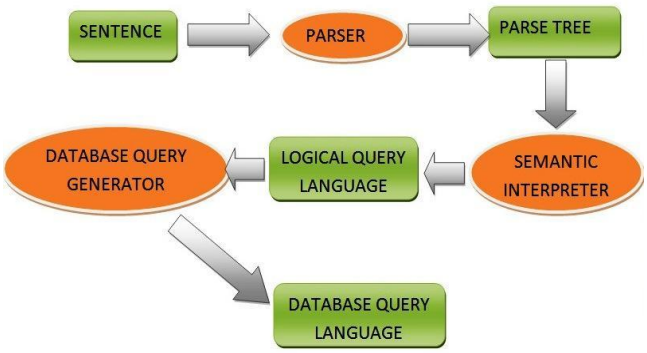

Figure 5.1 Intermediate Representation Language Architecture

V. ARCHITECTURE OF NATURAL LANGUAGE INTERFACE TO DATABASE

The most commonly used architecture of NLIDB system shown is in Figure 6.1 which uses the both semantic and syntactic grammar system architecture. The common architecture of NLIDB system is discussed below.

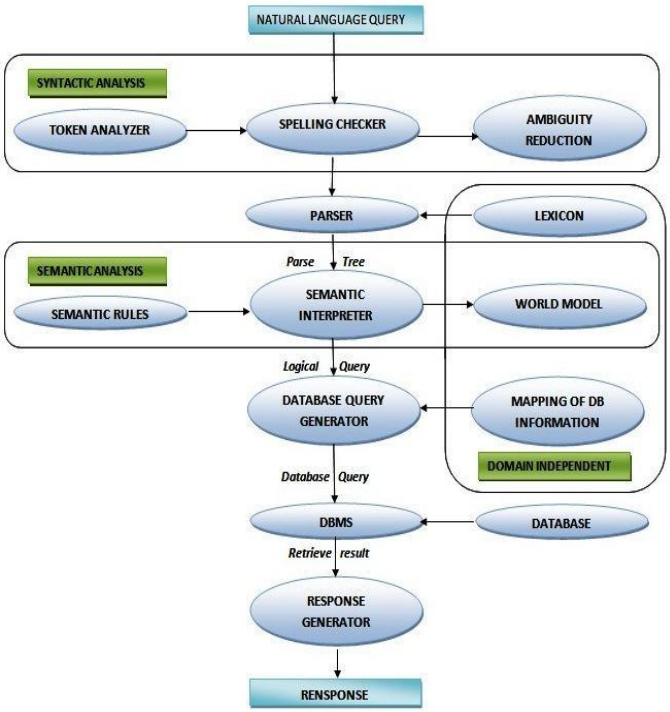

Figure 6.1 commonly used Architecture of NLIDB

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 of obtained from syntactic analysis which is 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 6.2.

Here, S→ sentence, V_P→ verb phrase, N_P→ noun phrase.

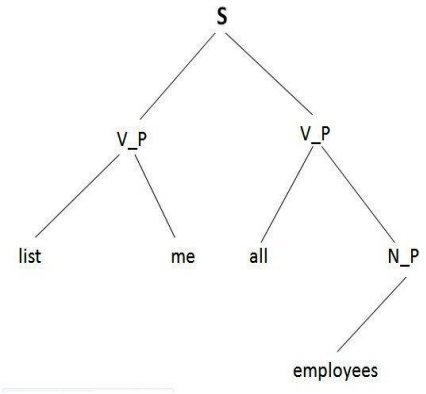

Figure 6.2 Parse Tree

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.

VI. CONCLUSION

In the last few decades many Natural Language Interface to Database systems (NLIDB) have been developed through which user can interact with the database in a more convenient and flexible way. The NLIDB systems developed so far are basically used for business purpose. The use of NLIDB system is not broad-spread and it is not primary choice for interfacing to database. This lack of acceptance is mainly due to the large number of deficiencies in the NLIDB system in order to understand a Natural Language. In the

 Copyright 2017.All rights reserved.

future, work could be done to improve the linguistic coverage by the NLIDB system. Use of NLIDB system could be made easier by avoiding the users to type long questions and allowing the system to display proper error message in case of failure of any query.

REFERENCES

[1] Niculae Stratica, Leila Kosseim and Bipin C. Desai, "NLIDB Templates for Semantic Parsing", Department of Computer Science Concordia University 1455 de Maisonneuve Blvd. West Montreal, H3G 1M8, Canada.

[2] Siasar djahantighi F, Norouzifard M, Davarpanah S H, Shenassa M H. Using natural language processing in order to create SQL queries. In: IEEE International Conference on Computer and Communication Engineering (ICCCE); 13-15 May 2008; Kuala Lumpur, Malaysia: IEEE. pp. 600 - 604.

[3] Li H, Shi Y. A WordNet-based natural language interface to relational databases. In: IEEE 2nd International Conference on Computer and Automation Engineering (ICCAE); 26-28 Feb. 2010; Singapore: IEEE. pp. 514 – 518.

[4] Mrs. Neelu Nihalani, Dr. Sanjay Silakari, Dr. Mahesh Motwani. "Natural language Interface for Database: A Brief review", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011 ISSN (Online): 1694-0814.

[5] Mrs. Neelu Nihalani, Dr. Sanjay Silakari, Dr. Mahesh Motwani. "Natural language Interface to Database using Semantic Matching", International Journal of Computer Application, Vol. 31, no.11, Oct. 2011 ISSN: 0975 – 8887.

[6] Androutsopoulos, G.D. Ritchie, and P. Thanisch, Natural Language Interfaces to Databases – An Introduction", Department of Computer Science, University of Edinburgh, King's Buildings, May Field Road, Edinburgh EH's Buildings, May Field Road, Edinburgh EH9 3JZ, Scotland, U.K. , Mar.1995.

[7] Kaur, and P. Bhatiya, "Punjabi Language Interface to Database", M.tech thesis, Department of CSED, Thapar University, 2010.

[8] Arati K. Deshpande, and Prakash. R. Devale, "Natural Language Query Processing Using Probabilistic Context Free Grammar", International Journal of Advances in Engineering & Technology, May 2012.,ISSN:2231-1963.

[9] Owda M, Zuhair B, Crockett K. Conversation-Based Natural Language Interface to Relational Databases. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops; 5-12 Nov 2007; Silicon Valley: IEEE/WIC/ACM. pp. 363 – 367.

[10] en.wikipedia.org/wiki/natural_language_processing.

 Copyright 2017.All rights reserved.