International Journal of Engineering Sciences & Emerging Technologies, Jan. 2016.
ISSN: 2231–6604 Volume 8, Issue 4, pp: 208-212©IJESET

CONVERSION OF NATURAL LANGUAGE QUERY TO SQL

Prabhdeep Kaur[1] and Shruthi J[2]
[1]Student, [2]Assistant Professor,
Department of Computer Science & Engineering,
BMS Institute of Technology, Bangalore, India

ABSTRACT

Most of the IT applications require storing and retrieving information from databases. The data retrieval requires the knowledge of database languages like SQL. Using Natural Language Processing the user submits a query as speech signal through the user interface and gets the result of the query in the text format. Hence the aim of NLP is to enable communication between people and computers without resorting to memorization of complex queries and procedures. Acoustic and Language models are used to convert speech utterance to English text query and Natural Language Processing techniques are applied on English text query to generate SQL query. This translation uses lexical analyser, parser and syntax directed translation technique. This system could also handle complex queries along with the basic queries asked by the user in the form of speech. Adding a data dictionary like Thesaurus is another suggestion, which could help automating the synonymous words during the semantic analysis.

KEYWORDS- *Natural Language Processing (NLP), Speech, SQL, English text query, Natural Language Interface for Database (NLIDB)*

I. INTRODUCTION

The main purpose of Natural Language Processing is that the computer must be able to interpret a speech signal and perform the required action. For casual users who do not understand database query language like SQL, an easy method of data access is asking questions to databases in natural language. Automatic speech recognition (ASR) is becoming more famous and hence is used widely in many applications. Here users can interact with the database with their voice for retrieving details from the database. Hence it is not necessary for the user to have prior knowledge about the SQL queries.

NLP is a technique that makes the computer understands the languages used by humans. While natural language may be easy for people to learn and use, it has been proved to be hard for a computer to master. Despite such challenges, natural language processing is regarded as a promising and important endeavour in the field of computer research. The goal is to inculcate the computer with the ability to understand and generate natural language so that user can address his computer through text as though he was addressing another person.

This system is concerned with the solution of the problems arising in the analysis or generation of Natural language speech, such as syntactic and semantic analysis or compilation of dictionaries and grammars necessary for such analysis. Two Different approaches are proposed for interfacing database to natural language query, one is the statistical technique and another one is classical rule based technique. The statistical technique requires large corpus of data to train the system in order to process the natural language query. The classical rule based technique is used for interfacing database to natural language query as it uses the knowledge of underlying database. Classical rule based technique can be much faster than the statistical technique.

The system generates Lex file automatically which is used while tokenizing the words involved in English text query and since Lex file contains underlying database information like column and table names so automatic generation of Lex file helps in making the System database independent. Main

objective of NLIDB is to accept the query sentence and try to understand it by applying lexicon, syntactic and semantic analysis and then convert it into SQL. Natural Language Interface for Database deals with structured text which has been parsed, also its entities and attributes have been identified before.

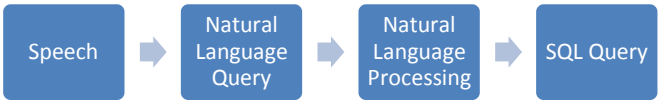

Fig 1: Problem description

Suppose if we want to view information of a particular employee from EMP table then we are supposed to use the following SQL query[7]: SELECT * FROM EMP WHERE e_name =’ABC’; But a person, who have knowledge of SQL, will not be able to access the database unless he knows the syntax of firing a query to the database. But with the use of NLP, this task of accessing the database will be much simpler. So if the user gives the speech signal as: “Give the information of employee whose name is XYZ”. The output would be displaying the information about the employee named XYZ. Both the statements would result in the same output.
The system parses the query and with the help of the dictionary, carries out different phases like morphological analysis, syntactical analysis, and semantic analysis and finally generates the SQL query. The system is independent of database i.e. it can be configured automatically for different databases; it only needs underlying database connection information for the configuration and underlying database schema.

II. SYSTEM DESIGN

There are six phases for conversion of speech into SQL. In first phase speech is converted into English text using speech recognition, in second phase the text is analysed whether it is syntactically correct or not based on grammar rules for valid queries, in third phase the text is mapped into an intermediate query using lexer, parser and syntax directed translation, in fourth phase we extract the clauses from the intermediate query, in fifth phase we find all the required tables and thus SQL query is formed, in sixth phase formulated SQL query is fired to database and thus obtaining the result.

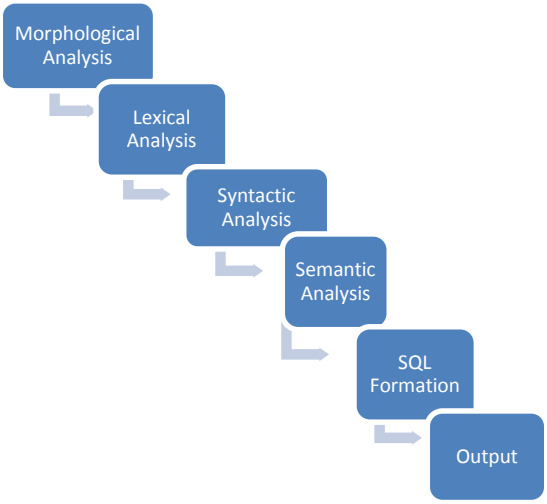

Fig2: System design

2.1 Morphological Analysis

During the morphological analysis, the sentence is broken down into tokens i.e. the smallest unit of meaning. For example, if the given input is ―find USN of Prabhdeep, then in this phase, each word

of the sentence, i.e. find, USN, of, Prabhdeep will be stored in a list. Individual words are analysed into their components and non-word tokens such as punctuation are separated from the words.
This analysis makes use of acoustic as well as language model. The acoustic model is a type of network of states such that it can identify a possible word on incoming of the input speech signal. Since many people speak a same word in different quality of voices so it is not always possible to get a perfect match for the input speech signal i.e. for a spoken word if need identify the phoneme sequence of that word there would be a possibility that we could not match an exact state in the model. Due to this, the acoustic model has been designed using the concept of Hidden Markov Model (HMM). Hidden Markov model is used in a system where a state of the system is not observed perfectly i.e. there is a hidden state that we can't explore perfectly using the submitted information[3]. To improve the accuracy of word recognition, a language model is used which is developed by defining grammar rules. For a speech signal there might be more than one possible word which has been generated using HMMs.

2.2 Lexical Analysis

Lexical Analysis is the first state in translation. It generates the tokens for the requested words as their types and then handover to the parser for further processing. Each word of the tokenized sentence is mapped with the meaning of the same word present in the data dictionary. There is an availability of various types of lexical resources to select such as WordNet. WordNet as a universal dictionary of language is an online lexical reference system that is used to identify properties of word such as nouns, verbs, adjectives, adverbs, synonyms. For example, from the list generated in the morphological phase, the words will be mapped as shown―give: select, ―salary: salary, ―employee: employee.

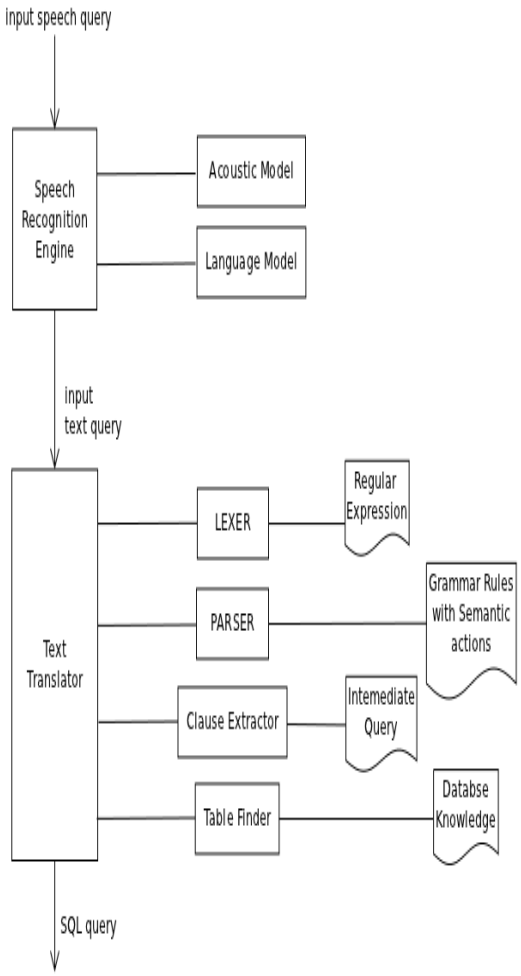

Fig3: Components of the system

2.3 Syntactic Analysis

For Syntactic analysis the attributes present in the given input query from the words generated in the lexical phase is found out. And then the tables which contain the attributes of the given input query are found out. For example, the output generated in the lexical phase, we derive the attributes in the query as ―salary and which belongs to table ―employee.
The Syntax analyser validates the English Text query whether the input query is syntactically correct or not. It validates the query by building a parse tree for the input English sentence. The parse tree is built with the help of the grammar rules also known as production rules. While building the parse tree Syntax analyser also uses the concept of syntax directed translation to perform semantic analysis at the next step. With the help of syntax directed translation the input English query is converted into an intermediate representation that is further processed in the next phase to generate final SQL query.

2.4 Semantic Analysis

Semantic analysis focuses on the study of meaning of the words present in the natural language query and what do they actually stand for. This level deals with checking the different conditions like WHERE clause, relational operators, aggregate functions, natural JOIN and build the SQL query accordingly. The semantic database helps us to present an equal query for different sentences. As for the following example-
Who is (are) the author(s) of the paper(s) ‘‘Conversion of Natural Language Query to SQL”
Who is (are) the writer(s) of the paper(s) ‘‘Conversion of Natural Language Query to SQL”
Who is (are) the author(s) of the resource(s)
‘‘Conversion of Natural Language Query to SQL”
Who is (are) the writer(s) of the resource(s)
‘‘Conversion of Natural Language Query to SQL”?
The pre-processor reads the database and identifies entities and their attributes and finally creates a list of synonymous and similar words using WordNet. After getting the SELECT and WHERE clauses the SQL query is still not complete as the FROM clause is not there in extracted query.

2.5 Table Finder

This phase finds the FROM clause which comprises of all table names which are required to fire the SQL query on the underlying database If there are more than one table names that are to be accessed then we have to perform the JOIN operation on all those tables. The join operation is performed on the basis of common attributes between two tables. So after performing all join conditions we append these conditions in WHERE clause and our SQL query is formulated.
For example, we can consider following SQL query pattern[5]:
SELECT attribute FROM entity WHERE
Default attribute = <value of entity.default_attribute>

2.6 SQL Generator

So after forming all join conditions we append these conditions in WHERE clause. Thus the final SQL query is generated which is used to retrieve the required information from the database.

III. FUTURE WORK

We suggest a system that deals with more complex queries that require to access more than one table for obtaining the solution to the questions asked by the user. The complex queries that can be considered are queries that make use of GROUP BY and HAVING clauses.
The query containing GROUP BY clause will gather all the rows together that contain specific data and thus will allow aggregate functions to be performed on the columns. This is shown by the following syntax:

```
SELECT column1,
SUM(column2)
FROM "list-of-tables"
GROUP BY "column-list";
```

Thus this clause partitions the relation into non-overlapping subsets. The HAVING clause allows to specify conditions that filter which group of results appear in the final results. The following shows how the HAVING clause has to be used:

```
SELECT column1, column2
FROM table1, table2
WHERE [conditions]
GROUP BY column1, column2
HAVING [conditions]
ORDER BY column1, column2
```

Before the final output is displayed, any specific conditions on the group functions in the HAVING clause are applied to the grouped rows.

We could also use Thesaurus for semantic analysis as it is a reference work that lists words grouped together according to similarity of meaning. It also contains more updated words and thus reduces the chance of ambiguities to arise.

IV. CONCLUSON

Natural Language Processing can change the complete working of the computer programming interface. The system uses speech recognition models in association with classical rule based technique and semantic knowledge of underlying database to translate the user speech query into SQL. The user does not have to know the table names, instead the system finds them out with the help of attributes specified in query. In future we can extend to formation of more complex SQL queries to improve the efficiency and accuracy of existing architecture.

REFERENCES

[1] S.Nareshkumar, N.Mariappan, K.Thirumoorthy, "Database Interaction Using Automatic Speech Recognition" International Journal of Innovative Research in Science, Engineering and Technology Volume 3, Special Issue 3, March 2014

[2] Prasun Kanti Ghosh, Saparja Dey, Subhabrata Sengupta " Automatic SQL Query Formation from Natural Language Query" International Journal of Computer Applications (0975 – 8887) International Conference on Microelectronics, Circuits and Systems (MICRO-2014)

[3] Sachin Kumar, Ashish Kumar, Dr. Pinaki Mitra, Girish Sundaram "System and Methods for Converting Speech to SQL" International Conference on "Emerging Research in Computing, Information, Communication and Applications" ERCICA 2013 pp: 291-298, Published by Elsevier Ltd.

[4] Axita Shah, Dr. Jyoti Pareek, Hemal Patel, Namrata Panchal "NLKBIDB - Natural Language and Keyword Based Interface to Database" 978-1-4673-6217-7/13/$31.00_c 2013 IEEE.

[5] F.Siasar djahantighi1, M.Norouzifard1, S.H.Davarpanah2, M.H.Shenassa "Using Natural Language Processing in Order to Create SQL Queries" Proceedings of the International Conference on Computer and Communication Engineering 2008 May 13-15, 2008 Kuala Lumpur, Malaysia.

[6] Filbert Reinaldha, Tricya E. Widagdo "Natural Language Interfaces to Database (NLIDB): Question Handling and Unit Conversion" 978-1-4799-7996-7/14/$31.00 ©2014 IEEE.

[7] Gauri Rao, Chanchal Agarwal, Snehal Chaudhry, Nikita Kulkarni, Dr. S.H. Patil "natural language query processing using semantic grammar"(IJCSE)International Journal on Computer Science and Engineering Vol.02, No02, 2010, 219-223