IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

SQL Query Formation Using Natural Language Processing (NLP)

Prof. Debarati Ghosal[1], Tejas Waghmare[2], Vivek Satam[3], Chinmay Hajirnis[4]

Professor, Dept. of Information Technology, Vidyalankar Institute of Technology, Mumbai, India[1]

BE Student, Information Technology, Vidyalankar Institute of Technology, Mumbai, India [2,3,4]

Abstract: While working on normal database system, to retrieve data from database we have to know about the SQL Query language to retrieve exact data from the database. But everyone doesn't have exact knowledge about the SQL Query language. For retrieving data from the database they have to enter the correct SQL Query. But without having any knowledge about SQL Query, they are unable to retrieve the data of their choice. To overcome this, we are doing our project on SQL Query formation using Natural Language Processing (NLP). This project aims at developing a system which will accept English query from user and convert it into SQL. This helps novice user who can easily get required contents without knowing any complex details of SQL languages. We can store huge amount of data in databases, but casual users who don't have any technical background are not able to access the data. Hence, there was a requirement for personnel with knowledge of SQL to retrieve data from the databases. So this paper proposes system that will convert English statement given by user to all possible intermediate queries so that user can select appropriate intermediate query and then system will generate SQL query from intermediate one. Finally system will fire SQL query on database and gives output to user. When an interpretation error occurs, users often get stuck and cannot recover due to a lack of guidance from the system. To solve this problem, we present a natural language query processing framework.

Keywords: NLP, SQL, Morphological, Lexical, Syntactic, Semantic.

I.INTRODUCTION

While natural language may be the easiest system for people to learn and use, it has proved to be the hardest for a computer to understand. 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.

In this project, we are translating English query into a SQL query using semantic grammar. The system will accept user's query in natural language as an input. The program will check whether the query is valid or not. Then we will generate tokens by performing the division of the question clause. Each token represents a single word in the user's query. The tokens from the query clause are compared with clauses already stored in the dictionary. The dictionary needs to be constantly updated. Then the algorithm scans the tokens and tries to find attributes present in the query. Then we find all the tables in the database which contain the attributes by comparing syntax and semantics.

Then we build the final SQL query and execute it on the database and return the result dataset to the user.

II. LITERATURE SURVEY

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.

In 1950, Alan Turing published his famous article "Computing Machinery and Intelligence" which proposed what is now called the Turing test as a criterion of intelligence. This criterion depends on the ability of a computer program to impersonate a human in a real-time written conversation with a human judge, sufficiently well that the judge is unable to distinguish reliably - on the basis of the conversational content alone - between the program and a real human.

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.

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.

III. SCOPE

To work with any RDBMS one should know the syntax of the commands (SQL). The Natural language processing is done in English i.e., the input statements have to be in English language. Input from the user is taken in the form like what, who, where, etc or find.

A limited Data Dictionary is used where all possible words related to the 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. The system and can be operated by people with average knowledge.

IV.PROPOSED SYSTEM

When user opens system he/she has to first login to the system. This is done to ensure that no unauthorized person should be able to retrieve data or modify the data in the database. User can ask queries to database in "give....", "show...", "i want...", "find..." formats. Our system also provides facility to update tables in database. User can insert values into tables.

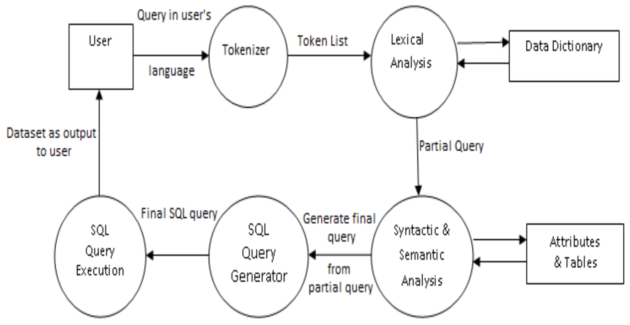
The user's query is processed step by step. There are four main steps or levels involved in conversion of natural language queries to SQL queries. These steps are known as Levels of Language, also known as Synchronic Model of Language.

1. Morphological Analysis:
In this phase, the sentence is broken down into tokens. Each word in the user's query is a separate token, i.e. we split the given input query sentence in natural language into all the words it contains and store the words in a list. For example, if the given input query is ―find salary of the employee, then in this phase, each word of the query, i.e. find, salary, of, the, employee will be stored in a list like {'find', 'salary', 'of', 'the', 'employee'}.

2. Lexical Analysis:
After the query has been broken down into tokens, the system will interpret the meaning of individual words. Each word will be mapped with the meaning of the same word present in the data dictionary. For example, from the tokens generated in the morphology phase, the words will be mapped as, find-select, salary-salary, employee-employee. The data dictionary will need constant improvements to ensure that the words in the user's query are mapped correctly.

3. Syntactic Analysis:
After mapping the words, we will find all the attributes present in the given input query from among the words generated in the lexical phase. Each of these attributes is checked with the attributes in the dictionary which contains all the tables along with their attributes. And then we can find the tables in the database which contain the attributes of the given input query. For example, for the given input query, we derive the attributes in the query as ―salary and which belongs to table employee.

4. Semantic Analysis:
Semantics focuses on the study of meaning of the words and the relation between words, phrases and what do they actually stand for. Linguistic semantics deals with the study of meaning which interprets human expression through language. In this stage we would check the different conditions like where clause, relational operators, aggregate functions and build the SQL query accordingly. The final SQL query for the given input query, after checking all the conditions, would be, 'select salary from employee'. Since there are no conditions mentioned in the user's query, we do not require a where clause in the corresponding SQL query.



V. ALGORITHM

- The first step in query formation is to process the input query, i.e. divide the user's query into tokens, which are individual words in the query
- Then replace the starting of the query with appropriate syntax. Identify the words in the user's query which are column names (attributes), the words which represent the name of the tables and values, if any, present in the user's query.
- Replace synonyms of column names or table names with the actual attributes or table names.
- Construct the query as follows:
    - Identify the attributes which the user wants to retrieve. This will be appended to the select keyword.
    - Identify the table to which these attributes belong. This will be appended to the from keyword.
    - Identify the conditions or values, if any, specified by the user in his/her query.
- If there is only one table to which all the attributes belong, there is no need of a join. Otherwise, perform join operation on the two tables using primary key of table 1 and attribute in table 2 which is foreign key of table 1. If any value has been specified by the user, concatenate the attribute with '=' and the value.
- Generate the final query and fire it on the database to get the required result which will be displayed to the user.

VI.PROS AND CONS

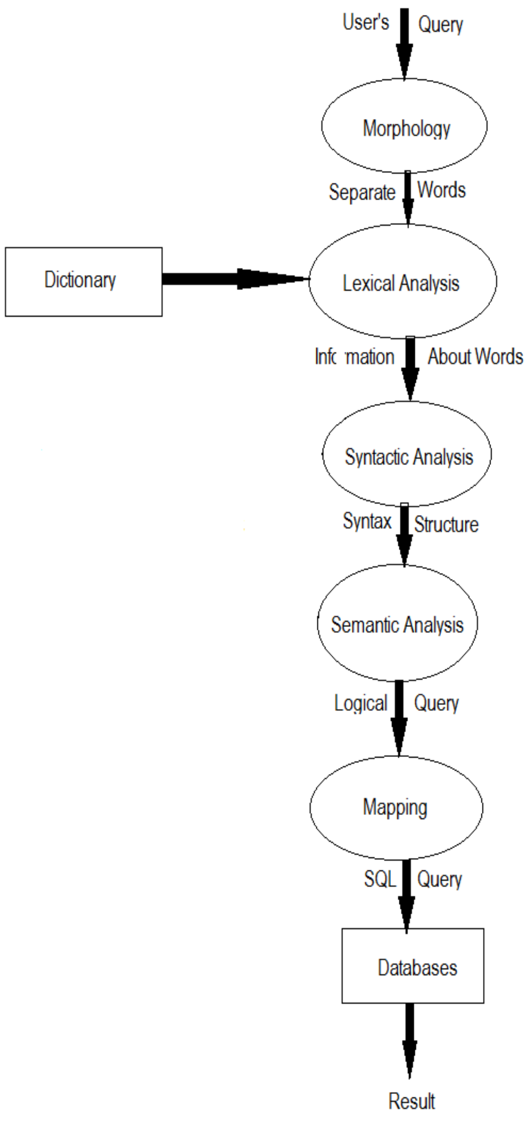
PROS

1. No prior knowledge of complex DBMS languages required
2. Any person who knows English Language can use this system.
3. Simple User Interface.
4. Efficient and Fast system with respect to response.
5. Can be used in various environments.
6. Easy to expand.

CONS

1. May need more attention, time and coding to handle complex sentences.
2. Since it is first version it may need to handle few bugs
3. multiple sentences may take some more time for processing for generating out

VII. FLOWCHART



VIII. CONCLUSION

Use of Natural Language brings ease for any human being. This system helps user to easily retrieve data from database using simple English language. The user need not learn complex query language like SQL.

We can add more synonyms for column names and table names so that system is able to handle more queries. This system provides some recommendations so that it is helpful for user. In future we can add some strong recommendation framework in this system so that user will have to take fewer efforts. The system also stores the successfully executed queries.

This system uses static database so if we want to add any other table in database we also have to add grammar to handle queries for that table as grammar is hard coded but we can also remove this problem by constructing a dynamic framework in which user can dynamically add new tables and remove older ones. In this architecture we have to generate grammar dynamically which can be future enhancement for this system.

REFERENCES

1. Natural Language To SQL Conversion System, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) Vol. 3 Issue 2, June 2013, 161-166.
2. Automatic SQL Query Formation from Natural Language Query, International Journal of Computer Applications.
3. Huangi,GuiangZangi, Phillip C-Y Sheu —A Natural Language database Interface based on probabilistic context free grammar, IEEE International workshop on Semantic Computing and Systems 2008.
4. A Survey of Natural Language Query Builder Interface to Database, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5 Issue 4, 2015.
5. Gauri Rao(IJCSE) International Journal on Computer Science and Engineering.