International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

Natural Language Interface to Database Using Modified Co-occurrence Matrix Technique

V. Thulasinath
M.Tech, CSE Department, JNTU College of Engineering Anantapur, Andhra Pradesh, India

ABSTRACT

Today, database is considered as one of the major source of information. Data stored in database can be accessed by using SQL queries, Those who are expert in SQL language can access information from database but non-technical user cannot retrieve data from database such as MySQL and oracle, Everyone is not able to write SQL queries as they may not be aware of the structure of the database, So this has led to the improvement of interface s to Database System. This is useful for non-expert users to query relational databases in their natural language. However, this project accepts a query entered in natural language (i.e., English) for accessing/retrieving database. This statement will be verified syntactically and then produces equivalent statement in Structured Query Language (SQL). Finally SQL statement will be executed by query executor to produce desired information from database. In proposed system we focused on design and implementation of a system using modified word co-occurrence matrix method which will provide access to database using queries in English language. As a result intelligent natural language interfaces to databases to be developed.

Keywords : Information retrieval; Natural language interface; Natural language processing; Structured Query Language (SQL); Word co-occurrence matrix technique.

I. INTRODUCTION

In current computing world, computer based information technologies have been generally used to several organizations, private companies, academic and education institutions to control their processes and information systems. Databases are extensive element in private and public information systems which are essential in various number of application areas. A general information management system is able to managing several kinds of data, stored in the database are known as Database Management System (DBMS). The information constantly stored in Database Management System, it became huge value data through relational databases. To retrieve information from a database, such make a standard query in way that the computer will understand and produce the expected output, almost all languages for relational database systems fallowed by Structure Query Language (SQL) norms. Structured Query Language (SQL) is an ANSI standard for accessing and modifying the data stored in relational databases. It is effectively employed in industry and is supported by major database management systems (DBMS). In recent times, there is an issues rising for non-expert users to query relational databases in a more natural language besets linguistic variables and terms, instead of operating on the values of the attributes. The new type processing method has been introduced for using natural language instead of SQL this process is called Natural Language Interface to Relational Database Systems (NLIRDB). NLIRDB is an approach that develops the interface to the database systems to enhance the users with feasible performance query processing in databases. The aim of research in natural language processing (NLP) is to provide user friendly environment to interact with computer in natural language such as English. Natural Language Interfaces to Database system (NLIDB) is field under NLP where data from relational database is accessed using questions in natural language where computers are used to store and process data when needed. The SQL language professionals can access database easily but non-technical users cannot. So the NLIDB system can provide platform to non-technical user to communicate with computer using natural language (E.g. English). The system which we have to provide development using JAVA language, MySQL and MS Access database. The developed system transforms natural language query to structured query language (SQL) and retrieves result from database.

II. LITERATURE SURVEY

A literature survey or literature review is ground study of particular project or specific subject. Study of references papers and old algorithms that we have read for designing the proposed methods. It also helps in reporting summarization of all the old references papers, their drawbacks. The detailed literature survey for the project helps in comparing and contrasting various methods, algorithms in various ways that have implemented in the researchThe main intention of NLP researchers is expertise on how human beings realize and use language, in order that appropriate tools and techniques can be developed to make computer system understanding and manipulate common languages to participate in the desired tasks. A number of researchers have tried to come up with expanded technology for performing lot of activities that form important components of NLP works. Jadhav Sneha, Raut Shubhangi, A.S.Zore, [1] introduces to Natural Language to Database interface where information is extracted from the database just by entering query in Natural Language. To retrieve the information from database one has to know the structure of database languages like SQL. Everyone is able to write SQL queries since they may not have the knowledge of database. And this has lead to the developing such a system where non-expert users compose their questions in their natural language and get the results. So here new concept is introduced to develop new type of processing called interface between natural language to database. Natural language to database interface enhances the users in performing flexible querying in database the new framework has been designed for store an intermediate processed data introducing new knowledge can be issued with simple SQL insert statements on top of the processed data, on the other hand existing extraction framework do not provide the capabilities of managing intermediate processed data. In this framework is most suitable for performing extraction on text written in natural sentences.

Neelu Nihalani, Dr. Mahesh Motwani, Dr. Sanjay Silakari [2] defines mapping of natural language queries to SQL. They were proposed a General architecture for an intelligent database interface and also a real implementation of such a system which can be connected to any database. One of the main characteristics of this interface is domain-independence, which means that this interface can be used with any database. Another characteristic of this system is ease of configuration. The intelligent interface employs semantic matching technique to convert natural language query to SQL using dictionary and set of production rules. The dictionary consists of semantics sets for tables and columns. The main advantage of the system is natural language is used for querying the database and Incorporated into the existing database systems. The presented system accepts flexible user queries and converts them into a standard SQL query. Expression mapping, stop words removal and semantic matching techniques have been utilized by the intelligent layer in the formation of the SQL query.

Anh Kim Nguyen, Phuong Hong Nguyen [3] study on constructing a natural language interface to relational databases, it accepts natural language fuzzy questions as inputs and generates answers under the form of tables or short answers. The question is parsed using a semantic grammar and then, it is translated into a SQL query using various translation rules. End of the result is the database management system is left to find the output in tables form from Relational data base with its own specialized optimization and Planning, The system accepts quantified questions and negative questions, which are very difficult to express in SQL syntax by non-expert users. The system can assist users to rephrase questions correctly to his/her intention. The system is portable to other domains. When applying to other domains, they only need to modify the domain-dependent dictionaries and knowledge sources.

III. MOTIVATION

Retrieving data from database requires knowledge of database languages like SQL. However, everyone should not know the awareness of the database query language, for that purpose to improve the natural language interface to the relational database. This leads to the development of interfaces to database system

IV. SCOPE

The proposed system is used to make the interface between the natural language and the query language. We make the natural language queries, then the interface generates related SQL query to access the data from the database. In the design we implement a system using modified co-occurrence matrix technique which will provide access to database using queries by the natural language.

V. EXISTING SYSTEM

The present system is a console-based system for each and every database separately. The user has to learn how to use that console to connect to a particular database and how to work on console to execute query statements. In the real time project development each and every user has a need to connect to different kinds of the databases that are at different locations (systems). Whenever the user wants to connect to a database then the client libraries that are required to connect to that database server has to be installed at the client. This process repeats in each and every user system. In the current system the queries are in high level languages like SQL. The person who is using that system must learn the SQL and write the queries in the High level languages. The existing system works a lot of man-hours both for learning how to use console to connect to different kinds of query statements. To connect to a database from a user system needs specific database client libraries has to install at the user system. Learn and study about a particular database console is time taking process and that console is used to connect to one database only. The drawbacks in the existing system are as follows

•Adjusting the console to display the results effectively
•Repeated execution of the same statements
•Can't use undo, redo, cut, paste options effectively from consoles.

VI. PROPOSED WORK

This system proposes a general architecture for a database interface and also a real implementation of such a system which can be connected to any database. One of the main characteristics of this interface is domain-independence, which means this interface can be used with any database. Another characteristic of this system is ease of configuration. The proposed system uses semantic matching technique for convert natural language query to SQL query using predefined dictionary and some kinds of rules. Making questions to databases in natural language like English is a very acceptable and easy method of data access from database system especially for non technical users, here interface module converts user Query into a corresponding SQL query. Natural language queries are given as input to the interface module. Then the interface module generates related SQL query to access the data from the database.

Database Query Generator:

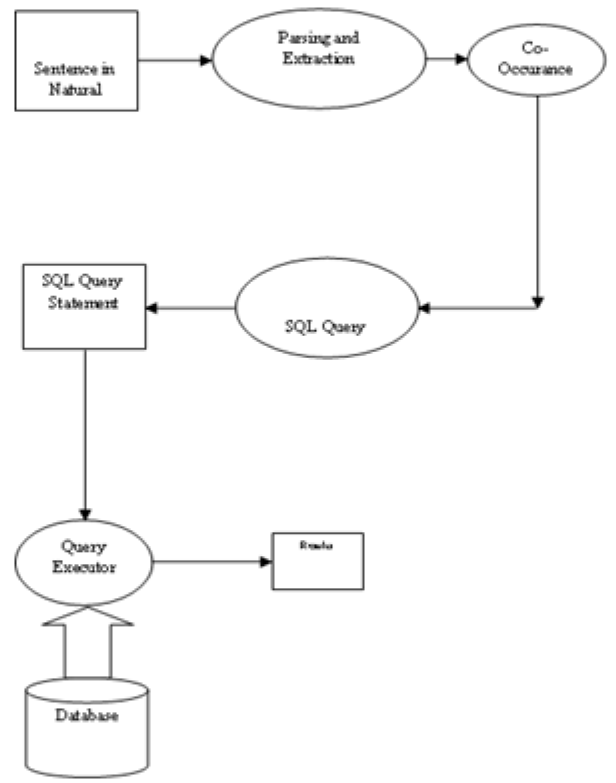
This is answerable for translating a query/command from natural language (i.e., English) to an intermediate form. The intermediate form is normally a standard database query language such as SQL. The intermediate SQL query is then transfer to Query Processor to produce the required information from the database as output.

Query Processor:

The Query Processor accepts the query in database query language such as SQL, and processes them, printing the required information from the database.

VII. METHODOLOGY

Co-occurrence matrix describes useful data for mapping and understanding the structures in the basic document sets. Various kinds of analysis have been carried out on this data and an important body of literature has been constructed, making to provide significant area of information. In a computer science research area based on database analytics processing, co-occurrence matrix has a new dimension because it can used extensively. In this environment, sometimes retrieving the information often from the document set. For this purpose we need to construct the co-occurrence matrix. Implementation of co-occurrence matrix algorithm for SQL NLP improves efficiency and reduction of complexity as well as time.



Word co-occurrence matrix algorithm is as follows:

1. Take input as natural language (English) query.
2. Fix moving window size (e.g. 5 words). All the words occurring within fixed window size are considered as co-occurring with each other.
3. Calculate and get modified word co-occurrence frequency matrix.
a. Take unique words from input
b. Eliminate stop-words from list obtained at step (English query from user is taken as input) i.e. consider only nouns, proper nouns, adjectives and numeric values in the same order as in the input. This method is called modified because in original method matrix is calculated using all the words from the input while in this method only some important keywords are used to generate frequency matrix.
c. Take words (Parsing and Keyword Extraction Module) and place as row and column header.
d. Consider first column header and first row header if both are same then take value as 0 in matrix at their intersection cell, if there is no word between column word and row word in input (i.e. if they are adjacent words) then take value as 5, if there is one word then take value as 4, if two words then take value as 3 and so on. This means use the distance between words to generate matrix. If there are more than 4 words between column and row header word in input then take value as 0 at intersection cell in matrix table. Do the same for every column and row and get final matrix.

Grammar

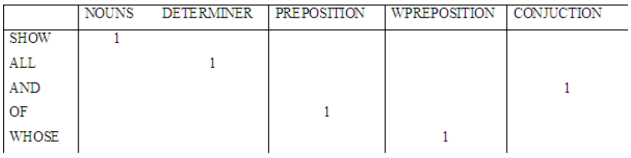
The dictionary is defined as below:
Verb {"show","list","display","get","find"}
Article {"a","an","the"}
Determiner {"full","entire","all"}
Preposition {"of","for","in","to","with"}
Wpreposition {"where","with","whose"}
Conjuction {"and","than"}
Participle {"being","currently"}
Operator {"more than","greater than","less than","equal to"}
Auxiliary verb {"is","are","where"}
Pphase {"who”,” whose"}

Parser

A natural language parser is a program that divide language into small components that can be analyzed and define grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Even thou these statistical parsers make done some mistakes but commonly work rather well. So, the parser will check whether the given sentence is syntactically valid or not. The parser uses grammar in order to check the syntax of the given query which is mentioned as follows

S -> VP PP
VP -> <verb> NP1
PP -><preposition> NP2
NP -> <determiner | article> | <article | determiner> | € & AQP
AQP -> <relation | attribute> CP | €
CP -> <conjunction> AQP
NP2 -> <determiner | article> | <article | determiner> | € & RDP
RDP -> <value | const.rel qualifier | € > & RLP
RLP -> <relation>

Ex: show all city and salary of employee whose name is thulasi



	NOUNS	DETERMINER	PREPOSITION	WPREPOSITION	CONJUCTION
SHOW	1				
ALL		1			
AND					1
OF			1		
WHOSE				1	

Ex: show me city and salary of employee whose name is thulasi

4. Create new vector for each word from table by concatenating its row vector and column vector from word co-occurrence matrix.

5. From the designed vector we find occurrence to design NLP query.

VIII. CONCLUSION

The proposed system accepts user query in natural language and translate it into SQL query and retrieve result from database. Syntactic parsing, keyword extraction, stop words removal, co-occurrence matrix generation, use of WordNet, stemming algorithm and semantic mapping techniques have been used for formation of the SQL query from natural language input. Developed system gives correct answers of simple queries, queries with logical conditions and aggregate functions. As presented system does not support all forms of SQL queries, further development is necessary.

IX. REFERENCES

[1]. Singh, H., & Seehan, D. Providing Inferential Capability to Natural Language Database Interface. Assistant Professor: Department of Computer Science Punjabi University Akali Phoola Singh Neighbourhood Campus, Dehla Seehan (Sangrur), Punjab, India `

[2]. Nihalani, N. (2010). An Intelligent Interface for relational databases. human-computer interaction, 6, 7.

[3]. Tang, L. R. (2008). Using machine learning approach for natural language interfaces for databases: Application of advanced techniques in inductive logic programming. Journal of Computer Science, Informatics and Electrical Engineering, 2(1).

[4]. Silberschatz, A., Korth, H. F., & Sudarshan, S. (1997). Database system concepts (Vol. 4). New York: McGraw-Hill.

[5]. Seidman, C., & Smith, P. (2003). MySQL: The complete reference. McGraw-Hill, Inc..

[6]. https://w3schools.com

[7]. https://developers.google.com/chart