IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

AN ALGORITHM TO TRANSFORM NATURAL LANGUAGES TO SQL QUERIES FOR RELATIONAL DATABASES

Ms. Shubhangi Pandurang Kamble Master of Engineering Student[Computer Engineering Department] TSSMs Bhivarabai Sawant College Of Engineering and Research , Narahe, Pune, Maharashtra, India

Abstract: In today's world information storing and retrieval plays an important role. Database systems play a key role in the new commercial system for information storage. For accessing the data from database, a person who is not having the knowledge of SQL may find themselves handicapped while dealing with the database. This presents a significant limitation in a developing country such as India, because even today, a very large majority of the population does not have the technical knowledge of how to deal with database systems. In such a case, Natural language processing (NLP) assumes a significant job. NLP is getting perhaps the most dynamic strategies utilized in Human-PC Interaction. It is a part of AI which is utilized for Information Retrieval, Machine interpretation, and Language Analysis. This paper, proposes a system which allows the user to query the database in a compatible mode language, through a convenient Graphical User Interface which results in the data required by the user.

Keywords: Natural language processing, Database, Artificial intelligence, Information retrieval.

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.

Purpose of Study

The internet has gradually become democratized and popularized, but databases still remain abstract for many people. However not every user familiar with structured Query Language (SQL) queries as they may not aware of structure of the database and they thus require to learn SQL. So non expert users need a system to interact with the relational database in their natural language such as English. For this database management system must have an ability to understand natural language (NL).

A model has been proposed to find solution to the problem. This helps people who are new and experienced in using SQL.

II REVIEW OF LITERATURE

This section reviews the different works concerned with the project. Some papers were studies and summarized as follows:

Paper[1] Anum Iftikhar, Erum Iftikhar, Muhammad Khalid Mehmood proposed a system for "Domain Specific Query Generation from Natural Language Text. It involves generation of SQL Queries using Stanford Parser. The paper revolves around ambiguity problems in NLP. Automated queries of NoSQL can be used as an application for the idea presented in the paper. It can also be used to design NL business etiquettes.

Paper[2] Prof. DebaratiGhosal, TejasWaghmare, VivekSatam, ChinmayHajirnis proposed a system for "SQL query formation using natural language processing". SQL query extraction from NLP is the idea discussed in the paper. This gives all users all possible intermediate queries. Appropriate intermediate query is selected by the user. The system then generates SQL query. This is done from the intermediate codes. System then executes the query and gives output to the user.

Paper[3]PrasunKantiGhosh, SaparjaDey, SubhabrataSengupta proposed a system for "Automatic SQL Query Formation from Natural Language Query". This system involves conversion of Natural Language Query to SQL language.It also presents the idea of Speech to Text Recognition. Python programming language has been used.

Paper[4]"Translating Controlled Natural Language Query into SQL Query using Pattern Matching Technique " is a system proposed by Rajender Kumar, MohitDua. Query to the database is given by the user and then the retrieval is done with the help of NLP. This system uses two Analysers such as morphological analyser and word group analyser. The main purpose of this analysers is to extract the keyword from input.Finding the type of keyword is the next step. It uses pattern matching technique to carry out this task.

Paper[5] Prof. Sonal Gore, NiketChoudhary worked on “Impact of IntelliSense on the accuracy of Natural Language Interface to Database “. An interface is used to ask query to the database in one‟s own language. Machine generates suggestions using intelligence. Based on previously typed words, sentences are formed. Suggestions frame a complete and correct query which can be used while connecting to the database for extracting data.

The related work includes the survey of various place people and their publications as a context of Research. The following implementation is done after studying various publications and sources.

III PROPOSED SYSTEM

The proposed model presents the idea of extracting SQL query using natural language processing for data manipulation and extraction.

The system will deploy a natural language understanding component that identifies speaker intents. The variables are needed for a specific nt example. Our solution uses the technique presented in the paper uses TensorFlow, Opencv, Keras, Numpy library, and also enhances it, which we are developing which will be schema specific.

SQL is known for its differentiating, high level language and close connection to the hidden data. These characteristics are utilized in our project.SQL is tool for manipulating data. To create a system which can generate a SQL query from natural language we need to make the system which can understand n atural language.

Following figure shows the architecture of system

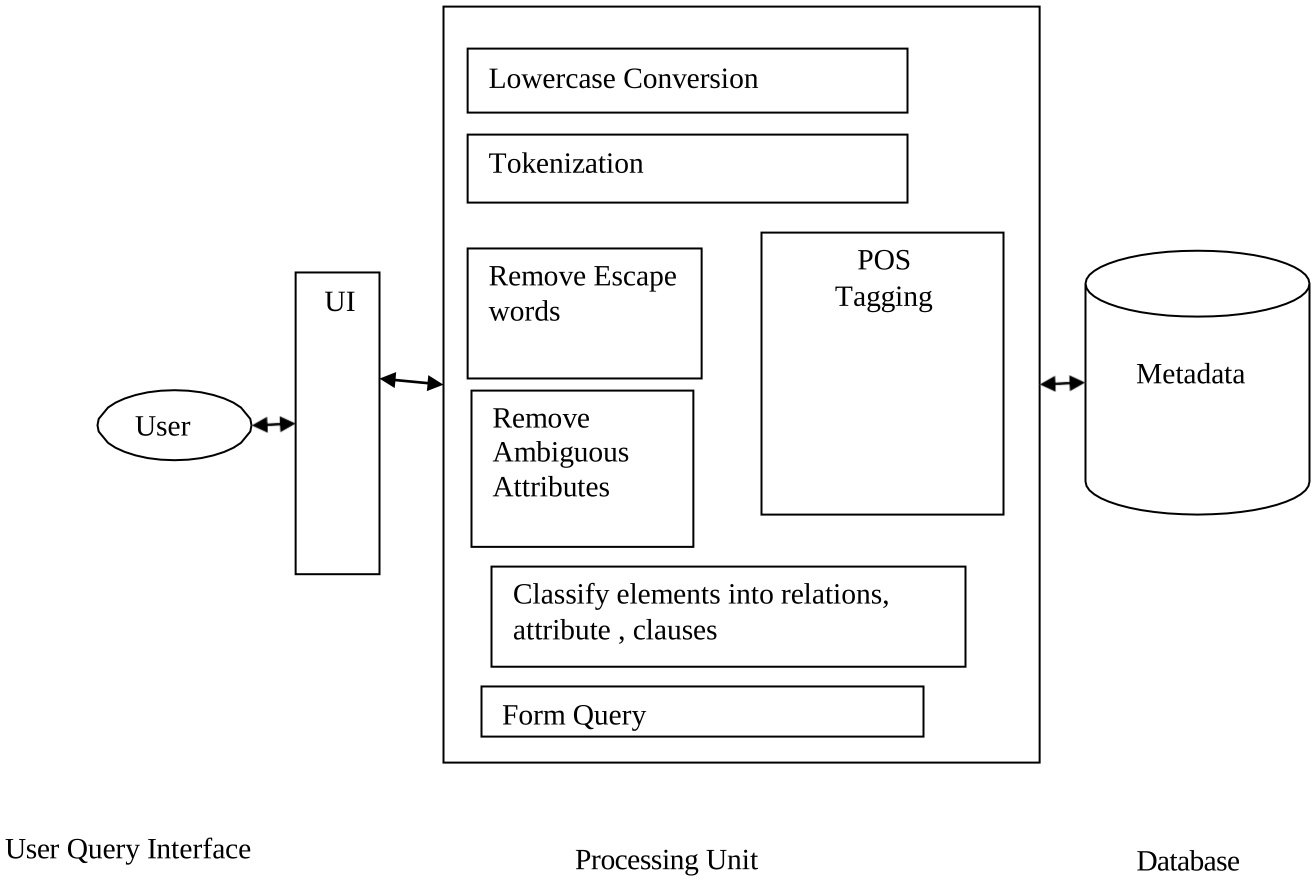


Fig. 5.1 System Architecture Diagram

IV RELATED WORK

A. NATURAL LANGUAGE PROCESSING FOR SPEECH SYNTHESIS

A system and the method interact with networked objects, via a computer using the utterance, speech processing and natural language processing. A data definition file relates networked objects and a speech processor. The speech processor searches a first grammar file for a matching phrase for the utterance, and searches a second grammar file for a matching state if the coordinating expression isn't found in the principal sentence structure record. The system also includes a natural language processor to search a database for a matching entry for the matching phrase.

The natural language processing is the computerized approach to analyzing text and being a very active area of research and development. This is based on the text to tasks on it (interrogation, addition, deletion), which translators of the language, natural to database query language have emerged

B. A BIT OF PROGRESS IN LANGUAGE MODELING

Speech conversion in which the text data is first input to the system. It uses high levels of modules for speech synthesis. It uses sentence segmentation which deals with punctuation mark with simple decision tree. Our perplexity diminishment is maybe the most noteworthy detailed contrasted with a baseline.

V IMPLEMENTATION

SYSTEM DESIGN

Existing system that gets a natural language sentence as an input, which is then passed through various phases of NLP to form the final SQL query.

EXTRACTING THE DATA

Speech in Hindi from the user is taken as the input converted into text using PyAudio and Speech Recognition Library. The audio can be provided through a microphone or similar device. Dictate feature of Microsoft Word can also be used for the speech recognition. Later, the text from the document can be retrieved using Python.

TRANSLATING SPEECH

The converted Hindi text is translated into English language using goslate library. Goslate library provides API for Python. It uses google translation website for conversion to English text.

LEXICAL ANALYSIS

A. Converting Text Data To Lowercase

Conversion of the text into lowercase is the primary step in lexical analysis. The lower() function converts all uppercase or capital letters present in the string into lowercase or small letters. This makes easy to understand the sentence and convert them into tokens.

The output of which is:

"which cities are located in greece."

B. Removing Punctuations

Punctuations like {, , :,"",‟,!,;} are eliminated from the sentence for tokenization. This step is very significant as punctuation does not add any extra info or value. Hence exclusion of such occurrences will have reduced the size of the data and increase computational proficiency. The replace function and regex is used for this.

Output:

which cities are located in greece

C. Tokenizing Texts

Splitting of sentences into minimal meaningful units is referred to as tokenization. Each result unit is known as tokens. It is easy to identify table name, attributes and their clauses using tokens. Libraries used in tokenization are NLTK, SpaCy, TextBlob.

['which', 'cities', 'are', 'located', 'in', 'greece']

D. Removing Stopwords

The mutual words that convey less or no meaning associated to other keywords are called stopwords. If such words are removed more important keywords can be focused on. Examples of stopwords {„I‟,‟are‟,‟the‟,‟in‟,‟of‟…}. The stopwords can be removed using NLTK library‟. These libraries are pre defined in Jupyter Lab.

Output:

['cities', 'located', 'greece']

SYNTAX ANALYSIS

A. Stemming

Stemming involves extraction of root words. For example, "study", "studies" and "studying" are stemmed into "study". This is also done by using the default libraries present in Jupyter Lab.

text=[‟I like to study‟, ‟She studies‟, „She is studying in college‟] Output: [‟I like to study‟, ‟She study‟, „She is study in college‟]

B. Lemmatizing

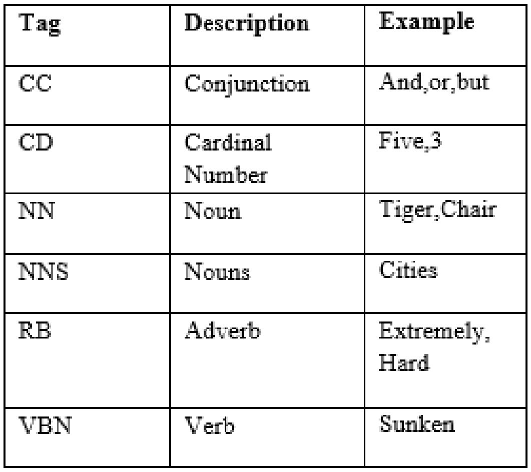
Lemmatizing is extraction of root words by checking the vocabulary. For instance: [„Got“, “secured“, „score“] is lemmatized into [„secured“].

SEMANTIC ANALYSIS

A. Noun-Pronoun-Verb Tagger

The extracted tokens are tagged as noun, pronoun, or verb. The various tags with their description are mentioned below

Table 1 : Parts of Speech Tags and Description



Tag	Description	Example
CC	Conjunction	And,or,but
CD	Cardinal Number	Five,3
NN	Noun	Tiger,Chair
NNS	Nouns	Cities
RB	Adverb	Extremely, Hard
VBN	Verb	Sunken

Output:
[('cities', 'NNS'), ('located', 'VBN'), ('greece', 'NN')]

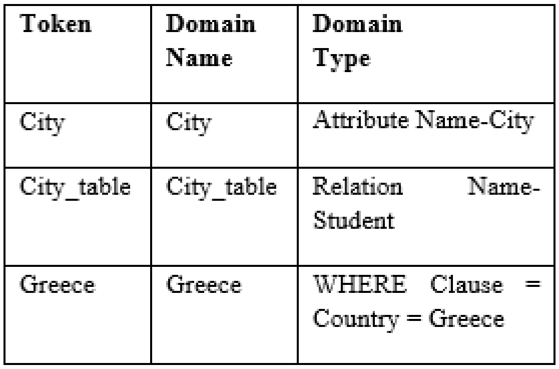
B. Ambiguity Remover

The most apt attribute is extracted and mapped with the relation after removal of ambiguous attributes existing in multiple lines. Ambiguity means one aspect can have more than one forms and meanings.
Output: No Ambiguity .

C. Relations-Attributes-Clauses Identifiers

Classification of relations, attributes, and clauses based on tagged elements is done after the elements into noun-verb-pronoun. This is done in order to easily recognize the relation name and its attributes to form SQL query.

Table 2: Relations-Attributes and Clauses



Token	Domain Name	Domain Type
City	City	Attribute Name-City
City_table	City_table	Relation Name-Student
Greece	Greece	WHERE Clause = Country = Greece

QUERY GENERATION

The final query is generated based on extracted elements after classifying the relations, attributes and clauses and further removing ambiguity. Then the query is analyzed and executed, and the outcomes are extracted from the database and are displayed to naive users.
Final Output:
SELECT City FROM city_table WHERE Country="greece"

VI ALGORITHM

Support Vector Machine Algorithm

Support Vector Machine (SVM) Algorithm (SVM) is a supervised machine learning algorithm that can be used for either classification or regression challenges. Support Vectors are simply the coordinates of individual observation.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

Mathematical Model:

Let S be the Whole system S= (1,PO) l-input
P-procedure O-output Input(1)
1=(Text As Dataset) Where,
Dataset-> different features of dataset as text Procedure (P),
P=(1, Using | System perform operations and calculate the prediction) Output(0)

VII OTHER SPECIFICATIONS

LIMITATIONS

- Poor Interface-SQL has a poor interface as it makes look everything very complex even when it’s not! Due to its difficult interfacing, users find it difficult to deal with the databases.
- Cost Inefficient -SQL Server Standard costs around 1,418/year. The high cost makes it difficult for some programmers to use it.
- Partial Control-SQL doesn’t grant the complete control over databases to its users. This is due to some hidden business rules.
- Security-Regardless of the SQL version, databases in SQL is constantly under threat as it holds huge amounts of sensitive data.

APPLICATIONS

- Data Definition Language (DDL)-SQL is used as a Data Definition Language (DDL). This allows users to autonomously make a database that can be structured, used and then destroyed when the purpose is fulfilled.
- Data Control Language (DCL)-Data Control Language (DCL) is a command of SQL. DCL commands are used to grant and take back authority from any user.
- Data Manipulation Language (DML)-SQL is also used as a Data Manipulation Language (DML). These

commands are used to modify the databases and are responsible for all form of modifications in the database. It is not auto-committed which means the changes made to the databases are not permanently saved.

❖ Transaction Control Language (TCL)-TCL commands can only be used with DML commands like INSERT, DELETE and UPDATE. These operations can’t beused while creating tables because they are automatically committed to the database.

VIII CONCLUSION

In this study, we present our work for the generation of SQL queries from natural language. To conclude, although it can be improved, this approach allows you to query any SQL database, thus meeting the portability objectives set, while keeping performance within the average of the applications already existing and covering a wide range of selection operations.

The aim is to evaluate correct sql translations for NLQ. The intelligent interface developed uses semantic matching technique which translates natural language query to SQL. It also uses set of production rules and data dictionary which consists of semantics sets for relations and attributes. A series of steps like lower case conversion, tokenization, speech tagging, database element extraction, SQL element extraction and ambiguity removal is used to convert Natural Language Query (NLQ) to SQL Query.

IX REFERENCES

[1] “A comparative survey of recent natural language interfaces for databases” by Katrin Affolter1 · Kurt Stockinger1 · Abraham Bernstein2.

[2] “A system to transform natural language queries into SQL queries by Arun Solanki1 • Ashutosh Kumar1”.

[3] ”Methodology of SQL queries generator to database set by natural language text” by Mariya Zhekova,George Totkov. “SQL Generation from Natural Language: A Sequence-to Sequence Model Powered by the Transformers Architecture and Association Rules” by 1Youssef Mellah, Abdelkader Rhouati, 1El Hassane Ettifouri,Toumi Bouchentouf and 2Mohammed Ghaouth Belkasmi.

[4] “Enhancement of Natural Language to SQL Query Conversion using Machine Learning Techniques” by Akshar Prasad1, Sourabh S Badhya2, Yashwanth YS3, Shetty Rohan4, Shobha G5, Deepamala N.

[5] ”Information Retrieval using Natural Language Interfaces by M. Sreenivasulu.

[6] “ Extracting Sql Query Using Natural Language Processing” by 1Nandhini S, B.Viruthika, 2Almas Saba, 3Suman Sangeeta Das
.

[7] www.edureka.co.in.

[8] www.google.com.

[9] www.python.org

[10] www.wikipedia.org .

[11] www.tutorialspoint.com.