AI Model to Generate SQL Queries from Natural Language Instructions through Voice

Aditya Sawant, Rohit Raina, Anuja Patil and Anand Pardeshi

Department of Information Technology, Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai, India

E-mail: adityaoz9990@gmail.com, rohitraina885@gmail.com, anujaspatil99@gmail.com, anand.pardeshi@fcrit.ac.in

Abstract. Data plays the most important role in the development of industries, small businesses. Even world leaders need the data to make analyses and make better policies for people. In almost every field where the work process is digitized need to store data and then retrieve it. According to statistics most of the data is stored in the relational database and for manipulations of the data, Structured Query Language(SQL) is commonly used. So for handling databases a person need to have specialized knowledge regarding the queries and had to remember the syntax of many complex queries. So to enhance data manipulation using SQL and to efficiently get the required query, the paper proposes a method for the generation of SQL query from natural language input, spoken(audio input) by the user. The model is constructed on NLP (Natural Language Processing) and Neural Networks (Deep Learning) technologies. Long Short Term Memory(LSTM) Model is used for predicting queries and is trained on the dataset with natural language as input and returns outline skeletal structure of the query as output. Then the output will be processed and the final query will be displayed to the user. The project also aims to benefit the people who are suffering from Repetitive Stress Injury (RSI), causing pain in the finger joints, which has been attributed to work requiring a long period of typing and also to those who are not familiar with SQL queries. As this system will readily provide the required query.

Keywords: AI, Neural Networks, LSTM, NLP, Speech to query, SQL.

1. Introduction

In the era of technology, many jobs have become much simpler and efficient. Technologies like AI, ML, DL are assisting humans in many domains such as medical, space exploration, Bio-technology, Automobiles and many more. Technology has removed the barrier of having specialized knowledge to do certain work like anyone can create web pages by using pre-build templates and modify them by using drag and drop methodology, Anyone can make 3D models using 3D printers. Retrieving the required data or information from relational databases is a tedious process. The person has to have specialized knowledge about Database Management systems, need to learn the syntax of the various complex queries. This makes it difficult for a non-technical person to retrieve data from databases. Advance technology can help to tackle this problem, where users don't have to remember the code snippets and have to do specialization in databases. Natural language processing can be used to interact with computers. And neural networks can be used to predict the semantic of user requirements and convey them to computers. This paper aims to combine this technology and create an environment where anybody can create a SQL query and work on relational databases. The user can interact with a computer specifying his data requirements from the databases in natural language by speaking it to an application that listens and converts the speech into text, and then preprocessing the text for understanding the user requirements and creating a structured query, which can be used to get the data from the database [1]. The neural network model is used as it gives the most accurate result compared to other technologies. The proposed model will take English language speech as input. There are various applications available in the market which is based on text to SQL query where the user has to type his requirements, but in this proposed model speech will be taken as input. So it will also help the disabled, paralyzed user. Many engineers are suffering from Repetitive Stress Injury(RSI) which has caused by continuous typing on keyboards. The muscles. nerves, tendons get severely damaged by repetitive motion of fingers. Hence this problem also will be solved by the proposed model.

2. Related work

2.1. A Model of a Generic Natural Language Interface for Querying Database [1]

In this paper a model for Natural language handling utilizing a data set (NLDBI) was proposed. This model depends on AI for questioning data sets subsequently empowering the point of interaction to further develop information in view of AI approach. Two methodologies for this was depicted in the paper, Linguist Component: In this approach three analysis morphological, syntactic and semantic was performed. Database Knowledge Component: The cycle comprises of two parts DBQ generation and DBQ execution. DBQ generation interprets the IXLQ which is made by the semantic analyzer into SQL. After the generation of DBQ, it will be executed by the Database Management System (DBMS), and afterward the responses will be shown in table.

2.2. Automatic SQL Query Formation from Natural Language Query [2]

In this paper proposed the Natural Language Processing system was carried out by means of the method known as Synchronous Model of language or "Levels of Language". The process had four stages called Morphology, lexical, Syntactic, Semantic. Each of these stages covers different tasks such as breaking down the sentences into tokens, after which interpret the meaning of individual words in which all tokenized sentences will get mapped along with the meaning of the corresponding words. After this it was found that the attributes present in the given query forms the same words that were generated in the previous stages. The Semantics part focuses on the study of the meaning of the words that are present in the input query and the relation between the signifiers present in it for example words, signs, phrases and their meaning. The project they made was for android so python for android was used for the speech recognition part.

2.3. Automatic SQL Query Formation from Natural Language Query [3]

This paper proposes a system that is capable of producing text or speech in English or natural language to SQL queries. The system will take the user's query in natural language format and will check whether the query is valid or not. If the query is valid then it will generate a token that will divide the query. Each token represents a word. The tokens are compared with clauses in the stored dictionary and then it tries to find the attributes in the query. Then it will find the tables in the database that will contain the attributes by comparing syntax and semantic hence building a query and executing it giving the user its required output from the database.

2.4. Extracting SQL Query Using Natural Language Processing [4]

This paper describes the model which presents the idea of extracting SQL query using natural language processing which can be used for data manipulation and data extraction. The implementation is done using python. It involves taking input, doing lexical analysis, syntax analysis, semantic analysis and finally query generation. The lexical analysis involves converting text to lowercase, removing punctuation, tokenizing texts and removing stopwords. Stemming and lemmatizing is done in syntax analysis. Semantic analysis helps to remove ambiguity, tag noun-pronoun-verb and identify relations-attributes-clauses.

2.5. Natural Language Processing with some abbreviation to SQL [5]

They developed a system to generate valid SQL queries after parsing natural language using open source tools and libraries. It also analyzed the abbreviations in NLP to get the required output from the database. Finally, it executed the query and fetched the required records. They implemented it using python. It had a GUI for the implementation of the model. The input given through speech or text gives the required output in the interface.

3. Proposed System

As the tool will be assisted to retrieve data from relational databases by providing SQL Query. The tool is made using various technologies combined together and runs in a unified way providing users with the accuracy, efficiency and easy to use environment. The Audience for the tool is anyone who has no domain or specialized knowledge of Database Management and wants to retrieve, manipulate the data from the database. For using the tool they have to speak their requirements in the natural English language like they converse their requirements to other peoples in daily life. The tool will listen to the vocals spoken by the user and convert them into text format. And the text then will have to go through various stages where it will be processed and analyzed and a SQL query will be returned to the user satisfying his requirements. The tool is a web-based application, where the user has to first fill in the information about the database on which the user wants to work [5]. The information is the table names, their attributes and general information about the database. The flowchart is as shown in Fig. 1. The following modules summarize the building of the tool:

- Preprocess the dataset using NLTK techniques like tokenization, lemmatization etc.
- Building the training set in the required format.
- Build an LSTM retrieval-based model.
- Training the model on the dataset.
- Build a Django based web application and integrate the trained model with it.

4. Implementation Approach

4.1. Creation of Original Dataset

We have created an original dataset for the training of the model. The format chosen for the dataset is JSON(javascript object notation) format. This format is efficient in data transferring, processing and analyzing. It is an attribute-value pair format. We have used 3 attribute-value pairs for a single SQL function. The attributes are tag, pattern, response. The Structure of the dataset for selecting a particular table is given as follows:

```
{"dataset":[
    { "tags" : "select_table",

    "Patterns" : ["give me the table
    table_name", "What is the query to
    print the table table_name",
    "show me the table table_name"],
    "Responses" : ["select * from table_name"]  }, ] }
```

In the tags attribute, we give the name of the SQL function. In the Patterns attribute, we have the possible ways the user could frame the requirement for a particular query to trigger and in the response tag, we have the skeletal structure of the query, which can be completed by processing the database information. The dataset must be include all the methods and functions used in SQL. In the dataset, the user has to provide the possible commands user can speak to get the required query. The number of commands provided will vary according to SQL function. So the user can provide more than 6 commands for a particular query to get maximum accuracy. The user's question will also be processed to bring it in the format shown in the pattern attribute.

4.2. Pre-Processing of Data

As the proposed system is a voice-based application, there are chances the user may provide redundant data with his requirement, which will affect the performance of the model. It becomes necessary to pre-process the data before feeding it to the model [4]. So there are various steps taken to convert the data into the required format. Steps are as shown in Fig. 1.

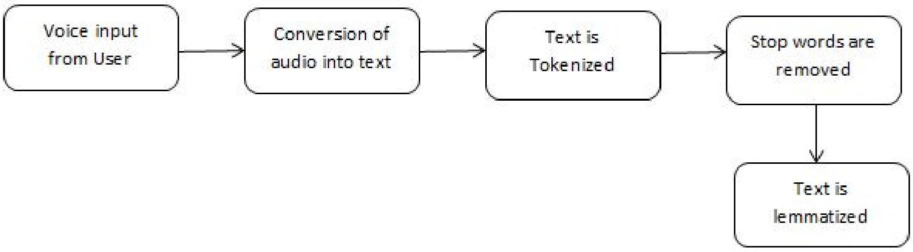

Figure 1. Steps involved in Preprocessing of data

4.2.1. Tokenization Tokenization means, breaking sentences into their constituent words. It can be done by, whenever the system will encounter a space it will consider it to have a token [2,3]. Eg: Before tokenization:["Show me the Student table"]

After tokenization:["Show"," me"," the"," Student"," table"]

4.2.2. Removing Stopwords There are various words that are frequently used but have no use while analyzing them, these words are called Stopwords. So they must be removed from the sentence to improve the performance of the model. It has been done by making a list of all stopwords. And check whether the word from the sentence lies in the stopword list, if it lies then it is removed. Eg: Before removing stopwords:["Show"," me"," the"," Student"," table"]

After removing stopwords:["Show"," Student"," table"]

Here "me" and "the" are removed.

4.2.3. Lemmatization Lemmatization deals with finding the root words. Various words can be written in different forms, considering the tense, quantities etc. So it can confuse the model as it will interpret the word as completely different.

Eg: Before lemmatization - [Two cars collided]

After lemmatization - [Two car collide]

4.3. Creation of Input and output set

The model will be trained on input and output data. The input data will be the patterns, the different ways the user can ask his requirement for triggering a particular function. And the output will be the tag or class to which that input pattern belongs.

4.4. Training of model

The input and output data will be trained on the LSTM(long short term memory) [6]. It is a very efficient and powerful sequence prediction neural network. LSTM is the most widely used algorithm in many for solving many industry problems as predicting the stock pattern by using sales as input, language translation, and predicting the next words in many messaging applications. In the proposed model LSTM model will be used for getting the intent of user requirements. By analyzing the sequence of words user has used in speech, it will return the tag which specifies the function or query the user wants. On receiving the tag, the model will get the skeletal structured query which is stored in the dataset. The skeletal query will be processed by using the information the user has provided about the dataset in the application. Then the model will return an executable query to the user as shown in Fig. 2.

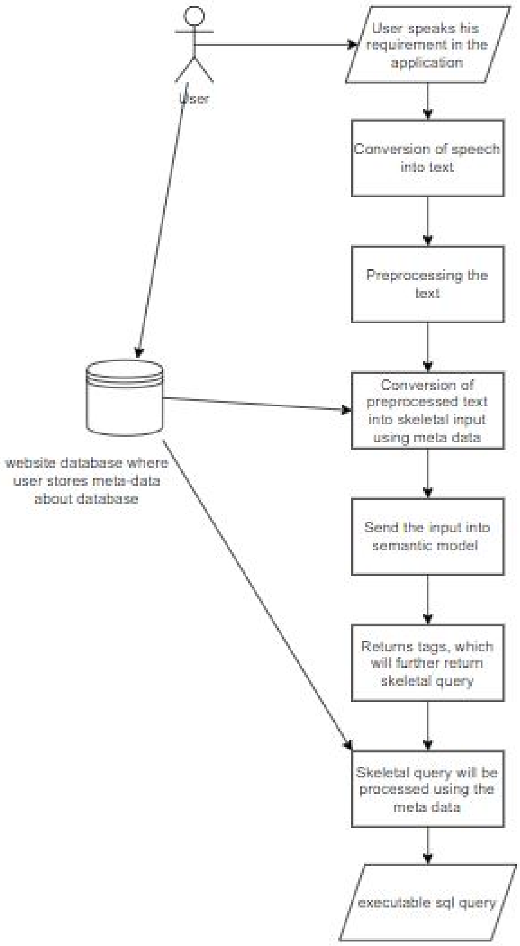
User
User speaks his requirement in the application
Conversion of speech into text
Preprocessing the text
Conversion of preprocessed text into skeletal input using meta data
website database where user stores meta-data about database
Send the input into semantic model
Returns tags, which will further return skeletal query
Skeletal query will be processed using the meta data
executable sql query

Figure 2. Flowchart of implementation

5. Results

The dataset has been trained on neural networks based LSTM model, which provide sufficient accuracy. The model has given accuracy of 93.47. To verify the accuracy of the model, it was run on several test cases and accordingly the accuracy was calculated. It was also observed that the model takes the time to return the query in order of milliseconds, this specifies the model make quick decisions. It is observed that the more the requirements are spelt in simpler way more accurate result will be returned. So the result also depends on the user, how efficiently and simply the user can specify its requirements. In a nutshell, the development of the system is an ideal for retrieving the data from a database. The commands tested on the model and the output returned is shown in Table 1.

Table 1. Commands tested on model.

Command given by user(audio format)	Output of Model
`Show the entire student table`	select * from student;
`Show the name column from student table`	select name from student;
`Give me grade column from student table`	select grade from student;
`Delete the student table`	drop table student;
`Alter the table student and drop email`	alter table student drop column email;

6. Future Scope

Currently in our project a web application is proposed which can convert natural language instructions for database queries via voice to corresponding SQL queries. It is currently capable of handling simple queries along with some complex queries. In future the support of other query languages as well as programming languages can be implemented using the same model proposed. The user will be able to switch between any programming language with the same instructions given. An extension can also be implemented using this AI for integrating into editors like VScode, Atom, Sublime etc. The user after installing the extension will be able code through voice with added functionalities like suggestions and auto-complete. The feature of converting code from one programming language to another can also be implemented.

7. Conclusion

Artificial Intelligence and Machine Learning have enhanced the life of most human beings, our paper had also proposed a method to enhance the coding experience. The project successfully fulfils the objective to help people suffering from RSI(Repetitive Stress Injury) and also to ones who are not from a coding background. The model provides an accurate and efficient query to users who provide input in natural language. The use of Natural Language helps users to easily retrieve data. This system will help many organizations such as education, medical, etc. For the maximum performance of the system, the database has to be updated frequently with specific words to the particular system.

8. References

[1] Bais, Hanane, Mustapha Machkour and Lahcen Koutti. “A Model of a Generic Natural Language Interface for Querying Database.” international journal of intelligent systems and applications. 8. 35-44. 10.5815/ijisa.2016.02.05.

[2] Ghosh, Prasun Saltlake, Kolkata Kolkata, Saparja Dey, Kolkata Sengupta, Subhabrata Assistant, Kolkata Saltlake,. (2014). “Automatic SQL Query Formation from Natural Language Query”, International Conference on Microelectronics, Circuits and Systems (MICRO-2014).

[3] Nagare, Indhe, Sabale, Thorat and Chaturvedi. ”Automatic SQL Query Formation from Natural Language Query” International Research Journal of Engineering and Technology (IRJET) Mar -2017

[4] Nandhini S, B.Viruthika, Almas Saba, Suman Sangeeta Das ”Extracting Sql Query Using Natural Language Processing” International Journal of Engineering and Advanced Technology (IJEAT) April 2019

[5] Chandrakala Kombade, Monika More, Shweta Patil, Anjalidevi Pujari ”Natural Language Processing with some Abbreviation to SQL” International Journal for Research in Applied Science Engineering Technology (IJRASET) Dec 2019

[6] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.