Transformation of Natural Language Text to SQL Query

Sanchita Nandi¹, Shravan. N², Swathi Kote³, Vrushank. V², Nandini. P⁵
Assistant Professor⁵
Department of Computer Science and Engineering
The National Institute of Engineering, Karnataka, India

Abstract:
Data in relational databases can be accessed and managed using a standard language named Structured Query Language. Any individual who needs to get the information from the database needs to know at least the fundamentals of SQL. Be that as it may, a plebeian with no basic knowledge on SQL would not know how to utilize SQL to manage the database. In this paper we have proposed a framework to conquer this issue by converting the Natural Language Query to Structured Language Query subsequently empowering any person without any insight on SQL to effortlessly access data from the relational database. In the proposed framework text or speech input is permitted. Speech input is first converted to text. SQL query is generated from the input in text format.

Keywords: Natural Language Query, Structured Query Language, database, text, speech.

I. INTRODUCTION

Natural Language alludes to the way we humans interact with each other. Everything that is expressed is data in unstructured format. A branch of Artificial Intelligence named Natural Language Processing deals with the interaction between the computers and human beings. Natural Language Query here, is a sentence in English language looking for data in response. This query can retrieve data from the database without having to learn the complexity of Structured Query Language. This makes it easier for the user to make use of the database system for his/her purpose. Speech recognition is used to convert the words or phrases in spoken language to machine-readable format by the machine or a program. Speech recognition software takes the input in Natural Language. The speech-to-text Application Program Interface converts this input to text format. The input is given through the Graphical User Interface that our framework provides. The GUI gives options to the user to provide the input through either speech or text. Once the input is entered the system will convert the sentence to an SQL query and retrieve the data from the database. Natural Language Processing is posed with a number of challenges such as the ambiguity in the words or phrases. One word could mean different in different contexts, which means one word could be mapped to more than one keyword. The biggest challenge of the system is the construction of SQL query.

II. RELATED WORK

Prof. Debarati Ghosal, Tejas Waghmare, Vivek Satam, Chinnmay Hajirnis suggested a framework for "SQL query formation using natural language processing". This framework gives an expert system which transforms simple natural language query into SQL query language by providing the user all possible intermediate queries so that the user can select suitable intermediate query and the system will output SQL query from intermediate one. Subsequently, the system will run the query and provide output to the user. [1]

Anum Iftikhar, Erum Iftikhar, Muhammad Khalid Mehmood suggested a framework named "Domain Specific Query Generation from Natural Language Text". The framework works by settling ambiguity issues in Natural Language Processing, utilizing Stanford parser to read through English messages and create SQL queries. The result of the proposed apparatus shall be utilized for computerized NoSQL databases querying and plans from rules of NL business. [2]

Prasun Kanti Ghosh, Saparja Dey, Subhabrata Sengupta proposed a framework for "Automatic SQL Query Formation from Natural Language Query". This framework will change natural language query to SQL language and Speech to Text Recognition for Android utilizing Python programming language. [3]

Prof. Sonal Gore, Niket Choudhary proposed " Impact of IntelliSense on the accuracy of Natural Language Interface to Database ". Here any layman can query the database in natural language. This framework proposes to handle errors in the system too. Utilizing AI, they improved IntelliSense such that the machine could produce recommendations dependent on formerly composed terms. These recommendations are used to outline an absolute and precise query. [4]

"Translating Controlled Natural Language Query into SQL Query using Pattern Matching Technique " is a framework suggested by Rajender Kumar, Mohit Dua. The framework works on querying from databases and makes use of natural language to fetch information. This framework is put forth as the client can input questions in Hindi to acquire information amongst the data in the database.

The framework utilizes two analyzers, for example, morphological analyzer and word bunch analyzer. The principal motivation behind these analyzers is to obtain the phrase from the Hindi sentence. It utilizes pattern matching methods. In order to decrease ambivalence, it utilizes the supervised Hindi interface and proposed query feature. [5]
III. PROPOSED METHODOLOGY

The proposed framework is intended to limit the communication hole between a human and a PC. It is created to encourage improved connection between the two. As it is realized databases just deal with standard queries written in SQL and it is less feasible for a typical individual to know SQL. Likewise, the users probably won't know about the database schema which includes table names, fields, types, etc. Hence, here a framework is planned which contains an insightful, intelligent layer that takes common language as input, and changes these sentences into standard SQL queries. This is how data or information can be retrieved from the databases. To start with, the client needs to login to the framework. Queries such as "put...","I want...","give...", "find...", "search..." can be asked. Query can be entered through text or speech format. The user queries are handled and processed stepwise. The system architecture is shown in Fig 1:

![System Architecture](image)

There are four fundamental advances or levels engaged in this conversion process. They are:

A. Tokenization
Tokenization will be performed in the framework on the input query by splitting the query into single words. These words act like tokens. A separate list will be formed for these tokens where they are stored in and sent to the Lexical Analyzer.

B. Lexical Analysis
Dictionary is used for the mapping of the tokenized list. These words will get supplanted by the words of the database present in the dictionary and sent to syntactic analysis.

C. Syntactic Analysis
In this progression, a dictionary of names and attributes of tables are kept up. Dictionary is used for mapping tokenized words with attributes. It is later sent to Semantic Analysis for additional preparing and processing.

D. Semantic Analysis
The framework will discover words which speak to conditions or any symbol, that will then get mapped with the dictionary.

The algorithm stages are as follows:

Stage1: The client enters or records the query in Natural Language text which is later put away in a string.

For instance, check the below query:

Get all the data of all students from the table students whose tenth score are lesser than 86, twelfth score are lesser than 81 and overall score are lesser than 66 in descending order.

Stage2: Individual words (tokens) are split from the sentence keeping in mind the spaces in it. These tokens are then collected and are put away in a different list.

The tokens are as follows for the query

Get, all, the, data, of, all, students, from, the, table, students, whose, tenth, score, are, lesser, than, 86, twelfth, score, are, lesser, than, 81, and, overall, score, are, lesser, than, 66, in, descending, order

Stage3: Only significant tokens are considered for additional handling by comparing the tokens with the overlook list.

Overlook List: the, an, are, is, to, marks, data, of, in, than and so on.

Stage4: Mapping is done for the selected tokens using the dictionary. These tokens get supplanted by their equivalent words (synonyms) and word type (attribute_name/table_name/value/keyword) is affixed to it.

Get, all, students, table, students, whose, tenth, lesser, 86, twelfth, lesser, 81, and, overall, lesser, 66, descending, order

Stage5: After this suitable linguistic structure of the query is concluded by watchwords, for example, select, insert, update and so forth.

For Example: For select question, as per sentence structure

I) Select watchword is appended by the tokens having word type as the attribute_name

ii) From watchword is appended by the token having the word type as table_name

SELECT<parameter(s)>,*<keyword>, students<table_name>, WHERE<condition>, sscc<attribute_name>, lesser, 86<value>, AND, hscc<attribute_name>, lesser, 81<value>, AND, total<attribute_name>, lesser, 66<value>, ORDER BY<condition>, DESC

Stage5: After this suitable linguistic structure of the query is concluded by watchwords, for example, select, insert, update and so forth.
Stage 6: Identify the conditions or qualities indicated by the client in his/her question, assuming any. Those conditions are mapped with the word reference i.e., the dictionary. (For instance, Less than is supplanted with " < " image)

I) The WHERE keyword/watchword is followed by the clause/condition specified in the query.

ii) The legitimate lot attribute name taken for ORDER BY query.

Stage 7: Execution of the final query created is done on the DBMS (Database Management System) in order to retrieve the conclusive outcome.

**Last Query for the model will be:**

```
SELECT * FROM students WHERE ssc < 86 AND hsc < 81 AND total < 66 ORDER BY fname DESC
```

Stage 7: Execution of the final query created is done on the DBMS (Database Management System) in order to retrieve the conclusive outcome.

**Last Query for the model will be:**

```
SELECT<parameter(s)>*FROM students<table_name> WHERE <condition> ssc<attribute_name> < 86<value> AND hsc<attribute_name> < 81<value> AND total<attribute_name> < 66<value> ORDER BY<condition>,fname<attribute_name>, DESC t attribute name taken for ORDER BY query.
```

IV. CONCLUSION

The proposed system gives a solution for converting speech into SQL so that people who are unfamiliar with the query languages like SQL can query a MySQL database with ease. The proposed solution uses the advancement in technologies like machine learning and audio processing to facilitate placement officers and other professionals who need to access the database on a daily basis to obtain the desired data from the database quickly without needing to query from the classical excel file. In the proposed system, only SELECT queries are implemented as they are the most complex types of queries in SQL. Hence, in future, including INSERT, DELETE and other types of queries like JOIN will be useful. Other future improvements would be to improve the accuracy of detection of tables of the database instead of the exact word matching. Since the main goal of the project was to parse the queries and form appropriate SQL queries, hence the front-end was not given importance. A better UI with good UX can help the users to query with great ease. Another major improvement will be to make it adaptable with non-relational and semi-relational databases. The project also opens a wider domain/fields of application of Natural Language Processing due to the availability of a large number of NLP libraries and APIs and motivates researchers to solve new problems using this approach.

V. REFERENCES


[5]. Rajender Kumar, Mohit Dua "Translating Controlled Natural Language Query into SQL Query using Pattern Matching Technique", IEEE2014.


[7]. https://pdfs.semanticscholar.org/79ad/75d6d46e26743182e561fb99b1eaf85a9563.pdf

[8]. https://www.geeksforgeeks.org/part-speech-tagging-stop-words-using-nltk-python/

[9]. https://www.researchgate.net/post/How_to_approach_a_system_to_convert_english_text_to_SQL_query