

# Providing Natural Language Interface To Database Using Artificial Intelligence

Anil Kumar H M, Ashwin Ram Kumar, Harshitha P, Mahadevaswamy, Sachin D N

**Abstract:** In today's world, the maximum number of operations are executed using a system involving data storage in the database. Non-programmers also try to access the information stored in the database. So it is a challenging task to make the system support search interface to access the information without even knowing the syntax of database language. This paper represents a method in which the user can give input in natural language (currently English) and retrieve the desired information in the same language by generating a query on the database in SQL language. If any interpretation error occurs, we provide a natural query recommendation framework (in the future) to make it easier for the users.

**Index Terms:** Database, Natural Language, Natural Language Processing, Tokenization, Stop word filtering, Structured Query Language (SQL), Artificial Intelligence.

## 1. INTRODUCTION

THIS era has become an era of data, information, and computers, etc. With so much data being generated and being used, Databases are being created to store, retrieve and manage such humongous amounts of data. Since the need for storing data in almost every company or institute is increasing, how much ever big or small the data may be, it is becoming an arduous task. The database has helped them to relieve from managing the work but Database tasks require the person to know about SQL Query. With so much need and dependence being generated on SQL these days due to the need of managing information online, it has become a tedious task and so is the pressure on SQL programmers. Hence we have planned to develop a system which makes use of Natural Language Processing to help the system understand the sentence or question given by the user in Natural Language. This means we no longer have to remember each syntax and each rule or memorize them to build a program but we can simply feed the question into our system and we will get the corresponding SQL Query which can be applied to the Database. We have used the WikiSQL dataset.

- Ashwin Ram Kumar has currently finished Bachelors program in Electronics and Communication engineering in Vidyavardhaka college of engineering
- Anil Kumar HM has currently finished Bachelors program in Electronics and Communication engineering in Vidyavardhaka college of engineering
- Harshitha P has currently finished Bachelors program in Electronics and Communication engineering in Vidyavardhaka college of engineering
- Mahadeswamy is currently Assistant Professor in Electronics and Communication engineering department in Vidyavardhaka college of engineering
- Sachin D N is currently Assistant Professor in Computer Science and Engineering department in Vidyavardhaka college of engineering

## 2 LITERATURE SURVEY

### 2.1 Computation of User-based Query using Natural Language Processing

This paper depicts that the information is been extracted that works on user-based natural language queries mainly on the health domain. By considering that, retrieving information from such kind of Q/A systems is one among the vital sources that help physicians to decide on patient treatment, by retrieving a pretty much amount of information in response to a particular user query, they have escalated the quality of patient care. This paper consists of four folds: it identifies the tags of grammar and thus checks the syntactic structure of the user query, covers different syntactic structure for English query by writing rules, clustering that helps in computational representations being mapped to the query, information extraction is achieved by querying the database in order to provide results that are semantic and efficient on health domain. [1]

### 2.2 A proposed Natural Language Query Processing System

In this paper, they have presented a system that provides a provision to the users for search interface particularly for online applications, numerous different databases, and search engines. This system provides a result for different type of queries that the user fires to the system which are accurate and efficient. Even if there is any spelling mistake in the user input, the spelling gets autocorrected. In this, two techniques are used for spelling correction, first is using a dictionary and second, without using a dictionary. They use word pair mining technique for correcting spelling mistakes in the user input. The system proposed in this paper provides a user-computer interface in the form of database query language and also there is a provision of auto-correction of misspelled words. [2]

### 2.3 A survey: Natural Language Interface to Databases

This paper gives an introduction to the Natural Language Interface to Databases (NLIDB) wherein it uses Natural Language instead of SQL which leads to the approach of building Natural Language Interface to Relational Database. Numerous benefits will be provided by the NLIDB system. In this system, input requests can be in some natural language. It is mainly proposed so that any kind of users, largely layman or inexperienced ones can easily access information. This results

in information retrieval from the database using natural language. Different approaches gone through for NLIDB systems development are symbolic approach (rule-based approach), empirical approach (corpus-based approach), and connectionist approach (using the neural network). [3]

## 2.4 A simplified model of Natural Language Interface for Querying Database

Here, the introduction of a method for developing a "Natural Language Interface with Databases (NLIDB)" system is explained. The system prepared an "expert system" which is implemented in a prologue that identifies synonymous words in any language. Initially, the input queries are parsed, following which the SQL query will be obtained that is transformed from the natural language expressions. They have proposed a system that involves the conversion of natural language question to MySQL query and this gets executed in the android application. For this purpose, there are two applications in which one runs on the server and the other on user mobile. User application uses Google API to translate the natural language to English and that will be sent to the server application that converts it into MySQL query. Its response is given to the user which on execution in the user app gives the desired result. [4]

## 2.5 Natural Language to SQL Conversion System

In this paper, they have proposed a system that provides a facility for users to update database i.e., insertion or deletion of values from the table. This system accepts an English statement given by the user, then it tokenizes the statement followed by removing unwanted words. Proceeding to this, synonyms are replaced with actual names by identifying synonyms of column names and table names. Based on criteria words, tokens are placed in 4 parts by the system. Then it generates many intermediate queries based on the semantics of the English statement entered by the user. The user must select one of the many intermediate queries which have best relevance to the user's intended query. Then the system will generate an appropriate query. By executing the query generated, the desired output is obtained. [5]

## 3 PROBLEM DEFINITION

SQL has become so much important in our lives in this era. Enormous amount of data can be stored in databases. So, instead of us doing all the work of managing databases, isn't it better if it does the work for us? So, this project aims at developing a system that accepts user entered English query and converts that into SQL. This will help the user to easily get the required data just by querying in the Natural Language. All said but creating such a system still poses difficulties. The main problem drops in to identifying the Database name, the SQL keywords within the query and also to understand the meaning of the query given to the system. Understanding the meaning as a human being is pretty easy but for a computer to understand the meaning of a given sentence is a tough job as remembering the syntax rules for creating a query is for human beings.

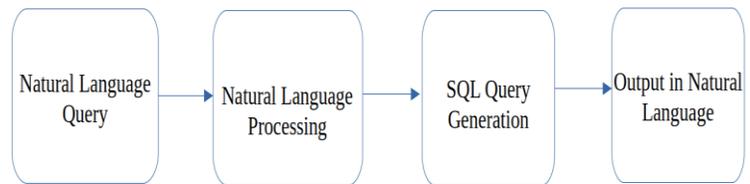


Fig. 1. Broad overview of this system

## 4 SYSTEM DESIGN / METHODOLOGY

This system or methodology is completely based on Natural Language Processing. When a Natural Language Query is given to the system, the system uses the query and other information from the database such as database name and columns and makes a SQL Query from all this information given to it.

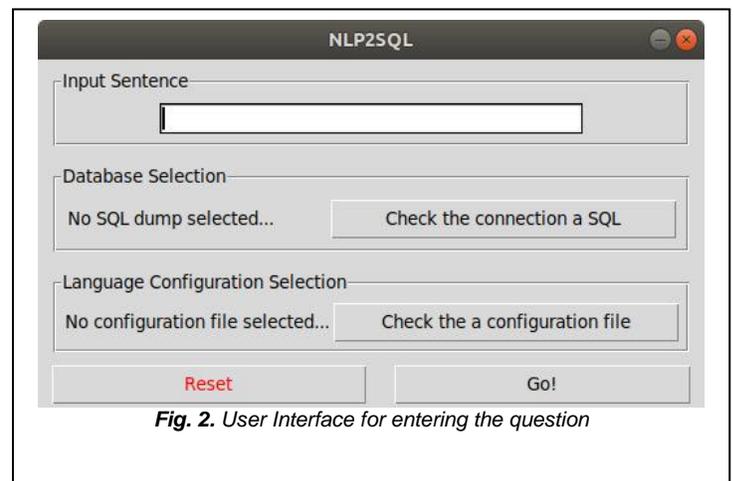


Fig. 2. User Interface for entering the question

First, the given Natural Language Query is taken from the textbox. The other parameters such as database path are taken and are loaded ready into the variables so that it would ease rendering of the system. Also, stop word lists are loaded. The input sentence is tokenized, meaning all words of the sentence are converted into a list format. The unwanted characters such as comma, quotes, etc. are removed from the input sentence. Along with removing unwanted characters, the stop words are filtered from the input sentence given by the user which leaves us with only important and useful words to help us construct the query. A list of keywords is also loaded which would signify/intend to mean any syntactic function of SQL in Natural Language, so it is easier for us to detect such keywords in the given input sentence.

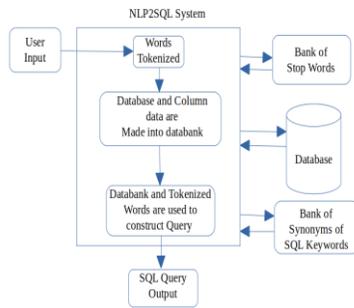


Fig. 3. Detailed blocked diagram of the system

Once the filtered input sentence is received, it is compared with the keywords list that is loaded and such keywords are used to find out the kind of operation that needs to be performed. Then keywords such as table name and column names are also simultaneously found out. Since all these keywords are jotted and where to use them is assigned, a JSON file is created using this, which contains all the conditions. This JSON file created will be read and the SQL Query is generated. This SQL Query generated is now displayed on the GUI for the user to see or execute. For Example, let us consider a question relating to a University marks Database named Analog. Let the Columns containing the marks be name IA1, IA2, IA3. For these set of names, if we would ask a question saying “How many people got IA1 marks more than 10 in Analog?”.

Table 1. Sample of Synonym bank

EQUAL: is	equal to	equals	are
DISTINCT: distinct	different	distinctive	unique
DESC: descending	decreasing	opposite	inverse
GREATER: greater	Greater than	more than	over

Then, the system would break down the query into a list of strings. Simultaneously, it would take column names and the name of the Database also. The string is tokenized, stemmed, compared with the list of SQL commands, names of columns and database given to form the query given below:

“SELECT IA1 FROM Analog WHERE IA1 > 10”

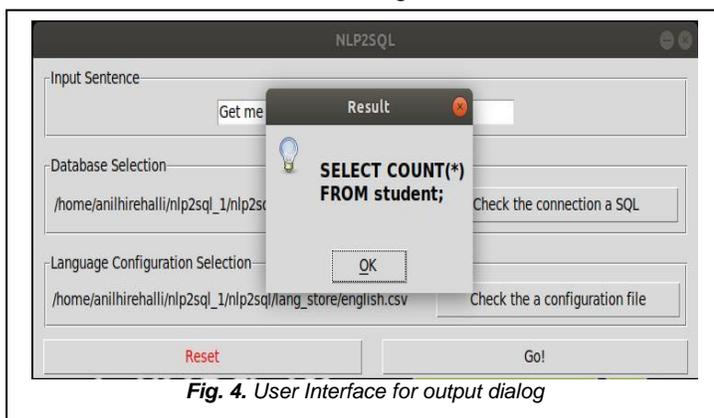


Fig. 4. User Interface for output dialog

sentence becomes,

["Get", "number", "student"]

Now, the important words left out are replaced with actual keywords from the synonym bank as shown in Table 1. So now the above list becomes:-

["SELECT", "COUNT (\*)", "student"]

This is then compared with the list of table names and column names to find out which table or column is to be used for the given operation. In this case, the student is identified as the table name. This information is more than enough to construct the Query i.e.,

SELECT COUNT (\*) FROM student

This is how we generate the SQL Query in our system.

## 6 RESULT

Well, it is very easy for us to recognize objects, shape, color, size, feel and other things as human beings but searching through vast amounts of data, remembering lots of syntaxes, rules are yet arduous tasks for us and that too when such a programming language is on high demand which higher pressure on the programmers. Hence, by automating this process, we can make things easier as well as faster than normal human beings.

## 7 FUTURE WORKS

We have planned to create a system which could suggest changes in the query so that the user can achieve his desired output. Also, we have planned to provide a voice interface to help illiterate as well as specially-abled people. To automate our system, we have planned an API version that could be used inside another software using similar techniques. Also, there are a few performance and accuracy improvements/changes that are needed to be addressed within the system.

## REFERENCES

- [1] Khurana, Priyanka, Mittar Vishav and Ruchika Yadav. "Computation of User-based Query using Natural Language processing." International Journal of Computer Applications 55.11 (2012).
- [2] Satav, Akshay G., et al. "A proposed natural language query processing system." International Journal of Science and Applied Information Technology 3.2 (2014).
- [3] Patel, Jaina, and Jay Dave. "A Survey: Natural Language Interface to Databases." International Journal of Advanced Engineering and Research Development (IJAERD) (2015).
- [4] Ashwath M, Balapradeep K N, Savitha C K. "A simplified model of natural language interface for querying database." International Research Journal of Engineering and Technology (IRJET).
- [5] Anil M Bhardgale, Sanhita R Gavas, Meghana M Patil, Pinki, R Goyal (2013). "Natural Language to SQL conversion system" IJCSEITR, 3(2), 161-166.
- [6] Garima Singh, Arun Solanki, "An algorithm to transform natural language into SQL queries for relational Databases," IAEES September 2016.
- [7] Prashant Gupta, Aman Goswamy, Sahil Koul and Kashinath Sartape, "IQS-Intelligent Querying System using Natural Language Processing," in International Conference on Electronics, Communication and Aerospace Technology, 2017.
- [8] Ikshu Bhalla and Archit Gupta. "Generating SQL from Natural Language." Stanford Publications, 2017, web.stanford.edu/class/cs224n/reports/6907018.pdf

sentence to a list. Then from the stop word list stored, the stop words along with waste characters are removed so the above

- [9] NAVID YAGHMAZADEH, et al. "SQLizer: Query Synthesis from Natural Language." University of Texas, Oct. 2017, [www.cs.utexas.edu/~ypwang/pubs/oopsla17.pdf](http://www.cs.utexas.edu/~ypwang/pubs/oopsla17.pdf)
- [10] K Javubar Sathick, and A Jaya. "Natural Language to SQL Generation for Semantic Knowledge Extraction in Social Web Sources." Indian Journal of Science and Technology, Jan. 2015.