International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 6, June 2020 ||

Conversion of Natural Language Text to SQL Queries

Abhishek P [1], Dr.S.S.Nagamuthu.Krishnan[2]

P.G. Student, Department of Master of Computer Applications, Rashtreeya Vidyalaya College of Engineering®, Mysore Rd, Bengaluru, India [1]

Assistant Professor, Department of Master of Computer Applications, Rashtreeya Vidyalaya College of Engineering® College, Mysore Rd, Bengaluru, India [2]

ABSTRACT: This paper explains an approach how to convert Natural Language to SQL Query. SQL (Structured Query Language) is a well-known tool for managing the data in the relational database. Usually to interact with the data in the database, one must either write the SQL query or SQL scripts which in turn executes on the database to work with data. But this task requires a person with the domain knowledge, a person who know the SQL to perform the task. It is impossible for a common man to perform this task or to write any queries. This puts a linguistic barrier between system and the person. To overcome this problem, to remove that linguistic barrier the proposed system helps a lot. The system makes use of some of the NLP(Natural Language Processing) techniques to process the data, understand the data and it gets the meaning of the sentence and creates the equivalent and accurate SQL query which is valid and can be used to interact with the database.

In this system, user inputs his query in natural language that is plain English text, the system first analyses the sentence, understand it and based on the context the SQL query is formed

KEYWORDS: Natural Language Query, Natural Language Processing, Speech-to-Text, SQL, Syntactic, Semantic, Data Dictionary.

I. INTRODUCTION

The intended purpose of this natural text to SQL converter is to provide ease of access to the data for everyone not only person who has good database knowledge. Fetching the required data from the database usually involves writing the SQL queries and the scripts which are executed to get the results back, which requires the person with the domain knowledge. Database querying has always been a specific task where common people cannot use it. So, this proposed project helps us in breaking this linguistic barrier between human and system interaction by converting the natural text into SQL queries. The system also makes use of several Natural Language Processing techniques which plays a major role in understanding what the sentence actually means. The natural language processing involves several sub tasks like tokenization, syntax parsing, Semantic analysis, sentence boundary detection, lemmatization and many more which are used in pre-process the data and clean the data to get the context and meaning of the sentence).

This system is designed for people who work on the database but don't have any knowledge of SQL. The System proposes the architecture for processing the natural English text to get SQL query using input as the text. The system as the ability to get the sentence and phrases, understand it's meaning to interpret it properly. Using natural language processing will give the solution to the problems arising in the analysis or generation of natural text, such as tokenization, semantic analysis and usage of dictionaries for mapping keywords with the appropriate SQL keywords and table name, column names present in the database.

The system presents the user with a GUI and error handling mechanisms which notify the user in case something goes wrong. The GUI presents the user with the field for getting the input, after getting the input from the user the system will convert the input to equivalent and accurate SQL query which then hits the database to fetch the results for the user. The user will also be able to see the generated query.

There are many challenges when it comes to converting the natural text to the appropriate and equivalent SQL query like ambiguity which means that one-word present in the sentence can have the multiple meanings, this results in one word getting mapped to different words. Another challenge is the formation of complex SQL query and next challenge

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 6, June 2020 ||

is about Discourse knowledge in which preceding sentence affects how the next sentence is getting interpreted for example if the user enters show me and fetch me at once, it maps to SELECT which rises the ambiguity of multiple select statements. The system takes care of this by eliminating multiple occurrences of same words.

II. PROPOSED SYSTEM

First, the user will be presented with the UI screen which has the input filed where user can ask their queries. The user can use phrases such as "get me...", "fetch me...", "find...","search…". The query given by the user is processed step by step. There are four main steps or levels involved in converting the natural text given by the user to the equivalent SQL query. These are as follows

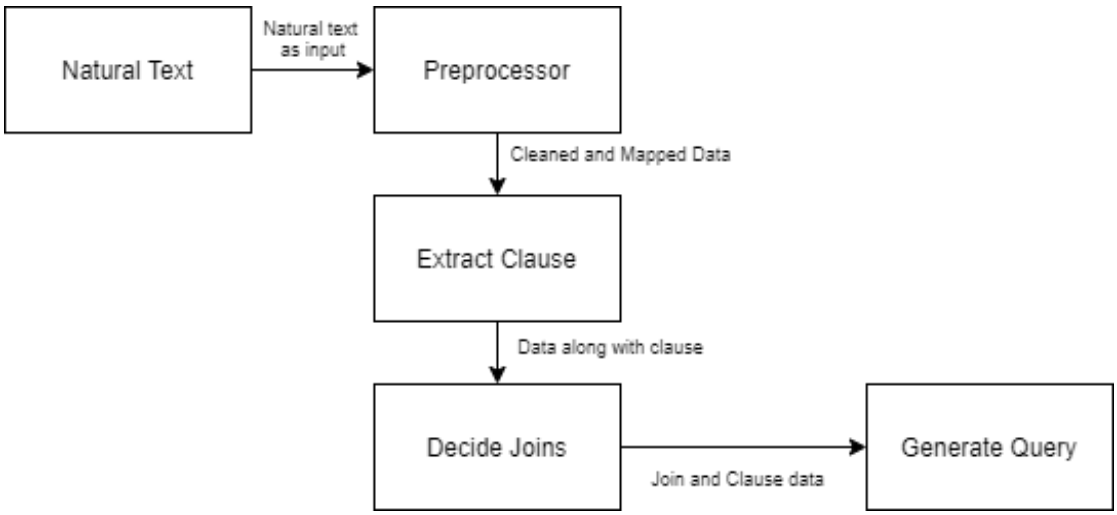

Fig 1. System Architecture

- Remove Punctuations

  System will remove the punctuations present in the sentence if any, punctuations serves no purpose in forming and is unnecessary and hence they are removed from the sentence and passed to the pre-processing task.

- Pre-processing

  The pre-processing class performs several sub tasks which are used to pre-process the sentence to get only what is required for forming the SQL query. These basically involve

- Tokenization

  System will perform tokenization on the entered query by separating it into single words. Each word represents a token. Then these words will be stored in a separate list and passed to pre-processing.

- Extract Clause Words

  System will find SQL clause words such as “where”,” from” which are required to be included in the SQL query for conditional queries. Ex. “less than equal to” to “<=” and so.

- Map column and table names

  System will find words which represent table name or column name and that word will get mapped with the dictionary which contains table names and database names.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 6, June 2020 ||

- Generate Query

After getting the necessary keywords and mapped words the system will now goes for generating the query, here based on the number of tables the joins are performed and also if the clause words are extracted, they will be included in the generated query for conditional queries.

III. ALGORITHM

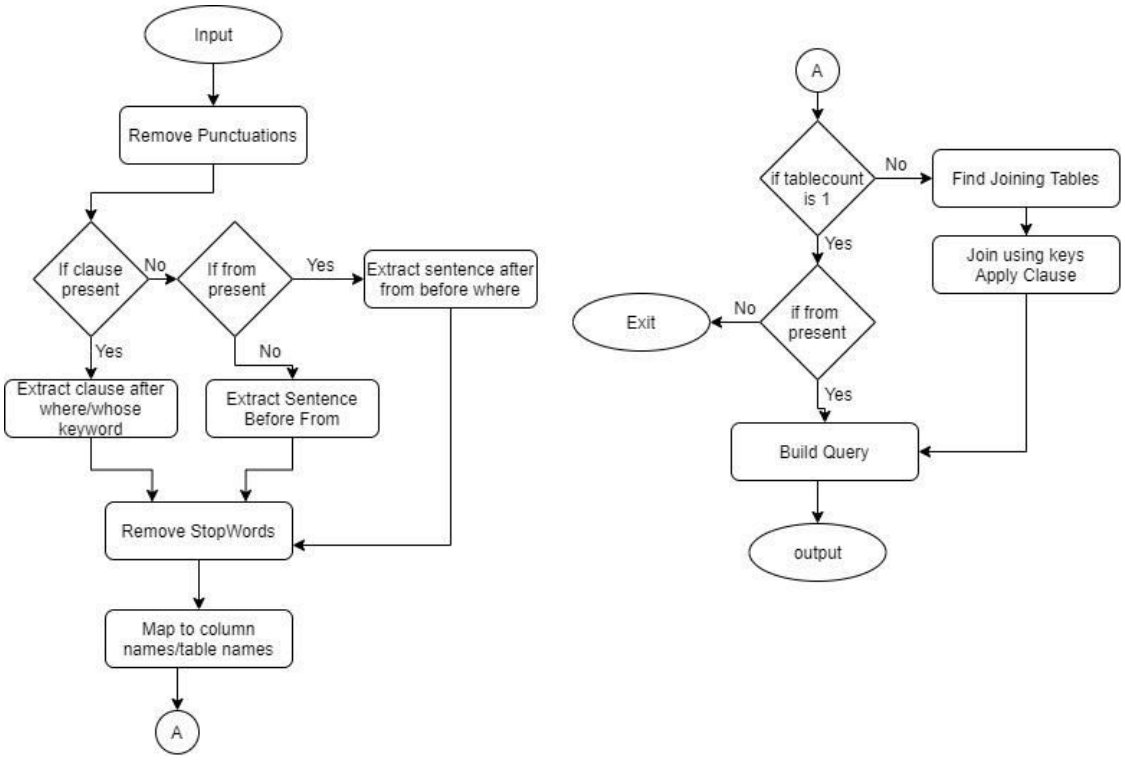

Figure 2. Approach

Step 1: Query entered by the user (in text) in Natural Language is stored in a string.

For example, consider the query

Fetch me all the patient's id whose organization id is 102

Step 2: Punctuations are removed from the sentence if any and sent to preprocessing step.

Tokenization:

[Fetch, me, all, the, patients, id, whose, organization, id, is, 102]

Mapping Keywords:

Keywords are mapped with equivalent columns and SQL keywords

[select, me, all, patient_id, where, organization_id, =, 102]

Step 3: Tokens are compared with the stop words; only important tokens are considered for further processing.

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 6, June 2020 ||

Stop words: the, a, are, is, to, of, in, than etc.

[select, patient_id, where, organization_id, =, 102]

Step 4: Selected tokens are mapped with dictionary of tables and columns to get tables need to be there in the query.

Patient_id => Patient Table
Organization id => Organization Table

Tables: [patient, organization]
Columns: [patient_id, organization_id]
Clause: [where organization_id = 102]

Step 5: After this appropriate syntax of the query is decided based on number of tables present in a list, to perform joins and clause present.

I. SELECT <columns> FROM <table_name>

II. SELECT <columns> FROM <table_name> WHERE <clause>

III. SELECT <columns> FROM <table_name><Joins> WHERE <clause>

Step 6: Build the query from by choosing one of the query templates from the previous step based on the columns, tables and clauses.

So, for the taken example, the outputted query is:
SELECT p.patient_id from patient p, organization o where p.patient_id = o.organization_id where o.organization id = 102

IV. RESULTS

The system was able to produce the equivalent SQL queries for the natural text given by the user, first the natural text received by the user would be sent to the pre-process task for data cleaning and then to the query generator for the formation of SQL queries. The queries returned were executed on the database behind and was able to fetch the results from the database

Natural Text	Query Generated
Fetch employee id where employee name is John	Selectemployee_id from emp where employee_name='John'
Fetch employee name where organization id is 202	Select e.employee_name from empe, organizationo where e.employee_id = o.employee_id
Fetch employee name, department id where organization id is 5	Select e.employee_name,d.department_id from emp e,department d, employee_dept_rltnem,organisation o where e.organisation_id=em.organisation_id and d.organisation_id=em.organisation_id and o.organisation_id=em.organisation_id
Get movie name	Incorrect query

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.512|

|| Volume 9, Issue 6, June 2020 ||

V. CONCLUSION

Interacting with the system using Natural language brings ease for any human being and breaks the linguistic barrier between the user and system interaction. Thus, this system will help people to easily interact with the database and the data present inside it. There is no need for person who has the domain knowledge on how SQL woks or need not to know how to write SQL query or scripts. On account of this application includes a middleware which accepts usersnatural sentences as input and converts them into equivalent and accurate SQL queries and then hit the database to retrieve the required data from the database. The system accepts the input gives the SQL query as output which is user friendly. The system will convert natural text into SQL language query and fetches the required data from database by querying it from the database. The user will also get to see the generated query.

REFERENCES

[1] ZhaorongZong ; Changchun Hong, "On Application of Natural Language Processing in Machine Translation", 2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Huhhot, China, 14-16 Sept. 2018

[2] Ruslan Posevkin ; Igor Bessmertny, "Translation of natural language queries to structured data sources", 2015 9th International Conference on Application of Information and Communication Technologies (AICT), Rostov on Don, Russia, 14-16 Oct. 2015

[3] Zhen Li ;Hanjing Li ; Mo Yu ; Tiejun Zhao ; Sheng Li, "Event Entailment Extraction Based on EM Iteration", 2010 International Conference on Asian Language Processing, Harbin, China, 28-30 Dec. 2010

[4] Prashant Gupta ; Aman Goswami ; Sahil Koul ; Kashinath Sartape, "IQS-intelligent querying system using natural language processing", 2018 3rd International Conference on Mechanical, 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20-22 April 2017

[5] Konstantinos Sintoris ; Kostas Vergidis, "Extracting Business Process Models Using Natural Language Processing (NLP) Techniques", 2017 IEEE 19th Conference on Business Informatics (CBI), Thessaloniki, Greece, 24-27 July 2017

[6] S Thejaswini ; C Indupriya, "Big Data Security Issues and Natural Language Processing", 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 23-25 April 2019Pappas, Vasilis, and Angelos D. Keromytis. "Measuring the deployment hiccups of DNSSEC." International Conference on Advances in Communications. Springer, Berlin, Heidelberg, 2011.