NATURAL LANGUAGE TO SQL CONVERSION USING DEEP LEARNING(LSTM) & NLP

[1]Prof. M.D.Sale, [2]Shreeteeja Salunke,[3] Harshada Jadhav,[4] Shraddha Shekhar, [5]Samiksha Shirgave

[1]Professor, Dept. of Computer Engineering, Sinhgad College of Engineering, Vadgaon, Pune-411041, Maharashtra, India,

[2-5]B.E. Student, Dept. of Computer Engineering, Sinhgad College of Engineering, Vadgaon, Pune-411041, Maharashtra, India.

Abstract : In today's world, large amounts of data are produced within a fraction of seconds. Data can be structured, unstructured and semi-structured. The proposed system focused on structured data handling in a simpler way. The RDBMS uses SQL to perform actions like search, edit, remove and add records to the database. People who are unaware about SQL find it difficult to do these tasks. This system helps such people without wasting time on learning SQL. The main objective is to handle more number queries and effective query formulation using ML Techniques like LSTM. The English language question to equivalent SQL Query formation is done using NLP, LSTM and self-made algorithms for basic AI processing. Due to this architecture, the system handles select, delete, create queries along with various conditions like order by, group by, foreign key, aggregate functions, relation operations, distinct clause. The queries are tested on the SQlite database.The overall accuracy of query generation for six tables is 86.3%.

***IndexTerms* - DBMS, NLP, LSTM, POS, CNN, RDBMS.**

I. INTRODUCTION

The database is basically a collection of organized information in a certain structure so that someone can retrieve , add , delete and edit records as per business need. The need of Database Management System(DBMS) is for maintaining data consistency, data integrity, accurate data, increasing memory and indexing. There is a necessity for experts who have knowledge of Structured Query Language(SQL) as DBMS is handled using SQL. SQL is a query language used for data manipulation in databases. SQL is quite a unique language which requires a strong logical approach to retrieve correct data. It requires a lot of time and capital for learning and understanding this language. And it is not possible for a person to learn SQL who is from a different background though there is a need for interaction with relational databases. This system focuses on converting Natural Language Query into SQL query using Deep Learning and NLP techniques.The working of the system breaks down into three parts: NLP, LSTM and QGenerator. In this system, NLP techniques used are Tokenization and POS Tagging and as for Deep Learning LSTM model is used. As LSTM has a memory unit, it finds accurate table names and attribute names. Final step includes conditional clause handling, query formation and execution of query.

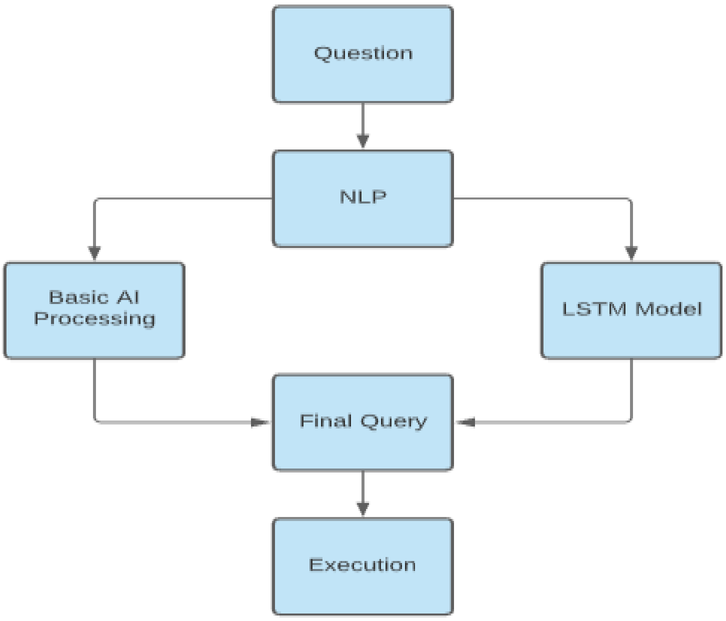

Fig. 1.1: Block diagram of the system

II. LITERATURE SURVEY

The first paper discusses conversion of Natural Language Query to Structured Language Query. The system was purposely made for Training & Placement Cell Officers for retrieving records from student databases without involving any expert in it. The technique used for generating SQL query is Natural Language Processing. In their system, input for natural language is text along with speech. The steps involved in the generation of query are Tokenization, Lexical Analysis, Syntactic Analysis, Semantics analysis. The user can execute the output query on the database. The system allows users to do queries like INSERT, DELETE, UPDATE. But along this simplicity there are limitations of the system. The system cannot handle ambiguity in queries, cannot handle complex and nested queries, cannot handle queries if the sentence contains both SELECT and INSERT query together.[1]

The second paper converts simple natural language into its equivalent SQL query using semantic grammar. The input is taken in the text format. The process used for conversion is NLP techniques. The NLP techniques used in this system are Morphological Analysis(Tokenization), Lexical Analysis, Syntactic Analysis,Semantic Analysis and Mapping. After this process the SQL query is executed on the database and accordingly the result is displayed. The limitations of the system is that it needs more time and coding to handle complex queries, it may need to handle few bugs, and multiple sentences may take some time for processing.[2]

The third paper discusses how natural language queries are converted into SQL query effectively. The input is taken in the format of text. This sentence is passed to OpenNLP techniques like Tokenization, POS Tagging, Lemmatization, Stemming to get the sentence in desired form. To find links between different tables, Table Linking Algorithm(TLA) is used. The accuracy of the system is 82% for 6 tables. The limitations for the system is that it is unable to find conditional clause elements i.e. queries which have another query embedded, queries with qualitative quantifiers cannot be processed.[3]

The last paper explains the Long Short Term Memory and its working.This paper also discusses how Deep Learning Algorithms are used in text classification modules and how accuracy can be increased using LSTM module and Convolutional Neural Network(CNN). This paper gives us an overall idea about the LSTM model.[4]

III. Working Flow

1) query. E.g. Show all records of the singer.
2) Then, input query will be converted into tokens and tagged words list.
3) The query type is extracted and according to the query type, function call occurs.
4) Using the tokens and LSTM model, table names and attribute names are predicted.
5) The presence of various conditions like between, not between, like, not like, relation operations, aggregate functions, order by , group by and foreign key relation will be checked . Then, where clause is created .
6) The integration of attributes, tables and conditions has to be done to generate more accurate SQL Query.
7) Finally, this generated query is displayed to the user. And execution of that query is done as per user wish.

IV. Modules

The natural language query to SQL generation is processed under three main modules . i.e NLP , LSTM and QGenerator.

4.1 NLP MODULE

Natural Language Processing deals with interaction with humans and computers. It is simply used to convert natural language into a language which the computer understands. It gives computers the ability to understand human language. Conversion of natural language includes many NLP techniques like tokenization, lemmatization, pos tagging, stemming etc. The NLP techniques used for proposed system are:

1) Tokenization: It is a process of splitting a sentence, paragraph or entire document into individual words. These words are called tokens.

E.g. 'find salary of the employee'

Tokens: ['find', 'salary', 'of', 'the', 'employee']

2) POS Tagging: In this process, tokens generated are mapped to the POS (Parts of Speech) tag to derive semantic meaning from them. The main objective of this step is to find nouns, verbs from the sentence which will help to find tables and attributes.

E.g. 'show the employee table'

Tagged list: { ('show,' 'VB'), ('the', 'DT'), ('employee', 'NN'), ('table', 'NN')}

where VB-Verb, NN-Noun, JJ-Adjective, DT-Determiner.

4.2 LSTM (Long Short-Term Memory)

LSTM is an advanced artificial recurrent neural network used in the field of deep learning. It can process single data points like images and whole sequences of data like speech or video too. Recurrent Neural Networks suffer from short-term memory. If the sentence is very long, it tends to forget the previous important information as it goes further. This happens because of the vanishing gradient problem. The gradient shrinks as it propagates back and if it becomes very small it does not contribute to the learning. LSTMs are used to overcome this problem. LSTM is composed of a forget gate, input gate, cell (memory unit) and output gate.

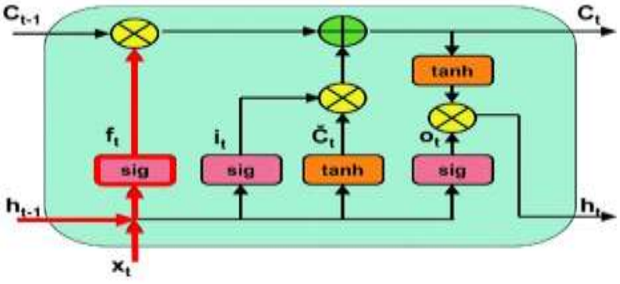

Fig. 4.2: Single Cell for LSTM

1) Forget Gate : The Forget gate decides what information to keep and what to discard. The h_{t-1} is the information from the previous hidden state (previous cell) and x_t is the information from the current cell. The □ represents the sigmoid function.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)\#(1)$$

2) Input Gate: The Input gate updates the current cell state. It decides which information is important and which is not by transforming the values into 0 to 1 using the sigmoid function. 0 means it is not important and 1 means it is important and the information should persist. Further these inputs are also passed to a tanh function and transformed into -1 to 1 in order to regulate the network. Both the outputs of sigmoid and tanh function are then multiplied and the sigmoid function decides the important and unimportant information.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)\#(2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t]) + b_C\#(3)$$

3) Cell state: All the information gained is then used to calculate the new cell state. The cell state is first multiplied with the output of the forget gate. This has a possibility of dropping values in the cell state if it gets multiplied by values near 0. Then we take the output from the input gate and do a pointwise addition which updates the cell state to new values that the neural network finds relevant.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t\#(4)$$

4) Output gate: The last gate which is the Output gate decides what the next hidden state should be. The previous hidden state and the current input is passed to a sigmoid function. Then the newly modified cell state is passed to the tanh function. The tanh output is multiplied with the sigmoid output to decide what information the hidden state should carry. The output is the hidden state. The new cell state C_t and the new hidden state is h, then carried over to the next time step. The formulation are given by:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)\#(5)$$

$$h_t = o_t * \tanh(C_t)\#(6)$$

4.3 QGenerator (Query Generator)

The half part of implementation of the system comes under this module. The extraction of the query type, finding the conditions , presence of aggregate functions and distinct clauses are checked by self-made algorithms. The conditions handled by system are as follows:

1) Relation Operators: greater than equal to (>=), less than equal to (<=), less than (<), more than (>) and equal to (=).
2) Aggregate Functions: min(), max(), avg(), sum(), count(attribute), count(*).
3) Distinct Clause: This clause helps to find unique values of a column. This system handles 'distinct' along with any of the aggregate functions.
4) Between and Not between : One will want records between numerical values or skip records between particular ranges. At that time, without repeating attribute names and without using the logical operations like and,or in the 'where' clause, a query will be generated.
5) Like/ Not Like : These operators are used to find patterns for a column. Our system handles few of the patterns.
6) Order by
7) Group by
8) Having clause
9) Foreign Key relationship between 2 tables
10) Inner join

IV. Result And Discussion

We tested natural language queries on a database containing 6 tables. We tried a total of 204 queries out of which 176 were successfully converted to equivalent SQL queries. The accuracy comes out to be about 86%. From the remaining queries 14 were partially correct and 14 were incorrect.

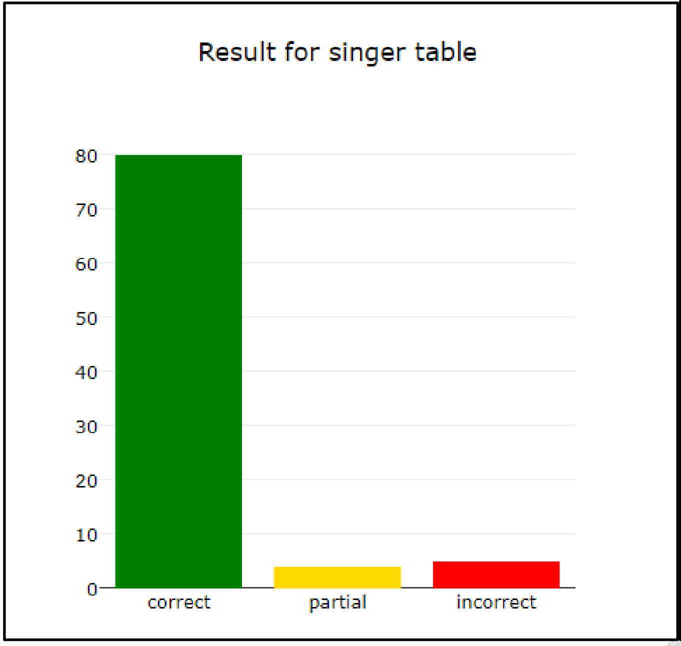

Fig. 5.1: Result for singer table

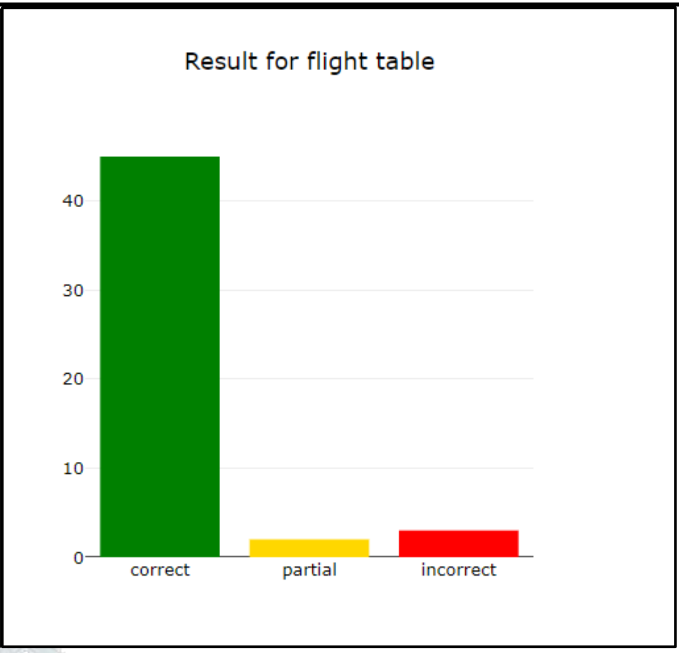

Fig. 5.2: Result for flight table

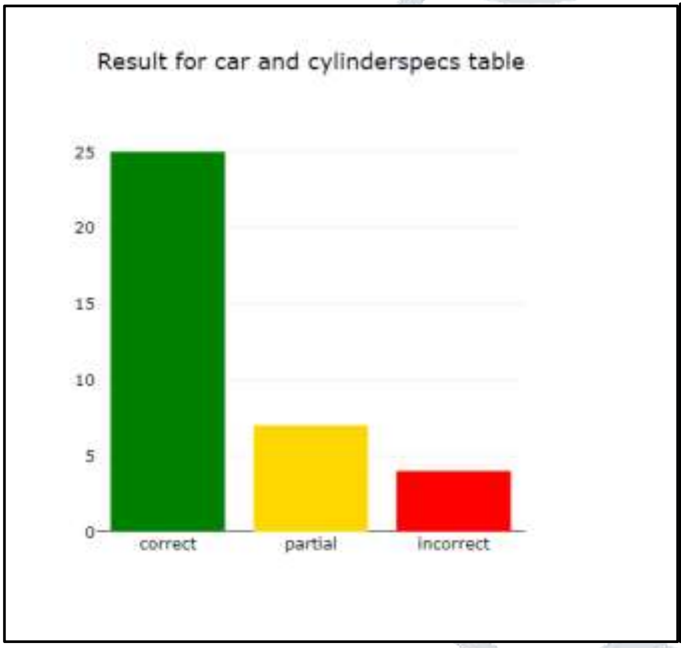

Fig. 5.3: Result for car and cylinderspecs table

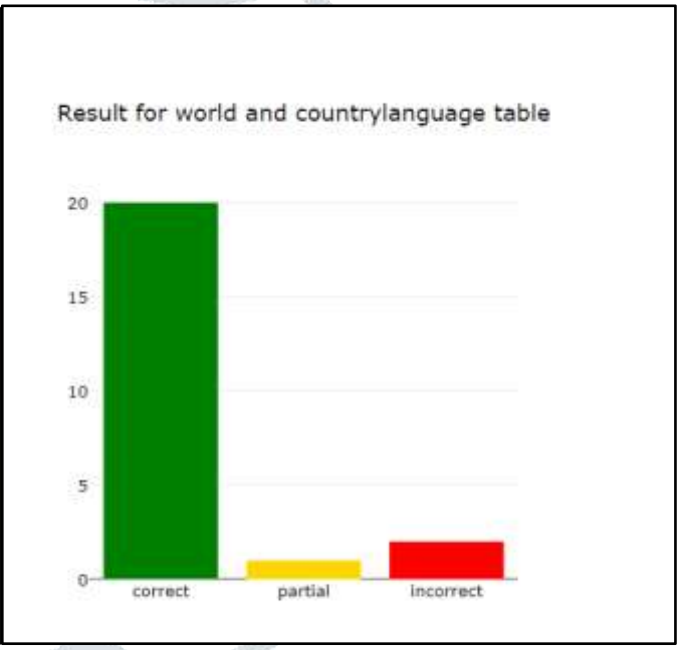

Fig. 5.4: Result for world and countrylanguage table

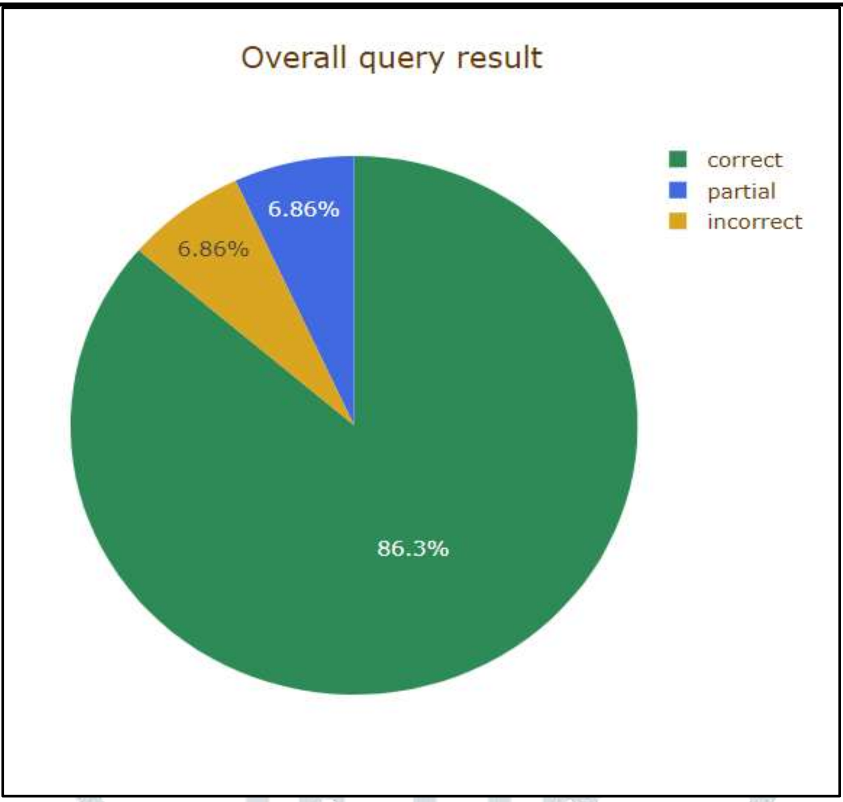

Fig. 5.5: Overall query result

Table -1: Analysis of Query Formation

No. of Tables	Correct Queries	Partial Queries	Incorrect Queries	Total	Accuracy %
Only one Table	129	6	8	143	90.20
Two Tables	45	8	6	59	76.27

5.1 Correct output queries:

Query 1: show details of singers
Converted Query: SELECT * FROM singer;

Query 2: how many singers are from germany and france
Converted Query: SELECT count(*) FROM singer WHERE country = "germany" or country = "france" ;

Query 3: show language and country of conid 2
Converted Query: SELECT countrylanguage.language, world.country,world.conid FROM world INNER JOIN countrylanguage ON world.conid= countrylanguage.conid WHERE world.conid=2;

Query 4: delete all records of flight
Converted Query: DELETE From flight ;

Query 5: generate table with tablename student and fields are (id int, name varchar(20), birth DATE, rollno int). Make id as unique and not null. Make rollno as the primary key. Make birth as not null. Make the name as not null.
Converted Query: CREATE Table student (id int UNIQUE NOT NULL, name varchar(20) NOT NULL, birth DATE NOT NULL, rollno int, PRIMARY KEY (rollno));

Query 6: delete records of singers id between 10 and 20 or name starting with a
Converted Query: DELETE From singer where id BETWEEN 10 AND 20 or name LIKE 'a%' ;

5.2 Partially correct queries:

Query 1: display the songs,name of singers, year where age is equal to 25
Converted Query: SELECT name,year,songs,age FROM singer WHERE age = 25 ;

Query 2: show flightno where airline is panorama
Converted Query: SELECT airline,flightno FROM flight WHERE airline = "panorama" ;

5.3 Incorrect queries:

Query 1: show flightno where country is spain
Converted Query: SELECT flightno,country FROM flight;

Query 2: show ***maximun*** age of singers
Converted Query: SELECT age FROM singer WHERE year = "maximun" ;

Mostly the queries are incorrect due to grammatical errors, spelling mistakes.

VI. Advantages And Limitations

6.1 Advantages:

1) System can be handled simply by knowing English Knowledge.
2) Making data-driven insights accessible to users without SQL skills.
3) Reduces time in learning structured query language.
4) It can be used in the student records database system, employee information system,etc.
5) User friendly, easy to understand and straightforward UI.

6.2 Limitations:

1) Language Constraint : The input query should be in English Language, in addition to that it must be grammatically accurate.
2) Abbreviate words : Avoid using short forms for words or phrases. Eg. gt or >= for greater than, lt or <= for less than.
3) Create Table Constraint : The column names with their datatype should be in the list.(The column names of the table while creation should be well specified along with their respective data-types that too in the form of a list.)

VII. Conclusion And Future Scope

This system successfully converts Natural Language query into SQL query using LSTM model and NLP techniques. This system simply uses English as an input language to retrieve data from the database. The system has the facility to execute generated query on database. The User Interface is simple and user friendly. This system successfully handles query types like: SELECT, CREATE, DELETE, GROUPBY, ORDERBY, HAVING, DISTINCT clauses, AGGREGATE FUNCTIONS - MIN, MAX, SUM, AVG, COUNT etc. and LIKE, BETWEEN clauses.

The future scope can be implemented to enhance the functionality and quality of query generation. Following points can be useful for future scope:

- This system doesn’t handle INSERT, UPDATE queries, this can be developed in future.
- This system handles only DML and DDL queries so the system can be improved which can handle DCL queries like grant, revoke, etc.
- This system handles only MySQL database and sqlite for database connectivity. Developing the system to handle other databases.
- Pattern queries can be improvised.

References

[1] Abhilasha Kate, Satish Kamble, Aishwarya Bodkhe, Mrunal Joshi, “Conversion of Natural Language Query to SQL Query”, 2nd International Conference on Electronics, Communication, and Aerospace Technology (ICECA 2018) IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1-5386-0965-1.

[2] Prof. Debarati Ghosal, Tejas Waghmare, Vivek Satam, Chinmay Hajirnis, “SQL Query Formation Using Natural Language Processing (NLP)” , International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016.

[3] Amit Pagrut, Ishant Pakmode, Shambhoo Kariya, Vibhavari Kamble and Yashodhara Haribhakta, “Automated SQL Query Generator By Understanding A
Natural Language Statement”, International Journal on Natural Language Computing (IJNLC) Vol.7, No.3, June 2018.

[4] Yuandong Luan, Shafu Lin, “Research on Text Classification Based on CNN and LSTM”, 2019 IEEE International Conference on Artificial Intelligence and Computer Applications, March 2019.

[5] Sachin Kumar, Ashish Kumar, Dr. Pinaki Mitra, Girish Sundaram, “System and Methods for Converting Speech to

[6] SQL”, International Conference on “Emerging Research in Computing, Information, Communication, and Applications”, ERCICA 2013 pp: 291-298, Published by Elsevier Ltd ISBN:9789351071020.

[7] Tanzim Mahmud, K. M. Azharul Hasan, Mahtab Ahmed, Thwoi Hla Ching Chak, “A Rule Based Approach for NLP Based Query Processing” , International Conference on Electrical Information and Communication Technology (EICT 2015).

[8] Aditya Narhe, Chaitanya Mohite, Rushikesh Kashid, Pratik Tade, Santosh Waghmode,” SQL Query Formation for Database System using NLP” , International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 8 Issue 12, December-2019.

[9] https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/

[10] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[11] https://www.datasciencecentral.com/profiles/blogs/top-nlp-algorithms-amp-concepts

[12] http://www.iaees.org/publications/journals/selforganizology/articles/2016-3(3)/algorithm-to-transform-natural-language-into-SQL-queries.pdf

[13] https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn