IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Formulating SQL Queries from Natural Language Processing for Students Using Mobile Learning

Parth Mody
B Tech, Computer Engineering
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India

Maanaav Motiramani
B Tech, Computer Engineering
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India

Param Sejpal
B Tech, Computer Engineering
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India

Abhitay Shinde
B Tech, Computer Engineering
Mukesh Patel School of Technology Management and Engineering, NMIMS
Mumbai, India

Abstract— **Mobile learning opens new worlds of information and personal development. To enable more personalized learning via mobile devices, several clever approaches should be implemented into mobile-assisted learning systems. This paper examines and explores several systems created with natural language processing (NLP) to extract relevant information from a database by utilizing structured natural language questions as input and SQL queries as output. Natural Language Processing (NLP) tools can be used to evaluate students' flaws throughout the mobile learning assessment process. Furthermore, the approach broadens its use beyond student learning by incorporating CSV data retrieval capabilities. Users, such as placement cell workers dealing with student databases, can perform natural language searches to retrieve useful information from CSV files. The design of the proposed model includes a user interface for submitting English inquiries, followed by NLP modules for analyzing the queries and mapping them to SQL queries. The SQL queries may then be run to obtain the necessary data from the database. It illustrates the power of natural language processing techniques in supporting mobile learning and enhancing data retrieval operations.**

Keywords— Natural Language Processing (NLP), Structured Query Language (SQL), CSV, Tokenization, POS tagging, Chunking, Parsing, Featured context free language, Speech to Text, Mobile Learning, User Experience, Data retrieval.

I. Introduction

Mobile learning opens new avenues for personal growth and learning. To deliver more tailored learning utilizing mobile phones, certain intelligent tactics should be included to mobile-assisted learning systems. This project assists users who are unfamiliar with query languages such as SQL (Structured Query Language) and the complications that come with them. When learning SQL, students run into a variety of issues. To discover a solution to this problem, a model has been developed. This is useful for folks who are new to SQL and have little experience with it.

The English language speech is used as the input and is then transformed to text. To produce an equivalent SQL query from a natural language query, many procedures such as tokenization, lemmatization, syntactic, and semantic analysis are used. Then comparison of the student's SQL query to the NLP-generated query is done. Natural Language Processing (NLP) techniques can also be employed to examine students' mistakes during the mobile learning evaluation process.

This project assists users who are unfamiliar with query languages such as SQL (Structured Query Language) and the complications that come with them. The technique can be used not only by students but also to extract data by placement cell personnel who work with student databases. CSV files (Comma Separated Values) are extensively used as a basic data format. Many of the data files that are published in open source and used by companies are frequently saved in CSV files due to their simple structure and ease of generation. The standard keyword-matching technique, which cannot describe the parameters for searching or processing any data on the search, limits searching for or retrieving expected data from CSV files. This paper introduces a novel approach that enables users to quickly obtain data from CSV files using natural language.

The subsequent sections of this paper will begin by providing an overview of the fundamental concepts and principles related to NLP and mobile learning. Following that, the existing literature will be examined, encompassing methodologies, findings, and implications of studies that have explored the application of NLP in mobile learning environments. Finally, a synthesis of the key findings will be presented.

II. Literature Review

The paper "NLP-based error analysis and dynamic motivation techniques in mobile learning" by *Christos Troussas; Akrivi Krouska* presents effective analysis of students' errors using NLP and dynamic techniques for motivating them in a computer-assisted language learning environment [2]. Similar techniques in mobile learning can be used for SQL query learning for students. Students can receive motivation in case of making errors.

In a paper by *Jasmeen Kaur; Bhawna Chauhan; Jantinder* "Implementation of Query Processor Using Automata and Natural Language Processing" Kaur presents the method of Querying in databases in natural language for data access [3], for the newbie's who have less knowledge about complicated database query languages such as SQL. This paper emphasizes on the structural designing methods for translating English Query into SQL using automata. The paper "Information Processing and Retrieval from CSV File by Natural Language" by *Chalermpol Tapsai* presents a CSV Data Processing Model (CSVDPM) that will allow users to retrieve and process data in CSV files by using their own familiar natural language without any additional training in extra application programs or computer languages [1].

The paper by Mohit Dua, Sandeep Kumar, and Zorawar Singh Virk presents the HLIDB (Hindi Language Interface to Database), a system designed to facilitate querying and interaction with databases using the Hindi language [4]. The HLIDB architecture consists of four phases: Tokenizer, Mapper, Query Generator, and Database Management System (DBMS). In the Tokenizer phase, Hindi sentences are split into tokens, which are then matched with the tokens stored in a lexicon. The Mapper phase saves the corresponding English word for each token, along with its type. The Query Generator formulates the SQL query based on the provided tokens, such as table names, column names, conditions, and commands. Finally, the DBMS executes the SQL query and converts the result into the user's preferred language. HLIDB enables users to input queries in Hindi, supporting various database operations such as selection, updating, and deletion. However, the system has certain limitations that need to be considered. One limitation is that the process of creating a table in the database is manual, requiring the specification of table and column names. This manual approach can be tedious and time-consuming, potentially hindering the user experience. Additionally, the current model of HLIDB only works with a single database, limiting its applicability to broader contexts where multiple databases may be involved.

The paper authored by Uma M, Sneha V, Sneha G, Bhuvana J, and Bharathi B aims to develop a system using natural language processing (NLP) techniques to enable users to access information from the railways reservation database by inputting structured natural language questions and receiving SQL queries as output [5]. The system follows a series of steps, including tokenization, lemmatization, parts of speech (POS) tagging, parsing, and mapping. The dataset used for system development consists of 2880 structured natural language queries related to train fare and seat availability. The achieved accuracy of the proposed system is reported to be 98.89%. The system operates by receiving an English query in text form, which is then processed through various NLP modules. Following the NLP phase, a mapping phase detects attributes in the English query, maps them to form the final SQL query, and retrieves the required information from the database to provide it to the user. The paper reports an accuracy of 98.89% based on the evaluation of 2880 test cases. Precision, recall, F1 score, and accuracy metrics are provided, indicating the effectiveness of the developed system.

The research conducted by Mithila Kundu Barsha, K. M. Azharul Hasan, and Irfat Ara focuses on the development of a Natural Language Interface to Database (NLID) using the Pattern Matching approach [6]. The NLID system utilizes regular expressions (regex) to define patterns, which are special text strings used to describe search patterns recognized by finite automata. In the NLID architecture, there are two main tasks: the linguistic task (NLP) and the database query task. The linguistic task involves processing the NL input, while the database query task is responsible for generating SQL statements and interacting with the database. To begin, the NL input query is tokenized into several tokens, breaking it down into meaningful units. The system then checks if these tokens match the predefined regular expression pattern. If there is a match, the process proceeds further; otherwise, it stops as the input does not conform to any of the defined patterns. The focus is primarily on the "where" condition in the input query, specifically the given value of a field. By utilizing regular expressions, the system can effectively handle different variations of the input query and identify the relevant SQL statements. In the database query task, the system performs the necessary database operations. The authors have developed a prototype database system for this purpose. The evaluation of the NLID system was conducted in the student information retrieval domain. The system was tested using a set of 50 questions based on the table entity. The performance of the NLID system was measured in terms of accuracy, calculated as the ratio of correct translations to MySQL queries attempted. The results reported an accuracy of 90.42%, with 85 out of 94 questions correctly translated. These findings demonstrate the effectiveness of the Pattern Matching approach and the use of regular expressions in developing a Natural Language Interface to Database.

The authors of the paper [7] presented an approach for the automated conversion of Natural Language Query to Structured Query Language for managing a relational database. The method proposed by them processes the NLQ using tokenization, lexical analysis, syntactic analysis, and then semantic analysis to obtain the SQL Query which would be executed in the database to produce the desired results. Their goal is to design a system for Training and Placement cell officers in a college who don't possess the technical expertise to use SQL to traverse the student database.

In [8], the authors developed a method to build a Natural Language Interface to Data Bases (NLIDB) system. The system first parses the input and then uses the syntactic knowledge, semantic knowledge, expert system, prolog and amzi to achieve the results. By using an expert system along with the traditional NLP-based approach, the paper aims to develop a completely automated system.

Paper [9] aims to build a process by which the SQL query is generated by using a machine learning algorithm that would analyze the presented natural language query and database. They used pre-processing techniques: lowercase conversion, removing escaped words, tokenization, PoS tagging, and NLP algorithms: word similarity, Jaro-Winkler matching algorithm, and Naive Bayes. This method is independent of language, domain, and database model, implementing a system with these factors would only increase the accuracy of the results. This implementation can be expanded in the future to increase the algorithm's performance and increase text extraction accuracy.

This paper [10] examines text-to-SQL translation using neural networks. The authors explore existing models, implement their own using Transformer and CNN, and analyze their performance. They find that incorporating SQL structures improves translation accuracy. Transformer and CNN encoders perform similarly to LSTM encoders in aggregator prediction. The project uses PyTorch with ADAM for training and regularization. Evaluation includes separate breakdowns for AGGandSELECT_COL predictions using exact match. The authors adapt Seq2SQL code for Python 3 compatibility. The study discusses the use of the WikiSQL dataset for query accuracy assessment.

This paper [11] proposes a new method to access databases using natural languages like Hindi, Marathi, etc. It utilizes Natural Language Processing (NLP), which involves mathematical and computational modeling of language and the development of various systems. The paper presents the architecture of the NLIDB system, which incorporates both semantic and syntactic grammar systems. It references a previous system called LIFERILADDER, designed as a natural language interface for a database containing information about US Navy ships. LIFERILADDER employed a semantic grammar to parse questions and query a distributed database. The paper also discusses several modules utilized in the NLIDB system, including a Spelling Checker, Ambiguity Reduction, and Token Analyzer.

This paper [12] explores natural language processing techniques for creating a user interface to access structured data. The authors discuss computational linguistics methods, machine translation, text annotation, and expert systems. They present a prototype interface that converts user queries into SQL queries for a database of program libraries and frameworks. The paper also touches on optimal user interaction strategies. In summary, it provides a literature survey on using natural language processing to develop user interfaces for structured data source.

III. Proposed Methology

The proposed model presents the idea of extracting SQL query using natural language processing. Python from google colab and PyCharm has been used for implementation of the model.

Retrieving the required information from a database is quite difficult for any common man and requires a lot of effort which needs the knowledge of the database structure. A Database Management System is incapable of dealing with queries framed in any other languages other than the standard database languages. So to make the retrieval more effortless and interactive for naïve user, our proposed work provides a facility through which a user is free to pose a query in English, which will be processed by several modules to form an equivalent SQL query.

The architecture of the proposed model is described below.

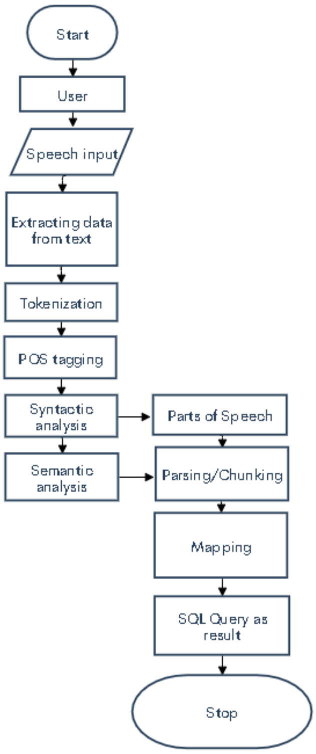


Fig. 1. Architecture of model

User submits an English query in the text form which is then sent into several natural language processing (NLP) modules. This NLP phase is followed by a mapping phase in which the attributes are detected in the English query, mapped to form the final SQL query, and may then be fed into the database to retrieve the required information and provide it to the user. Once the SQL query is generated the retrieval of data from DB will be an easy task.

Algorithm:

Input: Natural Language query in English text.
Output: SQL query

1) *User interface: user interacts with the system via GUI or voice input.*
2) *Lowercase conversion*
3) *Stop word removal.*
4) *Tokenize the input into list of words.*
5) *Perform POS tagging.*
6) *Relations-Attributes-Clauses Identifier*
7) *Query Formation*

The second part of the project is CSV data retrieval which has been implemented using hugging face table-question-answering transformer pipeline. The pipelines are a great and easy way to use models for inference. These pipelines are objects that abstract most of the complex code from the library, offering a simple API dedicated to several tasks, including Named Entity Recognition, Masked Language Modeling, Sentiment Analysis, Feature Extraction and Question Answering.

IV. Implementation

In the desired approach, some predefined structures are employed and the system is trained accordingly. The primary advantage of these structures is that they can be expanded whenever some new knowledge is discovered. It uses

- The escape word set which contains the list of stop words that occur in NLQ as shown in Table 1.
- A Noun set that contains all elements which are nouns in Table 2 and strictly limited to provided

data Dictionary of attribute & relation names and further Relation set and Attribute set are extracted from N.

- The semantic set contains the list of all possible semantics related to table names and fields in the database as shown in Table 3 and 4.
- A Variable set that consists of all String and Integer variables used in forming clauses.
- A Relation set consists of relation names that are encountered in user query or are added by analyzing the attribute names present in the NL query.
- An Attribute set that contains all attributes present in the user query as shown in Table 4.
- An Ambiguity check set that contains all attribute fields whose name are used in multiple relation as a field name excluding keys.
- The Conjunction training set consists of the list of Conjunctive clauses which occur in NL query and this set is generated at runtime. Whenever new conjunctive clause is encountered, it is appended to the existing Conjunction training set.

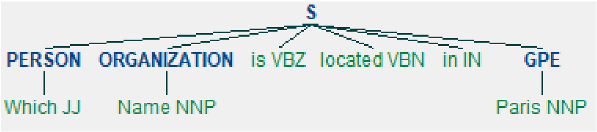


Fig. 2. Chunked tree

Escape words	
are	is
have	located
in	above

Fig. 3. Stop words

Noun	
Name	Salesman_id
Commission	City

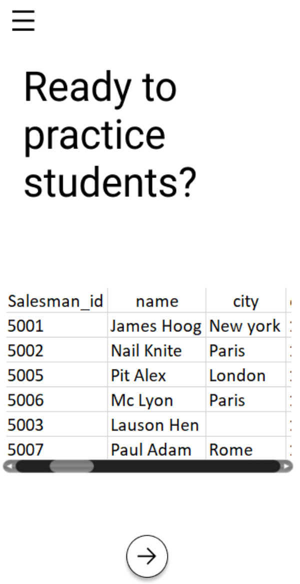
Fig. 4. Noun Set

Rules for attributes in relations		
Rule	Rule Symbol	Rule Description
Name	Salesman	Attribute for Relation 'Salesman'
Commission	Salesman	Attribute for Relation 'Salesman'
Salesman_id	Salesman	Attribute for Relation 'Salesman'
City	Salesman	Attribute for Relation 'Salesman'

Fig. 5. Rules for attributes

Rules for relation name		
Rule	Rule Symbol	Rule Description
City	City	Attribute for Relation 'City'
Cities	City	Attribute for Relation 'City'
Name	Salesman	Attribute for Relation 'Salesman'
Salesman	Salesman	Attribute for Relation 'Salesman'
Employee	Salesman	Attribute for Relation 'Salesman'

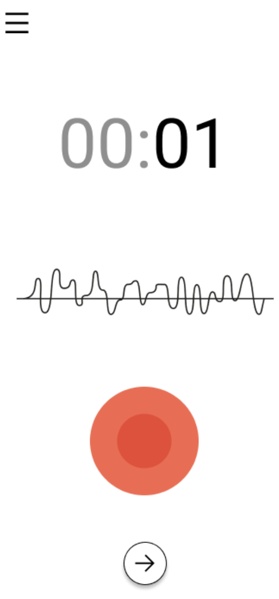
Fig. 6. Rules for relation name



Ready to practice students?

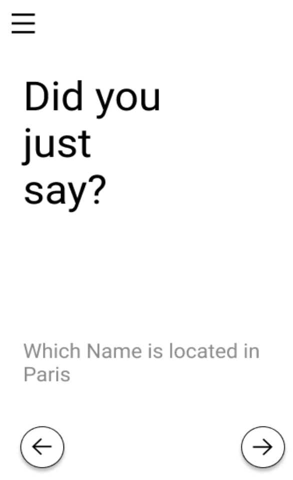
Salesman_id	name	city
5001	James Hoog	New york
5002	Nail Knite	Paris
5005	Pit Alex	London
5006	Mc Lyon	Paris
5003	Lauson Hen	
5007	Paul Adam	Rome

Fig. 7. Homepage



00:01

Fig. 8. Voice input with db

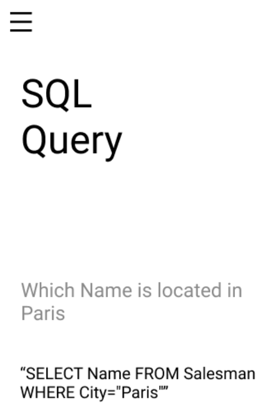


☰

Did you just say?

Which Name is located in Paris

Fig. 9. Voice to text



☰

SQL Query

Which Name is located in Paris

“SELECT Name FROM Salesman WHERE City="Paris"”

Fig. 10. SQL Query o/p

Algorithm:

Input: Natural Language query in English text.
Output: CSV data

1) *User interface: user interacts with the system via GUI or voice input.*
2) *Pipeline abstraction: table-question-answering.*
3) *Read CSV*
4) *Convert all columns into string type.*
5) *Query output*

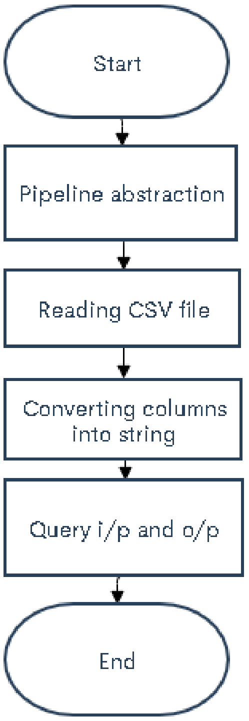
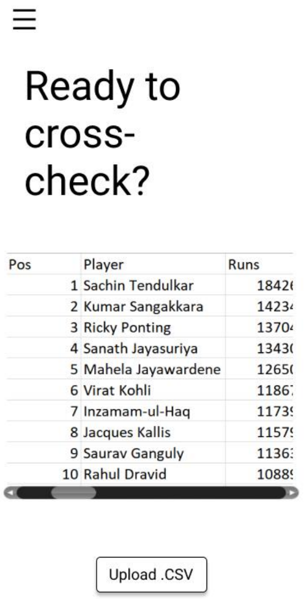


Fig. 11. Algorithm

The conventional keyword-matching technique limits searching for or retrieving expected data from CSV files since it cannot explain the conditions for searching or processing any data on the search. This study describes a novel method for quickly extracting data from CSV files using natural language. This CSV data retrieval can be used by people like salesman.



☰

Ready to cross-check?

Pos	Player	Runs
1	Sachin Tendulkar	18426
2	Kumar Sangakkara	14234
3	Ricky Ponting	13704
4	Sanath Jayasuriya	13430
5	Mahela Jayawardene	12650
6	Virat Kohli	11867
7	Inzamam-ul-Haq	11739
8	Jacques Kallis	11579
9	Saurav Ganguly	11363
10	Rahul Dravid	10889

Fig. 12. Homepage

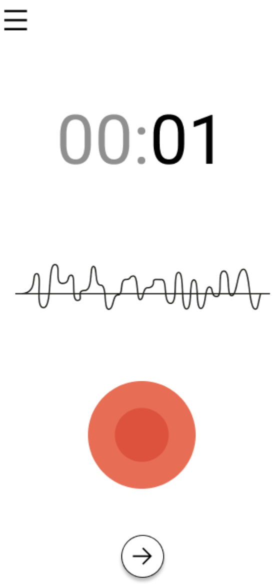
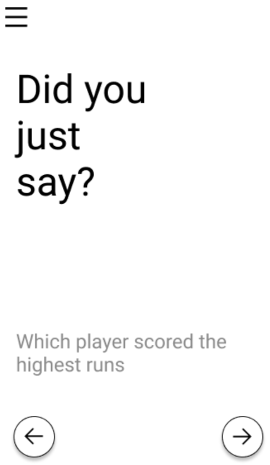
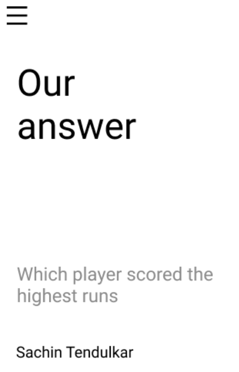


Fig. 13. Voice input with db


Did you just say?

Which player scored the highest runs

Fig. 14. Voice to text


Our answer

Which player scored the highest runs

Sachin Tendulkar

Fig. 15. CSV output

V. Conclusion

An innovative approach to mobile learning and data retrieval by leveraging natural language processing (NLP) techniques and SQL queries is proposed. The developed model provides a user-friendly interface that allows students and other users to interact with databases using English queries, which are then translated into SQL queries for efficient data retrieval. Through the incorporation of NLP modules such as tokenization, POS tagging, and syntactic and semantic analysis, the system is able to accurately understand the user's intent and map it to the appropriate SQL query.

Comparing this approach with existing literature, several papers have explored similar themes. However, their approach lacked the mobile learning aspect and focused solely on desktop applications. In contrast, the proposed model specifically targets mobile learning environments, providing a more accessible and flexible platform for students to interact with databases on the go.

Furthermore, this model extends its functionality to CSV data retrieval, which distinguishes it from existing literature. By enabling users to extract information from CSV files using natural language, the model provides a convenient and intuitive way to interact with data stored in such formats. This capability not only enhances the user experience but also expands the practical applications of the model beyond traditional database interactions.

To conclude SQL commands has become a power tool for providing the capability of creating and manipulating a wide variety of database objects, it is not only used to create or manipulate data but to also retrieve large amount of data for the database quickly and efficiently. SQL is not only used by students but is also used by software developers and database Administrators in writing Data Integration Scripts and business analyst. Data Science tools depend highly on SQL. Big data tools such as Spark, Impala are dependent on SQL. It is one of the demanding industrial skills in today’s world. SQL can be challenging at times for people who are not familiar with it. The proposed model in the paper not only eliminates the fear of not knowing the concept of SQL but even saves the time of the end user. By providing a user-friendly interface, incorporating dynamic query generation, and extending functionality to CSV data retrieval, this model offers a comprehensive solution for efficient and intuitive interactions with databases in the mobile learning context.

VI. References

[1] Chalermpol Tapsai, “Information Processing and Retrieval from CSV File by Natural Language”. Published in 2018 at IEEE 3rd International Conference on Communication and information Systems (ICCIS).

[2] Christos Troussas; Akrivi Krouska; Maria Virvou, “NLP-based error analysis and dynamic motivation techniques in mobile learning”. Published in 2019 at 10th International Conference on Information, Intelligence, Systems and Applications (IISA).

[3] Jasmeen Kaur; Bhawna chauhan; Jatinder Kaur Korepal, “Implementation of Query Processor Using Automata and Natural Language Processing”. Published in 2013 at International Journal of Scientific and Research Publications.

[4] M. Dua, S. Kumar and Z. S. Virk, "Hindi Language Graphical User Interface to Database Management System," 2013 12th International Conference on Machine Learning and Applications, Miami, FL, USA, 2013, pp. 555-559, doi: 10.1109/ICMLA.2013.176.

[5] M. Uma, V. Sneha, G. Sneha, J. Bhuvana and B. Bharathi, "Formation of SQL from Natural Language Query using NLP," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-5, doi: 10.1109/ICCIDS.2019.8862080.

[6] M. K. Barsha, K. M. Azharul Hasan and I. Ara, "Natural Language Interface to Database by Regular Expression Generation," 2021 5th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2021, pp. 1-6, doi: 10.1109/EICT54103.2021.9733592.

[7] A. Kate, S. Kamble, A. Bodkhe and M. Joshi, "Conversion of Natural Language Query to SQL Query," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 488-491, doi: 10.1109/ICECA.2018.8474639.

[8] F. Siasar djahantighi, M. Norouzifard, S. H. Davarpanah and M. H. Shenassa, "Using natural language processing in order to create SQL queries," 2008 International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 2008, pp. 600-604, doi: 10.1109/ICCCE.2008.4580674.

[9] M. Arefin, K. M. Hossen and M. N. Uddin, "Natural Language Query to SQL Conversion Using Machine Learning Approach," 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/STI53101.2021.9732586.

[10] Di bai, W. Jiang, Y. He, “Text-to-SQL Translation with Various Neural Networks”, CS224N Project Final Report. Stanford University (2012).

[11] P. Dhomne, S. Gajbhiye, T. Warambhe, V. Bhagat, "Accessing Database Using NLP", Volume: 02 Issue: 12 IJRET (Dec-2013)

[12] R. Posevkin & I. Bessmertny, “Translation of natural language queries to structured data sources”, Procedia Computer Science, 88, 3-8. (2016)