IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 1, April 2025

Impact Factor: 7.67

AI SQL Assistant using NLP

Pritesh Patil[1], Vaibhav Datkhil[2], Ashish Birajdar[3], Atharv Jagdale[4]

Professor, Department of IT[1]

Students, Department of IT[2,3,4]

AISSMS Institute of Information Technology, Pune, Maharashtra, India

Abstract*: This paper designs an NLP-based system which integrates databases to turn textual queries into SQL statements. Users can operate the system through natural language searches since they do not need to learn SQL commands. The system detects main information in user input and uses predefined keywords to guide data extraction and cleanup operations. The classifier uses multinomial logistic regression to analyze data through an identification process that recognizes selection or insertion or update or deletion queries among others. The system generates an appropriate SQL query for execution against the database since it determines the query type. The system generates user-friendly presentation of results to the end-user. The system enables users without programming experience to operate complex database systems by removing the requirement to create SQL commands. Modern enterprises must rely on simple user interfaces to broaden data accessibility because they implement information systems at an increasing rate. The system enables more users to make data-based decisions through its query-generation facilitation capability. The deployment of machine learning and NLP methods ensures continuous improvement of query understanding for system users through time therefore delivering more specific results.Through these measures the system provides democratic access to data that enables users from various industries to obtain database insights without requiring specialized technical qualifications.*

Keywords: Natural Language Query, Text-to-text, Structed Query Language (SQL), Database Query

I. INTRODUCTION

In the task of automatic Structured Query Language (SQL) query generation. Relational databases make use of the established query language titled SQL. Successfully writing and running SQL queries stands essential for students attending relational database courses. The developing field of Artificial Intelligence known as Natural Language Processing leads Human-Computer Interaction programs by example.[2]The operations which benefit from NLP include information retrieval alongside machine translation and linguistic analysis. The system functions to establish seamless communication between computer systems and human users through processes that don't require complex command remembering or procedural understanding. Natural language questions allow users to retrieve information easily since the system supports both expert and novice users who might lack familiarity with SQL and similar advanced query languages. The framework serves as a solution to address problems in natural language text analysis as well as generation for various purposes.

The current system identifies which type of SQL query the user demands from a range including SELECT, UPDATE, DELETE and other queries. Training accuracy serves as the essential part for this method to successfully predict user intentions. The model generates SQL code from its prediction to execute the required statement after making its output.

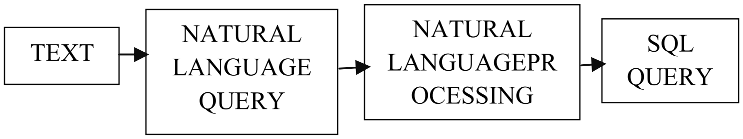
The paper continues with the following sequence: section 1 provides introduction of project. The paper starts with problem description in section 2 while section 3 delivers a short analysis of existing literature. Section 4 offers a concise description of the proposed system. The following sections make up this work: architecture of the system appears in section 5 and system results in section 6 while future project scope is discussed in section 7. Finally Section 8 conclusion of project.

DOI: 10.48175/IJARSCT-24850

II. PROBLEM DESCRIPTION

A significant amount of effort is required to extract the desired information from a vast repository of data in an information system. Natural Language Processing (NLP) is a method that allows user queries, entered in natural language (such as English), to be converted into SQL queries based on the input. Since an average person is not expected to know SQL, this system simplifies the process by enabling them to generate SQL queries effortlessly, making information retrieval easier, as databases only understand SQL

.



.Figure 1: Problem Description

Consider a database, referred to as DB. Within this DB, we have various tables with well-defined attributes, all properly normalized. If a user wants to access data from these tables, they would traditionally need to be proficient in SQL to query the database. Our system removes this technical requirement and allows users to interact with the database using their own language.

For example, if a user wants to view information about a particular student from the "Student" table, they would typically use the following SQL query:

SELECT * FROM employee WHERE clg_name="XYZ";

However, a person who doesn't know SQL will find it difficult to retrieve this data. But with the assistance of NLP, retrieval from the database gets simpler. This query would be restated in NLP as:"Give the information of the employee who works in the college."

Both the SQL query and the NLP query would produce the same result. The main difference is that with NLP, someone who is not familiar with SQL can easily get the data out of the database.

III. LITERATURE SURVEY

One proposed solution to automate natural language query conversion into SQL queries was described as Natural Language to SQL Query Conversion.[1]The official language to operate and handle data within relational databases relies exclusively on SQL. Users obtain updated data by supplying SQL queries containing correct structure and syntax elements. The Natural Language Sequential Query Language (NLSQL) framework automatically generates SQL queries through natural language processing (NLP) technology to handle such problems.

SQL text conversion technologies transform user questions into formal databases through automated processes thereby enabling untrained database users to interact with them. The extensive search area in neural semantic parsers (NSPs) leads to difficulties converting extensive and complex queries into nested SQL queries. [11]You can consider natural language query response over tables to be a semantic parsing problem but finding high-quality supervision to train semantic parsers remains challenging. The parsers produce logical forms as an intermediate stage which leads to complete meaning extraction..

Using Natural Language Processing to Generate SQL Queries Dynamically Processing Natural Languages (NLP) is now being increasingly used in different areas of human-machine communication. Though database queries consist of limited instructions, they might be complicated. Therefore, Writing, testing, optimising, and running SQL statements typically requires the assistance of qualified human resources. This procedure can be challenging, time-consuming, and occasionally prone to mistakes.[3] These challenges can be overcome by dynamically constructing queries with standardized processes.

Natural language text to SQL query conversion with generative AI brings in Semantic Synthesis, a new method that translates natural English text into correct, context-sensitive SQL queries. Employing sophisticated Generative AI methods, it fills the gap between user intent and SQL complexity, making database access easy for individuals without SQL knowledge. The system, with a Large Language Model (LLM) enhanced by transformers and an intuitive interface, produces SQL queries for the user-defined task and offers a clear, algorithmic explanation of the query formation process.

Conversion from They must study Structured Query Language (SQL), as many people may not be familiar with the database structure. [6] Consequently, machine learning-based natural language queries to SQL for non-experts Natural Language Processing (NLP) is a branch of artificial intelligence and computer science that deals with processing natural language data and interacting with computers using natural language. [5] In the case of average users, application of Structured Query Language (SQL) becomes difficult for them, and this is further compounded by their unfamiliarity with the database's structure. To increase interaction between databases and users, the smarter new interface is needed.

An algorithm for converting plain language into relational database SQL queries. To improve the communication between databases and users there is an increasing demand for intelligent interfaces in database applications. Not everyone is They must study Structured Query Language (SQL), as many people may not be familiar with the database structure. [6] Therefore, a system that allows non-expert users to communicate with databases in natural language, such as English, is necessary. A Database Management System (DBMS) must be able to understand Natural Language (NL) in order for this to happen.

Architectur for Converting Natural Language to SQL Queries . Natural language (NL) enquiries can be automatically transformed into structured query language (SQL) queries (NL2SQL). significantly enhance database accessibility for users. This approach outlines a standardized architecture for converting natural language into SQL queries, providing a clear framework for querying structured data.

An Interactive Approach to Generating SQL Queries from Natural LanguageIn this thesis, we present AI SQL Assistant, an natural-language driven interactive interface for SQL query generation. Inspired by a lack of usability of the current systems, sql was designed with a goal of employing it for complicated query generation. The interface provides users with a natural language and mathematical operations having a minimal structure. We tested sql with a five-subject first-use study, in which subjects were asked to create queries from the TPC-H decision support benchmark. [8]

Conversion of Automating the process of converting Natural Language enquiries into Structured Query Language (SQL) queries. [9] For relational database management systems, SQL is a standard tool. To retrieve or modify data from such databases, the right SQL query with the right keywords and syntax must be entered. In order to overcome this issue and translate natural language enquiries into SQL queries, we proposed Natural Language Sequential Query Language, or "NLSQL," a system in natural language processing.

By allowing anyone to examine the RDBMS by posing queries, text-to-SQ solutions removes the need for structured languages like SQL. Neural semantic parsers (NSPs) often struggle to translate long and complex utterances into the nested Structured query language (SQL) queries because of the large search area. Semantic parsing tasks are generally considered to be natural language query replies over tables. Since the logical forms generated are only used as a step before recovering the connotation, it is challenging to train semantic parsers from inadequate supervision. [10] efficiently automate the process of converting natural language queries to SQL queries. [4] Query Structure One of the most effective tools for managing data in a relational database management system is language. The user must enter the correct SQL query in order to retrieve or manage data. However, the necessary data cannot be retrieved by people who are not familiar with SQL. To get around this, we suggested a Natural Language Processing model that would translate a Natural Language query into a SQL query. This makes it easier for inexperienced users to obtain the necessary content without having to understand intricate SQL specifics. Complex questions can also be handled by this system.

IV. PROPOSEDMETHODOLOGY

In today's digital world, data is growing at an incredible rate. While there are many advanced database tools available to store massive amounts of data, not everyone knows how to interact with databases effectively. Many people struggle with writing queries to retrieve the exact information they need. To solve this problem, we are developing a system that converts natural language questions into SQL queries. This way, users can easily access data from a database without needing prior knowledge of database management.

Our system works in multiple stages:

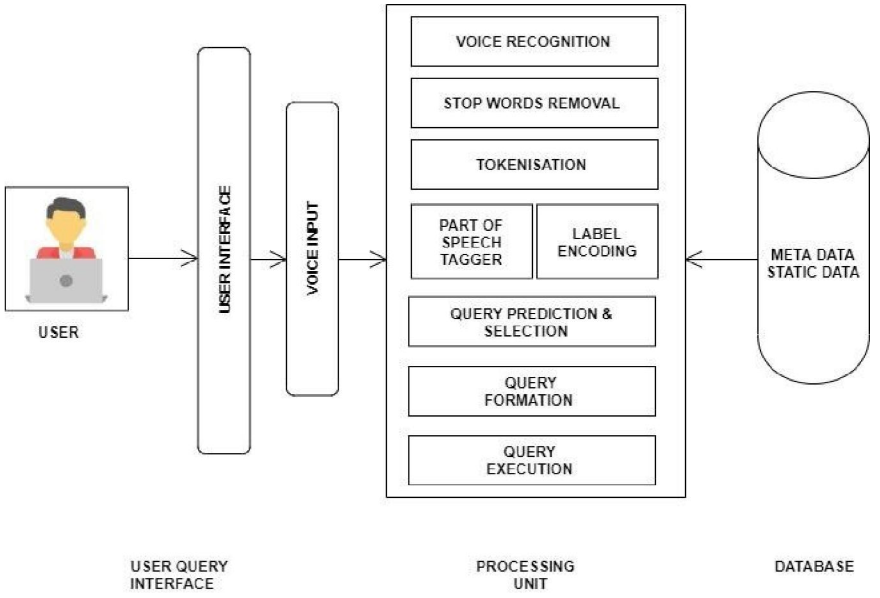
1. Voice Recognition: The user speaks their query, and the system converts their voice into text using speech recognition technology.
2. Text Processing: Once the voice input is converted into text, we refine it through several steps:

Tokenization: The sentence is broken down into individual words (tokens) and stored in a list.
Example: "Show me all the students from B.E."
This is converted into: ["Show", "me", "all", "the", "students", "from", "B.E"]
Stop Words Removal: Words like "me," "the," "from," etc., which don't contribute to the main query, are removed.
After removing stop words, the refined query becomes: ["Show", "students", "B.E"]
Parts of Speech (POS) Tagging: Each word is analyzed and labeled (noun, verb, adjective, etc.) to better understand the query's structure and intent.

3. Query Type Prediction: Using the Multinomial Logistic Regression Algorithm, the system predicts the type of SQL query required based on the processed text.
4. Query Validation & Execution: The system verifies the predicted query and then generates an appropriate SQL statement.
5. Fetching & Displaying Data :The system retrieves relevant data from the database and presents it to the user in an easy-to-read format.This system bridges the gap between users and databases by allowing them to retrieve information using simple voice commands—no need to learn SQL![7]

Multinomial Logistic Regression Algorithm: Multinomial Logistic Regression is an extended version of the standard logistic regression algorithm, designed for multi-class classification tasks. When logistic regression is applied to problems with more than two possible outcomes, it is referred to as Multinomial Logistic Regression.

V. ARCHITECTURE



VI. ALGORITHM

"Multinomial Logistic Regression is an extension of the logistic regression algorithm that is used for problems where there are more than two possible categories or classes. While regular logistic regression is typically used for binary classification (deciding between two classes), multinomial logistic regression adapts this concept to handle multiple classes. Essentially, it's the same logistic regression technique, but applied to situations where you're trying to predict from more than two possible outcomes."

VII. SYSTEM RESULT

System Results and Interface Overview:

In this section, we showcase the results of our proposed system. The system allows users to ask the same question in different ways using various query verbs such as Show, Find, Tell, Search, Give, List, and Display. Additionally, users can enter a natural language question without explicitly using a query verb. The interface, as shown in Figure 3, demonstrates how a natural language question (NLQ) like "Show all books" is translated into an SQL query.

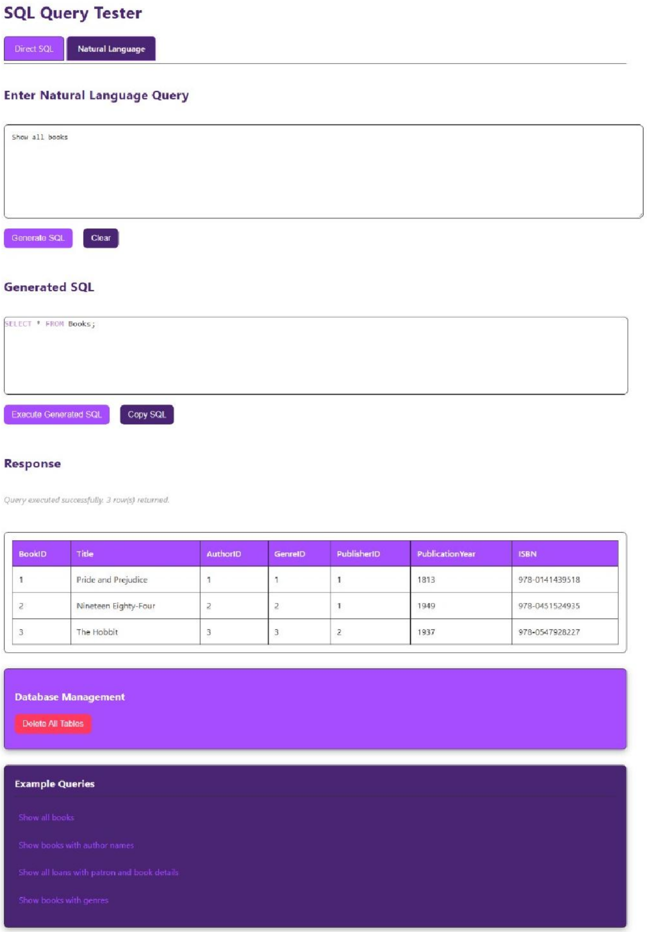
The interface consists of two textboxes:

1. The first textbox is where the user enters their natural language query.
2. The second textbox displays the corresponding SQL query generated by the system.

There are also two buttons:"Direct SQL" Button – Allows users to input SQL queries directly.

"Natural Language" Button – Converts the user's NLQ into SQL and retrieves the results.

Once the query is processed, the system displays the results in a clear tabular format, making it easy for users to interpret the data.



Show all books

```
SELECT * FROM Books;
```

Query executed successfully. 3 row(s) returned.

BookID	Title	AuthorID	GenreID	PublisherID	PublicationYear	ISBN
1	Pride and Prejudice	1	1	1	1813	978-0141439518
2	Nineteen Eighty-Four	2	2	1	1949	978-0451524935
3	The Hobbit	3	3	2	1937	978-0547928227

Fig 3. User interface

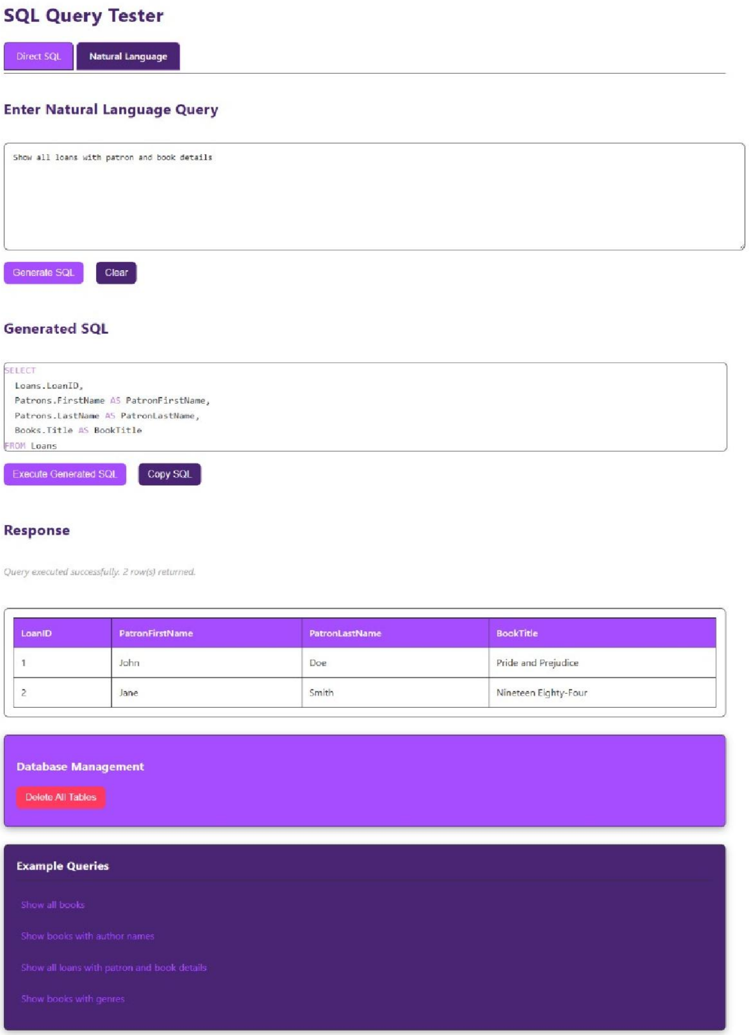
DOI: 10.48175/IJARSCT-24850

In the fig4 the user give a NLQ that is I need to design a relational database for an e-commerce systen. The database should include three tables: Products, Customers, and Orders. The tables should be related to each other to ensure data integrity and efficient querying the system will convert this natural language in to SQL query the SQL query shown below

CREATE TABLE Products (ProductID INTEGER PRIMARY KEY, ProductName TEXT, Price REAL, Description TEXT);

CREATE TABLE Customers (CustomerID INTEGER PRIMARY KEY, FirstName TEXT, LastName TEXT, Address TEXT, Email TEXT, Phone Number TEXT);

CREATE TABLE Orders (OrderID INTEGER PRIMARY KEY, CustomerID INTEGER, OrderDate DATE, TotalAmount REAL, FOREIGN KEY (CustomerID



Show all loans with patron and book details

```
SELECT
  Loans.LoanID,
  Patrons.FirstName AS PatronFirstName,
  Patrons.LastName AS PatronLastName,
  Books.Title AS BookTitle
FROM Loans
```

Query executed successfully. 2 row(s) returned.

LoanID	PatronFirstName	PatronLastName	BookTitle
1	John	Doe	Pride and Prejudice
2	Jane	Smith	Nineteen Eighty-Four

Fig 4. NQL to SQL Query

We're building an app that's super easy to use, even if you've never written a line of SQL before. Our goal is to make interacting with databases as simple as having a conversation. With our system, all you have to do is type a question or request in plain, everyday language, and we take care of the rest.When you ask a question, our app figures out what you mean and automatically turns it into an SQL query. This way, you don't need to know anything about SQL syntax to

DOI: 10.48175/IJARSCT-24850

get the results you need. Whether you're looking to find specific data, filter through records, or even calculate totals, the app handles all the technical stuff for you.In short, you can easily manage and retrieve data just by asking, and our system does all the heavy lifting behind the scenes. No more worrying about complicated queries—just simple, straightforward answers.

VIII. FUTURE SCOPE

The developed web application serves as a system which transforms verbal commands into SQL database requests. The system currently executes basic and advanced query commands with efficiency and we will enhance its features by integrating multiple query language processing in upcoming development.. This will allow users to type instructions in natural language and produce code in different languages, switching between them as easily as checking a box on a form.

Another intriguing option is to incorporate this AI system into mainstream code editors like VS Code, Atom, and Sublime Text as an extension. After installation, users will be able to code by voice, with some nice extras like auto-suggestions and autocomplete. In addition, we intend to introduce a code conversion feature where users can convert code between different programming languages with ease. This would improve developers' productivity by making coding easier and more efficient.

IX. CONCLUSION

In summary, the incorporation of Natural Language Processing (NLP) into database management systems provides a revolutionary solution to enhancing usability and accessibility. By allowing users to communicate with databases through natural language queries, our system closes the gap between non-technical users and sophisticated data operations. The NLP-based system facilitates the process of database retrieval by interpreting and translating natural language inputs into exact SQL queries, thereby reducing the barrier to entry into database management. While the existing system accommodates both primitive and certain advanced queries, further development to include more SQL structures will strengthen its capabilities. With continuing updates to the system's data dictionary and continuing extensions of query support, the system is hugely promising in giving people with minimal to no SQL skills the ability to effectively deal with and manage databases. Ultimately, the solution showcases the way that AI can transform and make technology more straightforward, intuitive, and accessible for everyone.

REFERENCES

[1] Singh G, Solanki A (2016) An algorithm to transform natural language into SQL queries for relational databases. Selforganizology 3(3):100–116

[2] Gkini , O. , Belmpas , T. , Koutrika , G. , Ioannidisd , Y. , An in-depth benchmarking of text-to-SQL systems . ´SIGMOD '21 , Virtual Event, China , June 20–25, 2021Frank Meng and Wesley W. Chu, "Database Query Formation from Natural Language using Semantic Modelling and Statistical Keyword Meaning Disambiguation", 1999.

[3] 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), pp. 1-5, 2019.

[4] A. Mohite and V. Bhojane, "Natural language interface to database using modified co-occurrence matrix technique," 2015 International Conference on Pervasive Computing (ICPC), Pune, 2015, pp. 1-4.

[5] T. Anisha, P. Rafeeque and R. Murali, "Text to sql query conversion using deep learning: A comparative analysis", proceedings of the International Conference on Systems Energy Environment (ICSEE) 2019, 2019.

[6] Wan FJ. 2000. A fuzzy grammar and possibility theory – based natural language user interface for spatial queries. Fuzzy Sets and Systems, 113: 147-159

[7] Sun, Ruoxi, Sercan O. Arik, Hootan Nakhost, Hanjun Dai, Rajarishi Sinha, Pengcheng Yin, and Tomas Pfister. "SQL-PaLM: Improved Large Language Model Adaptation for Text-to-SQL." arXiv preprint arXiv:2306.00739 (2023).

[8] Satav AG, Ausekar AB, Bihani RM, Shaikh A. A Proposed Natural Language Query Processing System. International Journal of Science and Applied Information Technology. 2014; 3(2).

[9] Finegan-Dollak C et al (2018) Improving text-to-SQL evaluation methodolar Xiv preprint arXiv:1806.09029

DOI: 10.48175/IJARSCT-24850

[10] Uma , M. , Sneha , V. , Sneha , G. , Bhuvana , J. , Bharathi , B. , Formation of SQL from natural language query using NLP . Second International Conference on Computational Intelligence in Data Science (ICCIDS) , 2019 .

[11] Debarati Ghosal, Tejas Waghmare, Vivek Satam and Chinmay Hajirnis, "SQL query formation using natural language processing", International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 3, March 2016.

DOI: 10.48175/IJARSCT-24850