International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Voice-Driven Natural Language to SQL Query Conversion Using NLP Techniques

P. Shashank Reddy[1] , P.Dhanush Reddy[2] , O.Shiva ganesh[3], Research Scholars

Mrs.B.Vasundhara Devi Varanasi[4], Assistant Professor

Department of Computer Science and Engineering,

Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad, Telangana 501301

Abstract : ***To get the accurate SQL queries in this application, input is given in the form of natural language, and using NLP techniques, relevant information is extracted to get SQL queries. This application is implemented for non-technical users, which is more user-friendly and has the capability to generate SQL code for complex queries and language with ambiguity. So, this application can be used by the individuals who have limited technical skills and who can enter the queries in the natural language. Even this application helps to generate the SQL code for those who do not have knowledge of SQL code. As per the user requirements, the user input is collected in the format of natural language and translated into SQL queries using algorithms from NLP algorithms. This application helps professionals in the business domain to save their efforts as well as time and analyze huge amounts of data by generating SQL queries automatically, as this application provides more accurate results for SQL query generation. Python libraries such as NLTK and SpaCy are used in this application programming.***

Keywords: SQL queries, natural language, natural language processing, Python programming, NLTK, Spacy libraries.

I. Introduction

Accessing relational databases through natural language presents developers with a major and complex technical barrier. A proper model needs to interpret natural language while it transforms the information into structured relational database queries for execution. Natural Language Interface to Database frameworks (NLIDB) came into existence through the adoption of natural language instead of SQL. NLIDB advances the development of intelligent database systems (IDBS) since it enhances database flexibility through better user experiences [9]. Presented as a model the technology produces database queries out of natural language inputs. Researchers have reached an exceptional 80% success rate in SQL query translation from natural language through neural algorithms working with public database collections [1].

We will evaluate current frameworks according to their incorporation of aggregation classifiers along with select column pointers and clause pointers within this paper. We will analyze the semantic parsing mechanism as well as neural algorithm operations involved in predicting aggregation and both column pointer and clause pointer functionality which constitute the crucial components of current solutions. Both the LUNAR system and the LADDER system were developed in the 1970s to allow non-technical users ask questions about moon rock samples and US Navy ships respectively through natural language inputs (Woods, 1972) [2].

The rapid growth of software and hardware technology during the last five decades made pre-1970 database systems incompatible with contemporary definition of databases (Bercich 2003; Frank 2018). The rise of industry needs triggered the appearance of several frameworks that transform natural language into database query requirements since the initial release. Our analysis of historic system developments led us to recognize research directions which focus on domain retrieval time and portability limitations and complicated configuration requirements [2].

The process of converting natural language inquiries to the database query languages SQL, Simple Protocol, and SPARQL requires complex procedures. The challenge becomes more significant because of the large scale and complicated database technologies that databases developers use today (Nadkarni, 2011) [3]. The current storage engines manage data through various data structure types including tabular format and NoSQL and textual graph format which demands separate query languages for data retrieval [4]. Multiple data formats found in translation work create additional problems during processing. The advancement of machine learning techniques enabled developers to create multiple frameworks which effectively translate natural language queries.

Research demonstrates that recurrent neural networks (RNNs) achieve better semantic parsing outcomes when used together with attention and copying mechanisms. Research using Seq2SQL demonstrates that splitting query decoding operations into target column, aggregation units and predicate evaluation functions produces better prediction outcomes. The model succeeds at recognizing semantically connected queries and this capabilities enable superior outcomes. Our research analyzes the existing frameworks through implementation-

based subcategories for evaluation of their resulting performance levels. The systems SQL-Net, Syntax SQL-Net, Grammar SQL, IR-Net, Edit SQL and RATSQL have unique performance metrics and operational algorithms that distinguish them from each other.

This section explores natural language to SQL translation models before offering an overview of spoken SQL query conversion solutions along with their model descriptions. We will present the DIY model in the discussion part while examining both its strengths and weaknesses for practical implementation. We will explore possible future developments of our work and improvements for the existing problem in our conclusion. The accuracy problem in these systems could possibly be addressed through ontological acyclic directed graphs which serve as an extended solution.

II. Literature Survey

Structured data in relational databases requires SQL queries for users to access the stored information. These users experience difficulties accessing the system because of their lack of technical skill. The development of natural language interfaces to database frameworks called NLIDBs enables users to interact with databases through natural language instead of SQL query systems. NLIDB brings forward the development of more intelligent database systems by enabling flexible and open database access. The implementation of sophisticated neural algorithms succeeds in translating natural language statements to SQL queries, which demonstrate an 80% success rate in deriving SQL commands from publicly accessible datasets. This paper evaluates different existing NLIDB systems while analyzing their operational performance regarding aggregation classifiers alongside column pointers and clause pointers. The paper examines how semantic parsing together with neural algorithms works to enhance the accuracy of queries. Analysis of speech to SQL application models begins with an assessment of their current capabilities as well as their weaknesses and potential development opportunities to generate better systems. [1]

Natural Language Queries (NLQ) transformation into SQL Queries with effective execution stands as the core subject of this research study. Relational databases are managed and data is retrieved using Structured Query Language (SQL) as this database management system challenges users who lack SQL expertise to obtain necessary data. The proposed model lets unskilled users including Training and Placement cell officers obtain database information from their student repositories without requiring SQL expertise. The system accepts sophisticated inquiries and functions with spoken requests that the program transforms into written text. The text query goes through an SQL conversion process before system execution leaving the user with an output result that redefines non-programmer database usage.[2]

The intelligent interface serves as an access point which enables users to operate databases without SQL expertise. The lack of understanding about database structures and query authoring among users makes the access simpler because natural language functions like English work. The developed system establishes a user-friendly interface which transforms natural statements into SQL statements through semantic matching alongside data dictionary usage. The system processes each component of user input following a series of steps while displaying the needed outcome to the user. [3]

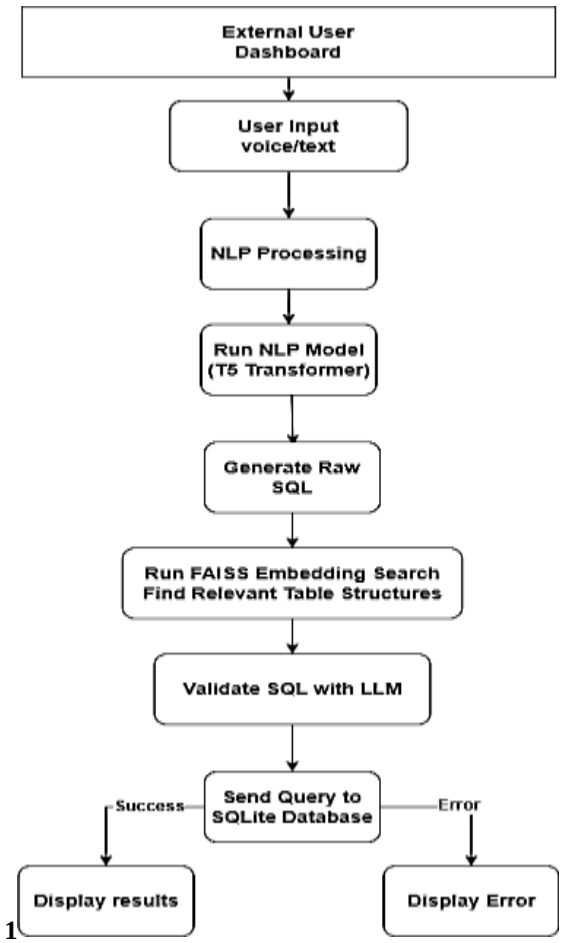
The system development process concentrates on constructing software which transforms end-user English statements into SQL code. The system proves beneficial because it enables users without SQL coding skills to work with database content. The system builds a sequence of intermediate queries through user inputs that lets users pick the most appropriate choice to create the final SQL statement. The system provides a recommendation function that allows users to rectify mistakes and maintain their database search operations steadily. [4]

The growing trend of basing decisions on data requires convenient access to database resources. The system uses natural language design as an effective tool to enable people from both technical and non-technical backgrounds to work with database information through simplified input methods. Research on deep learning models for this task continues despite limitations in available datasets and model assumptions preventing the creation of a single solution. This paper analyzes 24 neural network models along with 11 common datasets which support TEXT2SQL research giving details about design approaches and existing challenges and predicted applications. [5]

Natural Language Processing (NLP) enables machines to interpret spoken language through processing interactions with human dialogue. A user-friendly interface which enables non-technical personnel to ask queries in natural language remains necessary because these users face difficulties when working with SQL and database systems. The progression of Natural Language Interface to Database (NLIDB) systems became possible due to the technological requirement. The system uses tokenization and machine learning techniques to convert linguistic user requests into proper SQL statements through mechanisms such as word similarity checks and PoS tagging and Naive Bayes and Jaro-Winkler matching algorithms for precise results. [6]

III. Proposed Method

Proposed method uses fine tuned T-5 Transformer for conversion of natural language to SQL code. Proposed method block diagram is shown below,



1

Fig. Block diagram of proposed method

Proposed method includes following stages as,

a) User Input Text or Voice
b) NLP Processing
c) Run T5 Transformer
d) Generate raw SQL
e) Run FAISS embedding search and find relavant table structure
f) Validate SQL with LLM
g) Send query to SQLite database
h) Display results of success
i) Create a framework using Flask

In the below paragraphs, the proposed method is explained in detail, which demonstrate the flow of the process and pipeline of data processing.

1. User Input (query in natural language)

Both type of inputs voice and text is accepted by the proposed method. Natural language is used for input query.

2. NLP Processing

NLP techniques such as tokenization and data cleaning are used in this application to get data in proper format.

3. Run T5 Transformer

The fine tuned transformer based model T5 transformer is used for proposed method. It will take preprocessed natural language and provide SQL queries.

4. Generate Raw SQL

Obtained SQL at this stage may not be perfect but attempts to understand the user interaction related to database.

5. FAISS Embedding Search

FAISS is used for indexing and embedding columns and tables which are part of schema elements.

6. LLM for Validation of SQL

Large language models helps in getting matched schema with better accuracy from raw SQL data obtained. Semantically and syntactically SQL schema is correct or no verified.

7. SQLite Database Query check

The obtained SQL code is pass through SQLite database. If the matched data is obtained then it will retrives the data else other operations related to database are performed.

8. Display results if correct

Success message and table of data is shown once SQL is generated.

9. Flask framework for results demonstration

Flask is the web based framework which is used for interactive user interaction with we application.

IV. Results Analysis

Proposed model natural language to SQL code using T5 transformer which is fine tuned for better performance. Database search similarity FAISS is used in the proposed application. Flask framework has been used for UI with better interface which can be easily accessible to user. Results of the proposed system are shown as below ,

Figure-4.1 Image of HomePage

Flask framework is used to prepare the web application and above page is the home page for proposed application. Home page is prepared using HTML.

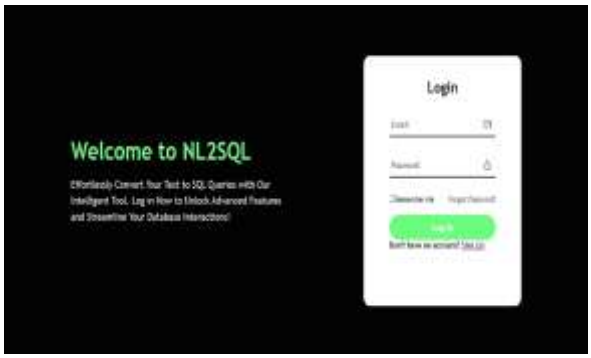


Figure-4.2 Image of Login Page

First user need to go for signup. Once user is signup then user can go for login to use proposed application. User has to enter correct credentials to get access to the application page.

Figure-4.3 Image of Dashboard

After successful login to the application user will get the above HTML page in which user can enter a SQL query or Natural Query for further conversion.

Figure-4.4 Image of Query Execution

In above , user entered the query and query got executed. It is shown that 'Query Executed Successfully'. Proposed transformer model is used for this conversion.

Figure -4.5 Image of Output of Query Execution

In above figure it is observed that user entered a natural language as 'Show all the data in teachers database' and below we can see the query results in tabular format.

Figure- 4.6 Image of Error in Query

In above user has entered 'show all the data in hello database' but as there is no such database our application will reply with 'Unable to render your input. Please reframe elaborately'.

V. Conclusion

The NL to SQL conversion project makes it easy for users to interact with databases using simple, everyday language instead of writing SQL. It uses a T5-based transformer model to translate natural language into accurate SQL queries. The system smartly understands the database's structure using real-time schema extraction and embeddings, adapting to any changes. Features like query history and secure session management improve usability and safety. With optimizations like GPU acceleration and faiss-based schema matching, it handles large datasets efficiently. This tool is helpful in business, education, and healthcare, showing how NLP can simplify complex data tasks.

References

1. Baig, Muhammad Shahzaib, Azhar Imran, Aman Ullah Yasin, Abdul Haleem Butt, and Muhammad Imran Khan. "Natural language to sql queries: A review." *International Journal of Innovations in Science Technology* 4 (2022): 147-162.
2. Kate, Abhilasha, Satish Kamble, Aishwarya Bodkhe, and Mrunal Joshi. "Conversion of natural language query to SQL query." In *2018 second international conference on electronics, communication and aerospace technology (ICECA)*, pp. 488-491. IEEE, 2018.
3. Singh, Garima, and Arun Solanki. "An algorithm to transform natural language into SQL queries for relational databases." *Selforganizology* 3, no. 3 (2016): 100-116.
4. Bhadgale, Anil M., Sanhita R. Gavas, P. R. Goyal, and M. M. Patil. "Natural language to SQL conversion system." *International Journal of Computer Science Engineering and Information Technology Research* 3, no. 2 (2013): 161-166.
5. Kumar, Ayush, Parth Nagarkar, Prabhav Nalhe, and Sanjeev Vijayakumar. "Deep learning driven natural languages text to SQL query conversion: a survey." *arXiv preprint arXiv:2208.04415* (2022).
6. Arefin, Minhazul, Kazi Mojammel Hossen, and Mohammed Nasir Uddin. "Natural language query to SQL conversion using machine learning approach." In *2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1-6. IEEE, 2021.
7. Arefin, Minhazul, Kazi Mojammel Hossen, and Mohammed Nasir Uddin. "Natural language query to SQL conversion using machine learning approach." In 2021 3rd International Conference on Sustainable Technologies for Industry 4.0 (STI), pp. 1-6. IEEE, 2021.
8. Brunner, Ursin, and Kurt Stockinger. "ValueNet: A Natural Language-to-SQL System that Learns from Database Information." arXiv preprint arXiv:2006.00888, 2020. [Online]. Available: https://arxiv.org/abs/2006.00888
9. Wang, Tianshu, et al. "DBCopilot: Scaling Natural Language Querying to Massive Databases." arXiv preprint arXiv:2312.03463, 2023. [Online]. Available: https://arxiv.org/abs/2312.03463
10. Rebei, Amine. "Fine-Tuning Language Models for Context-Specific SQL Query Generation." arXiv preprint arXiv:2312.02251, 2023. [Online]. Available: https://arxiv.org/abs/2312.02251
11. Fan, Yuankai, et al. "Metasql: A Generate-then-Rank Framework for Natural Language to SQL Translation." arXiv preprint arXiv:2402.17144, 2024. [Online]. Available: https://arxiv.org/abs/2402.17144
12. Shi, David. "Text2SQL: Converting Natural Language to SQL." Medium, 2023. [Online]. Available: https://medium.com/@changjiang.shi/text2sql-converting-natural-language-to-sql-defa12c2a69f
13. "Build a robust text-to-SQL solution generating complex queries, self-correcting, and querying diverse data sources." AWS Machine Learning Blog, 2023. [Online]. Available: https://aws.amazon.com/blogs/machine-learning/build-a-robust-text-to-sql-solution-generating-complex-queries-self-correcting-and-querying-diverse-data-sources/
14. "Use natural language to execute SQL queries." Microsoft Semantic Kernel Blog, 2023. [Online]. Available: https://devblogs.microsoft.com/semantic-kernel/use-natural-language-to-execute-sql-queries/