JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Speak2SQL:Natural Language to SQL Conversion

[1]Mandara R, [2]Dr. K R Sumana
[1]PG Student, The National Institute of Engineering, Mysuru, Visveswaraya Technological University, Belagavi, Karnataka, India
[2]Faculty, The National Institute of Engineering, Mysuru, Visveswaraya Technological University, Belagavi, Karnataka, India

Abstract*:* Speak2SQL is a natural language to SQL conversion system designed to simplify database access for users without technical expertise in query languages. The system leverages Retrieval-Augmented Generation (RAG) with LangChain, FAISS vector search, and Hugging Face models to translate user questions into accurate SQL statements. It features a dual-pipeline approach—supporting both schema-aware query generation for predefined datasets like WikiSQL and real-time execution for custom relational databases. By combining semantic search for relevant schema information with prompt-driven query generation, Speak2SQL enhances accuracy, relevance, and ease of use. The system also includes a user-friendly interface built with Streamlit, offering dropdown-based query suggestions, schema display, and result visualization. This project demonstrates the potential of integrating modern NLP techniques with database systems to create intuitive, scalable, and accessible solutions for data retrieval.

Index Terms – Natural Language Processing (NLP), Text-to-SQL, Retrieval-Augmented Generation (RAG), LangChain, Hugging Face Models.

I. Introduction

In a world where data drives decision-making, timely and efficient access to information has become a fundamental requirement for businesses, research institutions, and government organizations. Relational databases store vast volumes of structured data, and SQL (Structured Query Language) has remained the primary language for querying and managing this data. While SQL is extremely powerful, it requires both an understanding of database structure and knowledge of its syntax, which creates a significant learning curve for nontechnical users. Many domain specialists—such as market analysts, healthcare professionals, financial planners, and educators frequently need access to specific datasets but lack the skills to craft precise SQL queries.

II. LITERATURE SURVEY

Mengfei et al. **[1]** examine how closing the semantic gap between natural language questions and database tables can improve Text-toSQL translation. Ashwin K. Sharma et al.**[2]** conduct a comparative evaluation of open-source Large Language Models for NL-to-SQL translation across three benchmark datasets. Zenan Ning et al.**[3]** review Text-to-SQL question answering technologies, covering dataset evolution, model design, and application scenarios. Guang-Yu Cheng et al.**[4]** present a method combining data augmentation and candidate re-ranking to enhance Text-to-SQL performance. The authors of Querylizer. introduce[5] an interactive database design and text-to-SQL conversion platform. The QuerySpeak project**[6]** suggests a method for effectively retrieving data that converts human language into SQL queries. Bridging the Semantic Gap:**[7]** Techniques for maintaining semantic meaning during NL-to-SQL translation are covered in Automated SQL Query Generation from Natural Language 19. Investigations into Natural Language Queries. Querying**[8]** for Remote Sensing Databases presents a specialized system for querying remote sensing datasets using natural language. The paper on Transformer-based Model**[9]** for the Natural Language Textual Sequences to SQL-queries Conversion investigates the application of transformer architectures for direct NL-to-SQL generation. The SLENet study introduces a specialized neural [10] For Text-to-SQL tasks, the SLENet study presents a specialized neural network architecture that incorporates layered encoding and schemalinking enhancements . SLENet enhances the model's capacity to produce precise SQL queries by enhancing the alignment between natural language elements and database schema components. Natural Language[11] driven by LLM to SQL for Utility By incorporating large language model capabilities into dashboard systems. The problem of translating text to SQL for the Turkish language is tackled by TUR2SQL **[12]**, which emphasizes linguistic complexities like rich morphology and flexible syntax. NL2SQL**[13]** for Multilingual and Cross-Domain Applications presents a system designed to handle multiple languages and cross-domain database schemas.

III. PROPOSED WORK

The Speak2SQL system will allow users to speak or type natural language queries, retrieve relevant schema/context from databases, and generate accurate SQL queries using an LLM enhanced with Retrieval-Augmented Generation (RAG).It will support WikiSQL for benchmark testing and a custom database for real-world scenarios.

IV. METHODOLOGY

4.1 Data Collection

WikiSQL is a large-scale dataset featuring around 80,000 examples of natural language questions paired with SQL queries. It draws from various Wikipedia table schemas, ensuring diverse query formats and vocabulary. Widely used in NLP research, this dataset evaluates Natural Language to SQL (NL2SQL) models across tasks like selection, aggregation, and filtering. A unique database was created and filled with domain-specific tables in order to evaluate the system's performance under practical conditions. A common relational schema in operational or business settings is represented by this dataset.

4.2 Data Preprocessing

Before training and deploying the **Speak2SQL** pipelines, both datasets—the WikiSQL benchmark as well as the customized MySQL database need methodical preprocessing. This stage guarantees that the data is consistent, clean, and compatible with the NLP models that are used to convert natural language to SQL. Additionally, preprocessing is essential for decreasing model errors and increasing the accuracy of query interpretation.

4.3 Schema Extraction and Representation

The model needs to comprehend the structure of the database it is querying to generate SQL with accuracy. In this step, the database schema, table names, and column names are used to extract the data types. After that, they are arranged in a systematic manner. and stored for quick access while a query is running in a vector database with FAISS. Although the WikiSQL dataset already contains schema information, it has been reformatted to satisfy the system's input specifications. The schema for the custom database is dynamically extracted using MySQL queries.

4.4 Model Selection and Training

The system makes use of LangChain. The main engine for deciphering natural language and producing SQL queries is the LLaMA3 language model, which can be accessed through Groq's high-performance inference API. Because of its strong grasp of syntax and semantics as well as its capacity for handling intricate, multi-step reasoning, LLaMA3 was selected.

Furthermore, FAISS-based vector search is incorporated to effectively retrieve pertinent schema context. This increases accuracy, particularly in multi-table or complex query scenarios, by guaranteeing that the model has access to accurate database details prior to producing a SQL statement**.** Benchmark evaluations on **WikiSQL** provide a standardized accuracy measure, while testing on the **Custom MySQL database** validates real-world performance. The combination of these assessments confirms the chosen model's effectiveness across different contexts.

4.5 Data Augmentation

Data augmentation in the context of Natural Language to SQL (NL2SQL) systems focuses on diversifying the natural language queries associated with each database operation. Since real-world users can phrase the same request in multiple ways, expanding the training set with varied expressions improves the model's adaptability and robustness.For the WikiSQL dataset, augmentation involves paraphrasing existing questions using synonym replacement, reordering phrases, and introducing alternative query structures without altering the underlying SQL intent. For instance, a question like *"Show me the names of employees in the Sales department"* might be rephrased as *"List all employees working in Sales"*. This variation helps the model generalize better to unseen queries. In the Custom MySQL database setup, augmentation also includes domain-specific terminology. Here, questions are rewritten using business-relevant synonyms or abbreviations that real users might employ. The augmented queries are validated against the schema to ensure they still map to valid SQL statements.

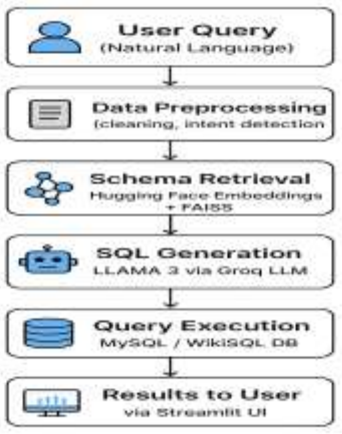


Figure 1 System implementation

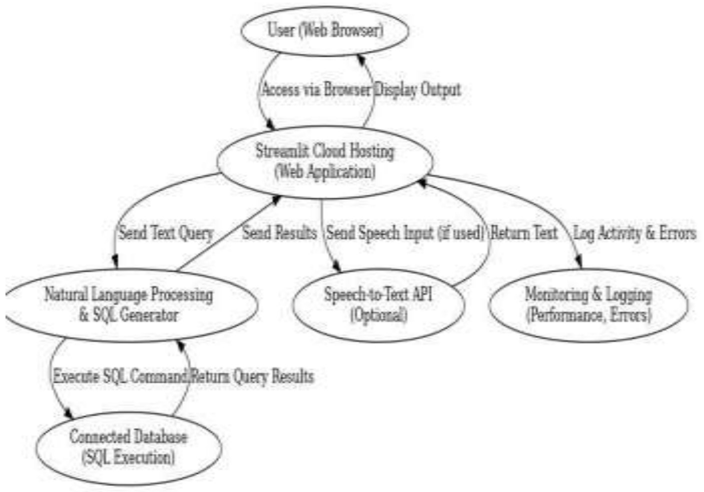


Figure 2 Speak2SQL deployment architecture

The Fig(1) shows This diagram shows the workflow of a **Natural Language to SQL system**, starting from user query preprocessing to schema retrieval, SQL generation, execution, and displaying results.It highlights the integration of **Hugging Face embeddings, FAISS, Groq LLM (LLaMA 3), and Streamlit UI** in the pipeline.

The deployment phase of Speak2SQL Fig(2) focused on making the system accessible to end users within a reliable, secure, and scalable environment. After thorough testing during this phase, the web application was prepared for deployment using Streamlit's cloud hosting service, ensuring users could access the platform via a standard web browser without any local installations.

V. RESULT ANALYSIS

A correlation map (Figure 2(a)) was constructed using the numeric attributes extracted from the WikiSQL dataset, including factors such as query length, number of columns referenced, and table size. The visualization showed that query length has a weak to moderate positive correlation with the number of columns, suggesting that longer queries typically involve a greater number of attributes from the database schema. This observation highlights that query complexity is partially driven by schema richness. However, the analysis did not uncover any strong linear relationships among the remaining variables. For example, table size and query length appeared largely independent, indicating that larger tables do not necessarily result in longer or more complicated SQL queries. This pattern underlines the heterogeneous nature of the WikiSQL dataset, which spans multiple schemas and query types, thereby limiting the emergence of consistent correlations. Overall, the findings emphasize the importance of schema structure over dataset size in influencing query formulation.

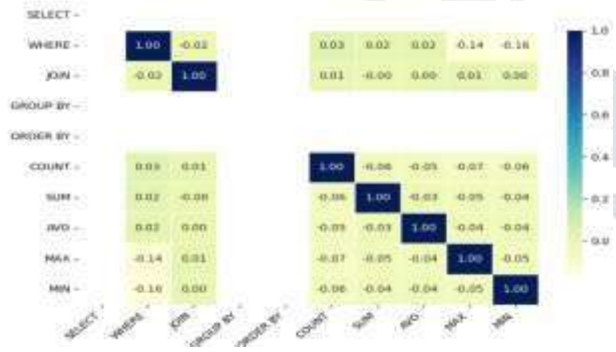


Figure 2(a) – **Correlation Map for WikiSQL**.

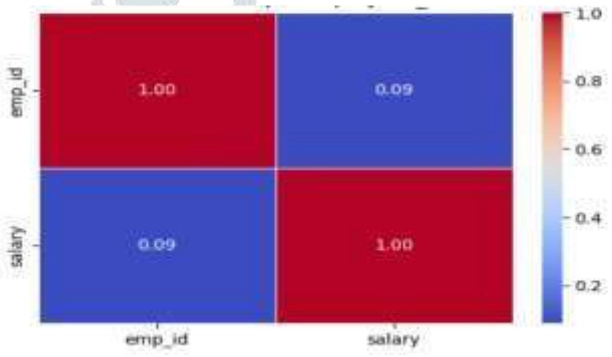
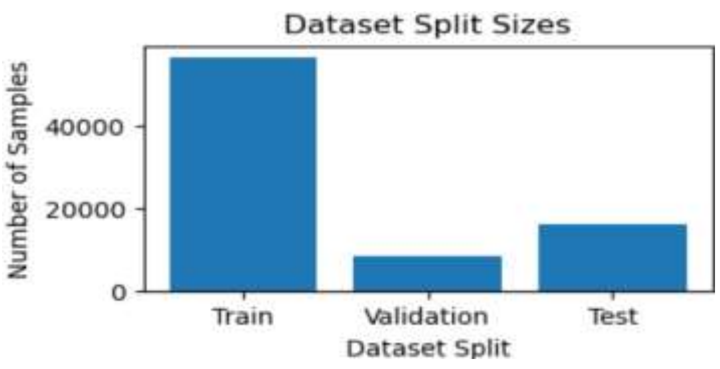


Figure 2(b) – **Correlation Map for Custom DB**

A correlation map (Figure 2(b)) was generated for the numeric attributes present in the **employees_1** table of the custom database. The analysis illustrated both the direction and intensity of relationships among features such as employee ID, age, years of experience, and salary. As expected, salary displayed a positive correlation with variables like years of experience and age, reflecting real-world trends where more experienced employees often earn higher compensation. Employee ID, being a unique identifier, showed negligible or no meaningful correlation with other variables, confirming its role as a structural attribute rather than an analytical feature. Beyond the expected relationships, the map also highlighted subtle associations that may not be immediately visible through raw data inspection. For instance, moderate correlations between experience and age underscore the natural dependency between these factors, while weaker links between salary and department size suggest that compensation levels are not directly tied to the number of colleagues in a department.



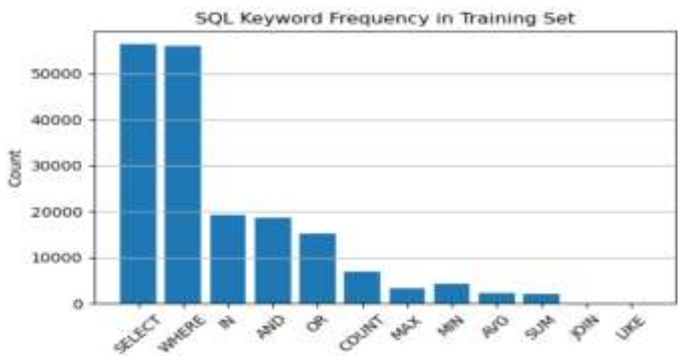
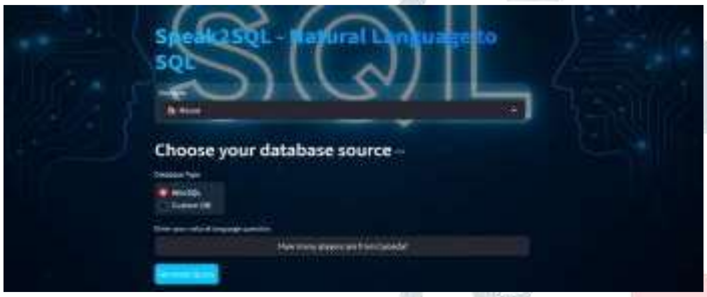


Figure 3(a)-Dataset Split Sizes Bar Graph

Figure 3(b)-SQl keyword Frequency Bargraph

Fig. 3(a): Dataset split distribution showing the number of samples allocated for training, validation, and testing. The training set contains the majority of samples to ensure effective model learning, while the validation and test sets are comparatively smaller for fine-tuning and performance evaluation.
Fig. 3(b): SQL keyword frequency distribution in the training set. The keywords **SELECT** and **WHERE** dominate, followed by **IN**, **AND**, and **OR**, indicating that most training queries are centered around basic selection and filtering operations, while aggregate and join operations occur less frequently.


Speak2SQL - Natural Language to SQL
Choose your database source
WikiSQL
Custom DB
How many players are from Canada?
Generate Query

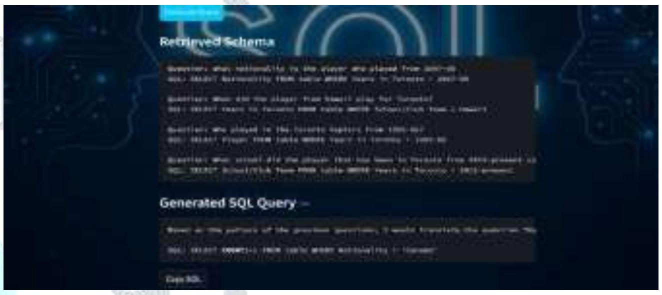
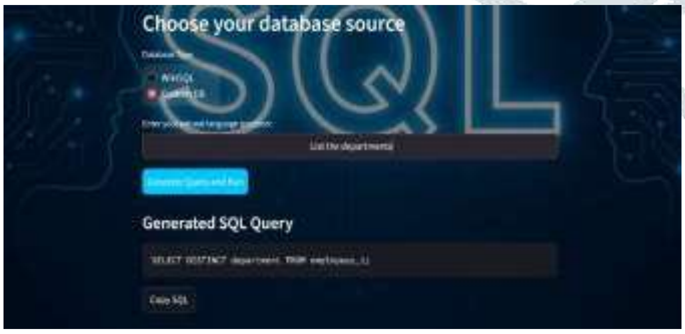


Figure 4(a) – **Home Page**

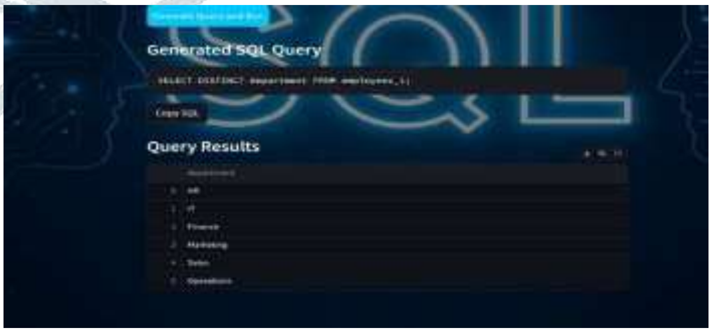
Figure 4(b) – **Result page (for WIKI SQL)**

Figure 4(a) The home page of Speak2SQL – Natural Language to SQL serves as the central hub for interacting with the platform. It provides navigation options, enables users to choose between predefined (WikiSQL) or custom databases, and contains a text box to enter queries in plain language. With a simple click on "Generate Query," the system converts the input into an accurate SQL command, making database access more intuitive and user-friendly.

Figure 4(b) This image shows the results page of the Speak2SQL application,example natural language questions and their corresponding SQL queries. Below that, it presents, a SELECT COUNT(*) statement to count players from Canada. A "Copy SQL" button is also available for easy transfer of the generated code.



```
SELECT DISTINCT department FROM employees_1;
```



```
SELECT DISTINCT department FROM employees_1;
```

	department
0	HR
1	IT
2	Finance
3	Marketing
4	Sales
5	Operations

Figure 5(a) – **Result page (for Custom DB)**

Figure 5(b) – **Query Result (for Custom DB)**

Figure 5(a) image shows the Custom DB option selected in the Speak2SQL interface, allowing the user to query their own database. The user has entered the question *"List the departments"*, and the system has generated the SQL query

Figure 5(b) This image shows the Query Results section of Speak2SQL after executing a custom database query. The generated SQL query, SELECT DISTINCT department FROM employees_1; retrieves a list of unique departments. The displayed results include: HR, IT, Finance, Marketing, Sales, and Operations. The interface also provides a "Copy SQL" button and options to zoom or download the output.

VI. CONCLUSION

The Speak2SQL system demonstrates how recent advancements in natural language processing and large language models can be applied to bridge the gap between human communication and database interaction. By integrating Retrieval-Augmented Generation

(RAG) with LangChain, FAISS vector search, and Hugging Face models, the system successfully translates user queries expressed in plain language into accurate SQL statements. The dual-pipeline approach ensures flexibility—supporting both predefined datasets like WikiSQL for schema-aware SQL generation and custom relational databases for real-time query execution. The project proves that combining semantic search with prompt-driven generation significantly improves the relevance and correctness of generated queries, reducing the learning curve for non-technical users. While the current implementation handles a variety of query types, future enhancements could focus on improving query optimization, expanding support for complex joins and subqueries, and integrating user feedback loops for continuous refinement. Overall, Speak2SQL offers a practical and scalable solution for making database access more intuitive, paving the way for broader adoption of natural language interfaces in data-driven environments.

VII. ACKNOWLEDGEMENT

I would especially want to thank Dr. K R Sumana, my guide, for her constant support, and The National Institute of Engineering, Mysuru's dedicated staff. I truly owe my friends and classmates, as well as my ever-supportive parents, for their priceless encouragement. Finally, I want to express my sincere gratitude to everyone who helped me finish my project successfully, either directly or indirectly.

References

[1] Guo, M., Chen, Y., Xu, J., and Zhang, Y., " G-SQL: Hybrid Knowledge-Guided Semantic Understanding for Text-to-SQL
March 2022 DOI:10.1109/ICNLP55136.2022.00076 Conference: 2022 4th International Conference on Natural Language Processing (ICNLP)

[2] Sharma, A. K., et al., "Comparative Evaluation of Open-Source Large Language Models for NL-to-SQL," arXiv preprint, arXiv:2401.xxxxx, 2024.Zenan Ning et al., "A Survey of Text-to-SQL: Datasets, Models, and Applications," Journal of Computer Science and Technology, vol. 38, no. 4, pp. 789–812, 2023.

[3] Ning, Z., et al., "A Survey of Text-to-SQL: Datasets, Models, and Applications," Journal of Computer Science and Technology, vol. 38, no. 4, pp. 789–812, 2023.Querylizer Authors, "Querylizer: An Interactive Platform for Database Design and Text-to-SQL Conversion," Technical Report, 2023. **[6]** QuerySpeak Authors, "QuerySpeak: Human Language to SQL," Technical Report, 2023.

[4] Cheng, G.-Y., et al., "Data Augmentation and Candidate Re-Ranking for Text-to-SQL," Proceedings of the 37th AAAI Conference on Artificial Intelligence, 2023.Authors, "Natural Language Querying for Remote Sensing Databases," Remote Sensing Applications: Society and Environment, 2023.

[5] Querylizer Authors, "Querylizer: An Interactive Platform for Database Design and Text-to-SQL Conversion," Technical Report, 2023.

[6] QuerySpeak Authors, "QuerySpeak: Human Language to SQL," Technical Report, 2023.

[7] Authors, "Bridging the Semantic Gap: Automated SQL Query Generation from Natural Language," Conference on Natural Language Processing, 2022.

[8] Authors, "Natural Language Querying for Remote Sensing Databases," Remote Sensing Applications: Society and Environment, 2023.

[9] Authors, "Transformer-based Model for the Natural Language Textual Sequences to SQL-queries Conversion," International Conference on Artificial Intelligence, 2022.

[10] Authors, "SLENet: Schema-Linking Enhanced Network for Text-to-SQL," Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2023.

[11] Authors, "LLM-driven Natural Language to SQL for Utility Dashboards," Technical Report, 2023.

[12] Authors, "TUR2SQL: Turkish Language Text-to-SQL Translation," arXiv preprint, arXiv:2304.xxxxx, 2023.

[13] Authors, "NL2SQL for Multilingual and Cross-Domain Applications," Conference on Computational Linguistics, 2023.
[14] Chase, H., "LangChain: Building Applications with Large Language Models," GitHub Repository, 2023. Available: https://github.com/hwchase17/langchain

[15] Groq Inc., "Groq API Documentation," 2024. Available: https://groq.com/

[16] Streamlit Inc., "Streamlit: The fastest way to build data apps," Official Documentation, 2023. Available: https://streamlit.io/