CHATGROQ SQL: CONVERSATIONAL MYSQL INTERFACE USING LLMS, K-NN RETRIEVAL AND BEAM DECODING

Project Reference No.: *49S_BE_1545*

College : *Guru Nanak Dev Engineering College, Bidar*
Branch : *Department of Information Science and Engineering*
Guide(s) : *Prof. Shilpa T*
Student(s) : *Mr. Sarbjot Singh*
Mr. Shaik Nizamuddin Sufiyan
Mr. Syed Saqeef Hussain
Mr. Mohammad Sakib Shaik

Keywords:
Conversational AI, SQL Generation, Large Language Models (LLM), K-NN Retrieval, Vector Database.

Introduction:

1. In today's data-driven environment, large volumes of information are stored in relational databases such as MySQL. Accessing and retrieving this data typically requires knowledge of Structured Query Language (SQL), which can be complex and challenging for non-technical users. This creates a barrier for effective data utilization and limits accessibility for a wider range of users.

2. This project aims to develop a conversational interface that enables users to interact with databases using natural language instead of writing SQL queries. By allowing users to ask questions in plain English, the system simplifies the process of data retrieval and makes it more intuitive and user-friendly.

3. The system leverages advanced Large Language Models (LLMs), such as LLaMA 3, to understand user intent and accurately convert natural language queries into corresponding SQL statements. To improve the precision of query generation, k-NN based semantic retrieval is employed to identify the most relevant database schema elements, including tables and columns, based on similarity with the user query.

4. Furthermore, Retrieval-Augmented Generation (RAG) is used to improve performance by combining user queries with relevant schema context. This helps generate more

5. accurate and context-aware SQL queries. The system also includes validation mechanisms to ensure the correctness and safety of queries before execution.
6. Finally, the results obtained from the database are presented in a clear, structured, and human-readable format, improving user understanding and experience. Overall, the project provides an efficient, scalable, and reliable solution that bridges the gap between complex database systems and non-technical users, making data access faster, easier, and more accessible.

Objectives:

1. To design and develop a conversational interface for querying MySQL databases using natural language.
2. To enable users to interact with databases without requiring knowledge of SQL syntax or schema structure.
3. To implement Large Language Models (LLMs) for understanding user queries and generating SQL statements.
4. To improve query accuracy using k-NN semantic retrieval and Retrieval-Augmented Generation (RAG)

Methodology:

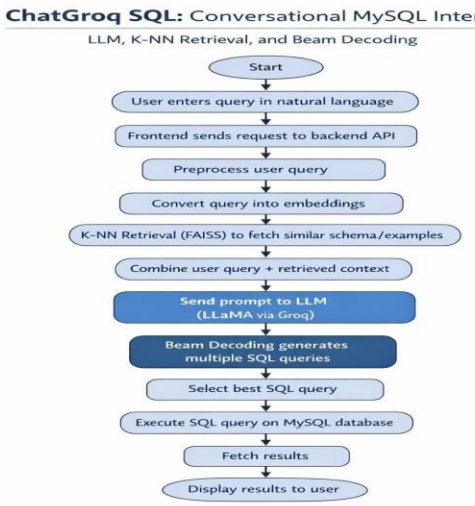
The project aims to develop a system that converts natural language queries into valid and executable SQL queries using advanced AI techniques. It integrates Large Language Models (LLMs), semantic retrieval, and context-aware generation to ensure accurate and reliable results.

The overall process is divided into the following steps:

1. **User Input and Preprocessing**
   The system accepts user queries in natural language through an interactive interface built using **Streamlit**. The input is preprocessed using techniques such as text cleaning, normalization, and tokenization to ensure consistency and better understanding by the model.
2. **Schema Embedding and Storage**
   The database schema (tables and columns) is converted into **vector embeddings** using embedding techniques and stored for efficient retrieval. This enables semantic comparison between user queries and schema components.
3. **Schema Retrieval using k-NN**
   Relevant tables and columns are identified using **k-NN (k-Nearest Neighbors)** based semantic similarity. The system compares query embeddings with stored schema embeddings to fetch only the most relevant schema information, improving efficiency and accuracy.

4. **Context Enhancement using RAG**
   The retrieved schema information is combined with the user query using **Retrieval-Augmented Generation (RAG)**. This enhances contextual understanding and provides the LLM with precise and relevant information for better SQL generation.
5. **SQL Query Generation using LLM**
   A **Large Language Model (LLM)** such as **LLaMA 3 (via Groq API)** is used to generate SQL queries. The system utilizes **LangChain framework** (ChatPromptTemplate, RunnablePassthrough, Output Parsers) to structure prompts and manage the query generation pipeline.
6. **SQL Validation and Execution**
   The generated SQL query is validated for syntax and logical correctness before execution. The system uses **SQLDatabase (LangChain Community)** to securely execute queries on the MySQL database, ensuring safe and reliable operations.
7. **Result Presentation**
   The query results are presented in a structured and human-readable format through the **Streamlit interface**, along with a clear explanation and the SQL query used, improving usability for non-technical users.
8. **System Integration and Technologies Used**
   The system integrates multiple technologies including **Python**, **LangChain**, **Groq API**, **MySQL**, **Streamlit**, and environment management using **dotenv**. Memory handling and optimization are supported using Python utilities like **gc (garbage collection)**.

Below is the detailed diagram of the proposed system architecture illustrating the end-to-end workflow from user query input to SQL query execution and result presentation.



Result and Conclusion:

1. In conclusion, the project presents an intelligent conversational system for querying MySQL databases using natural language, making data interaction more intuitive and accessible. By integrating technologies such as Large Language Models (LLMs), k-NN retrieval, and Retrieval-Augmented Generation (RAG), the system effectively understands user intent and generates accurate, context-aware SQL queries.
2. The system also incorporates validation mechanisms to ensure the correctness and reliability of generated queries. Results are presented in a clear and user-friendly format, enabling even non-technical users to interact with databases without requiring SQL expertise.
3. Overall, the project provides a scalable, efficient, and practical solution for AI-driven database interaction, bridging the gap between natural language and structured data systems.

Project Outcome & Industry Relevance:

1. The project successfully develops an intelligent conversational system that enables users to interact with MySQL databases using natural language queries. It integrates Large Language Models (LLMs) with k-NN (k-Nearest Neighbors) based schema retrieval to generate accurate, efficient, and context-aware SQL queries. By converting database schema elements into vector embeddings and retrieving only the most relevant components through similarity search, the system significantly improves both performance and precision.

2. Furthermore, the incorporation of Retrieval-Augmented Generation (RAG) enhances the system's ability to interpret user queries within the appropriate context, resulting in more reliable SQL generation. The system also includes mechanisms for SQL validation and execution, ensuring that only correct and safe queries are processed. The output is presented in a structured and user-friendly format, making database interaction accessible even to non-technical users.

3. From an industry perspective, the project addresses the growing need for intelligent data access solutions in environments with large and complex datasets. It eliminates the dependency on manual SQL writing, thereby reducing effort, minimizing errors, and improving productivity. Such systems are highly applicable in domains such as business intelligence, data analytics, and enterprise data management, where efficient and accurate data retrieval is critical.
4. Overall, the project delivers a scalable, robust, and industry-relevant solution for modern AI-driven database interaction.

Working Model vs. Simulation/Study:

- The project is implemented as a **fully functional working model**, not a theoretical simulation.
- It supports **real-time interaction** with the user through natural language queries.
- The system connects to an **actual MySQL database** and executes queries dynamically.
- Database schema is converted into **vector embeddings** for efficient processing.
- **k-NN (k-Nearest Neighbors)** is used to retrieve only relevant schema components based on the user query.
- A **Large Language Model (LLM)** generates SQL queries using the retrieved schema context.
- The generated SQL queries are **validated** to ensure correctness and safety before execution.
- The system executes queries and displays **real-time results** to the user.
- It uses tools like **LangChain, Groq API, and Streamlit** for implementation and interface.
- In contrast, a **simulation/study** only demonstrates concepts without real execution or live data interaction.
- Simulation models often use **static datasets**, whereas this project works with dynamic, real data.
- The working model demonstrates **end-to-end functionality**, from input to result presentation.
- The system is **scalable and efficient** due to k-NN-based schema filtering

Project Outcomes and Learnings:

- Developed an **intelligent natural language-based system** for querying MySQL databases.
- Integrated **Large Language Models (LLMs)** for automated and accurate SQL query generation.
- Implemented **k-NN-based schema retrieval** to improve efficiency and relevance of queries.
- Gained understanding of **vector embeddings and Retrieval-Augmented Generation (RAG)**.
- Acquired practical experience in **SQL validation, execution, and backend integration**.
- Strengthened skills in **problem-solving, system design, and building scalable AI-driven applications**.
- Gained exposure to **real-world challenges** in AI-based database interaction.
- Improved ability to build **user-friendly interfaces** for technical systems.
- Developed knowledge of combining **AI, machine learning, and database technologies**.

- Overall, the project enhanced both **technical expertise and practical implementation skills** relevant to industry applications.

References:

1. J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” arXiv preprint arXiv:1810.04805, 2018.
2. T. B. Brown et al., “Language Models are Few-Shot Learners,” arXiv preprint arXiv:2005.14165, 2020.
3. V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning,” arXiv preprint arXiv:1709.00103, 2017.
4. P. Lewis et al., “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” arXiv preprint arXiv:2005.11401, 2020.
5. T. Scholak, N. Schucher, and D. Bahdanau, “PICARD: Executing Language Models on Structured Input and Output Data,” arXiv preprint arXiv:2109.05093, 2021.
6. J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with GPUs,”
7. *IEEE Transactions on Big Data*, 2019.

Future Scope:

The future scope of this project includes:

1. **Voice-Based Conversational Querying**

   Enhancing the system to accept voice input, allowing users to interact with databases through speech for a more natural and user-friendly experience.

2. **Conversational Memory and Context Awareness**

   Enabling the system to remember previous queries and support follow-up questions, making interactions more intelligent and human-like.

3. **Automatic Data Visualization**

   Integrating with tools like Power BI and Tableau to automatically convert query results into charts and dashboards.

4. **Self-Learning and Error Correction System**

   Implementing feedback-based learning where the system improves over time by learning from user