Exploring Enhanced LLM System for Dynamic Database Query: A Comprehensive Study

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

Name
Department of Computer Science and Engineering
Lovely Professional University
Jalandhar, India
nameexample@gmail.com

***Abstract*— In the era of large language models (LLMs), the need for accurate, real-time, and contextually rich information retrieval has become more pronounced. Retrieval-Augmented Generation (RAG) combines the generative power of LLMs with external knowledge sources, enabling dynamic and precise responses to user queries. This paper presents a RAG-enhanced LLM system specifically designed for dynamic database query generation and information retrieval. By integrating a retriever module with a generator, the system ensures that user queries are translated into accurate database queries while maintaining contextual relevance. Our approach reduces hallucination in LLMs and improves the interpretability of results by grounding responses in verifiable data sources. The system is evaluated using both structured and unstructured datasets, demonstrating improved performance compared to traditional LLM-only systems.**

Keywords—RAG, Large Language Model, Information Retrieval, Natural Language Query, Dynamic Database Query, Contextual Retrieval, Data-Driven NLP, Knowledge-Augmented Generation

I. INTRODUCTION

The rapid development of Large Language Models (LLMs) like GPT-4 has transformed how machines understand and generate human-like language. While these models demonstrate exceptional performance in various natural language tasks, they often suffer from hallucinations and limited access to up-to-date or domain-specific knowledge. This gap in dynamic knowledge representation calls for hybrid systems that can combine LLM capabilities with robust retrieval mechanisms to enhance performance.

Retrieval-Augmented Generation (RAG) addresses this issue by integrating a retrieval module with an LLM, allowing the system to pull relevant documents from a knowledge base before generating an answer. This paradigm grounds the model's outputs in real data, significantly improving the reliability of its responses. RAG-based systems are particularly suitable for applications where information must be dynamically sourced, such as in answering question, legal tech, or healthcare systems.

In the context of databases, translating natural language queries into structured query language (SQL) has remained a critical challenge. Traditional models require rule-based methods or extensive training datasets to learn the mapping between natural and formal language. However, these approaches often lack the adaptability needed for diverse and ever-changing database schemas. By using a RAG-based approach, this challenge can be mitigated through dynamic retrieval of schema context and examples during query generation.

This research explores the use of a RAG-enhanced LLM system to enable accurate, dynamic database querying and information retrieval. The system first retrieves relevant database metadata and context using a retriever, which is then fed to the LLM to formulate the appropriate query. This not only increases the query's accuracy but also allows the model to adapt to schema changes with minimal retraining.

The proposed system also supports unstructured knowledge retrieval, extending its utility beyond relational databases. By leveraging vector stores and embedding models, our system can query enterprise documents, logs, or articles to answer questions. This makes it an ideal solution for enterprise AI assistants, research tools, and intelligent query systems across various domains.

II. LITERATURE REVIEW

Recent advancements in natural language processing have led to the widespread adoption of LLMs in various applications. OpenAI's GPT and Google's PaLM have shown remarkable capabilities in natural language understanding and generation. However, these models operate in a closed-book fashion, relying solely on pre-trained knowledge. This limitation makes them less suitable for tasks requiring real-time or domain-specific knowledge access. The introduction of Retrieval-Augmented Generation (RAG) by Facebook AI represents a critical evolution in this space.

Lewis et al. (2020) introduced the RAG framework, which combines a retriever and a generator to produce grounded and fact-based responses. The retriever accesses external documents or knowledge bases, while the generator uses the retrieved context to generate the final output. RAG has been shown to significantly reduce hallucinations and increase factual correctness in generated responses, especially in open-domain question answering tasks.

Natural language to SQL generation has also been an active research area, with models like SQLNet, Seq2SQL, and Spider setting early benchmarks. These models rely heavily on fixed schemas and large annotated datasets. Recent approaches incorporate schema encoding and attention mechanisms to improve generalization, but they still lack dynamic adaptability. Integrating RAG with SQL generation can improve schema understanding and adaptability to new environments.

Several studies have explored the integration of vector databases like FAISS, Weaviate, and Pinecone for document retrieval in LLM-powered systems. These tools allow embedding-based similarity search, enabling efficient and scalable retrieval. When combined with LLMs, they facilitate robust and context-aware information systems. Our proposed approach adopts this paradigm to support both structured and unstructured data retrieval.

In parallel, the emergence of tools like LangChain and LlamaIndex (formerly GPT Index) has further simplified the creation of RAG pipelines. These frameworks provide utilities for chaining LLMs, retrieval modules, and prompt templates in a modular way. Prior research using LangChain has demonstrated improved outcomes in enterprise search, chatbots, and educational platforms. Our work builds upon these foundations, proposing a unified system tailored to database and document query scenarios.

III. PROPOSED METHODOLGY

To improve the model's ability to generalize across different speakers and environments, we apply a variety of data augmentation techniques during the preprocessing stage. These include operations such as time shifting, pitch scaling and random cropping, all of which simulate real-world distortions that might occur in practical scenarios. This augmentation helps the CNN become more robust to variations and prevents overfitting on the training data.

IV. DATASET

V. DATA LOADING AND EXPLORATION

An exploratory data analysis (EDA) phase was conducted to evaluate the balance and diversity within the dataset. A bar chart was plotted to visually assess the distribution of samples across the three categories. This helped ensure that the data was sufficiently stratified and would support an unbiased training process. The visualization also informed further preprocessing strategies by highlighting any slight imbalances or patterns that could affect model performance

VI. DATA PREPROCESSING

VI. MODEL TRAINING

The Model is using

VII. PERFORMANCE EVALUATION

Performance Evaluation

A. EVALUATION METRICS

Evaluation Metrics

VI. PERFORMANCE COMPARISION

Performance Comparison

VIII. RESULTS AND DISCUSSION

Results and Discussion

IX. FUTURE DIRECTIONS

Future Directions

X. REFERENCES