LumenAI: A Conversational Analytics Assistant for Intelligent Database Analysis

Statement of Work

Mohamed Kassem | mok206@g.harvard.edu, **Erich Gunsenheimer** | erg480@g.harvard.edu, **Wilson Guo** | wig979@g.harvard.edu

Background

Advancements in Large Language Models (LLMs) and Machine Learning (ML) have transformed database analysis, making it more accessible. Many organizations store vast amounts of structured data, yet non-technical users struggle to extract insights due to SQL complexity. While NLP-based solutions have attempted to bridge this gap, LLMs offer a more sophisticated and intuitive approach.

LumenAI democratizes database analytics, enabling users to interact with databases via natural language. It interprets SQL schemas, generates optimized queries, suggests business insights, and automates visualizations—offering a conversational AI experience for decision-makers without SQL expertise.

Problem Statement

The rapid growth of structured data in relational databases has created significant challenges for non-technical users who need actionable insights but lack proficiency in SQL and database querying. While business intelligence tools attempt to bridge this gap, they often require manual setup of dashboards, pre-defined reports, and technical intervention from data teams, leading to bottlenecks in decision-making.

Current NL2SQL solutions have shown progress in converting natural language to SQL queries. However, they struggle with complex database schema, chaining SQL queries, user feedback, and narrative explanations. Additionally, data visualization often requires careful context dependent considerations of multiple factors, often necessitating ongoing adjustments from users and experts.

LumenAI addresses these limitations by integrating fine-tuned LLMs, structured data reasoning, and visualization heuristics to create an AI-powered analytics assistant. The system will:

- Automatically interpret database schema and relationships to understand how tables connect.
- Generate SQL queries dynamically for complex, multi-step business questions.
- Allow iterative refinement through a conversational interface, enabling users to refine their queries interactively.

- Provide textual explanations and KPI insights along with query results, making data more actionable.
- Automate visualization selection to choose the best charts based on data structure and intent.

We intend to explore using Graph Neural Networks(GNNs) for Schema Relationship Extraction (Graph-Bert/ GNN-SQL), fine-tune LLMs on NL2SQL Datasets to generate complex SQL(CodeLlama), use AutoTS/ Prophet on historical KPI data to forecast future trends, and deploy Data2Viz for generating visualizations.

Resources

- **Pre-trained LLMs (GPT-4, DeepSeek, CodeLlama)** fine-tuned on SQL and structured analytics.
- **Sample SQL databases** such as **TPC-H, Northwind, and Chinook**.
- **NL2SQL datasets** like **WikiSQL, Spider, and Kaggle's NL2SQL_Query_Dataset** to enhance SQL query generation accuracy.
- **Business KPI datasets** from Kaggle and government sources.
- **Explore visualization models** such as Data2Viz for automated chart generation.

High-Level Project Stages

1. **Data Acquisition & Preprocessing:** Collect SQL schemas, train NL2SQL models.
    a. **WikiSQL**: essentially two public json datasets - one describing the tables and another describing the queries. The first dataset will be helpful in constructing narratives and interpretations of user's SQL schemas. The second dataset can be used to train models to create desired SQL queries.
    b. **Spider**: a public dataset used to train models to generate SQL commands from human prompts. The spider dataset will be helpful in training our model to accurately capture what is being asked and how to respond with SQL. Dataset comprises more than 100 different tables and 7000 training points.
    c. **Kaggle's NL2SQL_Query_Dataset**: Open kaggle dataset also used to train models to go from human speech to SQL. Dataset comprises more than 14000 human prompts.
2. **Prototype Development:** Implement database parsing, LLM-driven query generation.
3. **KPI & Visualization Module:** Integrate automated analytics recommendations and visualizations.
4. **Chatbot Development:** Build a conversational interface for real-time interactions.
5. **Testing & Optimization:** Improve query performance, refine visualization selection.
6. **Deployment & Integration:** LumenAI will be containerized using Docker and orchestrated with Kubernetes on Google Cloud Platform ensuring high availability, efficient resource allocation, and seamless scalability.