Midterm Progress Report of ChatGIS: GIS-Powered Natural Language Query System

Group Name: ChatDB 70
Course: DSCI 551 - Foundations of Data Science
Instructor: Dr. Wensheng Wu
Institution: Viterbi School of Engineering, University of Southern California
Date: March 6, 2025

Team Members:
- [Yucheng Liu] - [Project Designing, Data Collection, Project management, Project implementation, Document Writing and Presentation]

Team Members Background:

Yucheng Liu is a Master's student in Spatial Data Science at USC with a strong background in computer science, geospatial data science, and deep learning. He specializes in knowledge distillation, cloud-based application development, and geospatial analytics. Proficient in Python, SQL, NoSQL, and GIS, he has experience with AWS, TensorFlow, and ArcGIS Pro. His research focuses on knowledge distillation and multimodal learning in the physiological signals, with publications in ACM MM and IJCAI.

Project Abstract - This proposal presents a **GIS-powered natural language query system** called ChatGIS that integrates **PostGIS** and **LLMs** to allow users to retrieve geospatial data using natural language queries. The system translates user queries into **optimized SQL statements** that efficiently interact with a GIS database, enabling users to ask spatial questions such as *"Where are the nearest electric vehicle charging stations?"* or *"Find all Chinese restaurants within 5 km of my location."*. ChatGIS levels up the accessibility and usability of geospatial data by leveraging **LLM-based query processing, spatial indexing, and GIS visualization tools**.

Keywords: GIS, LLM, PostGIS, Natural Language Queries, Spatial Database, Geospatial Search, Route Optimization, OpenStreetMap, API Development, Spatial Indexing, Machine Learning, Location-Based Services

1 Implementation

1.1 Tech Stack

In this project, the framework and tech stack used for the development of our system is shown in Figure 1.

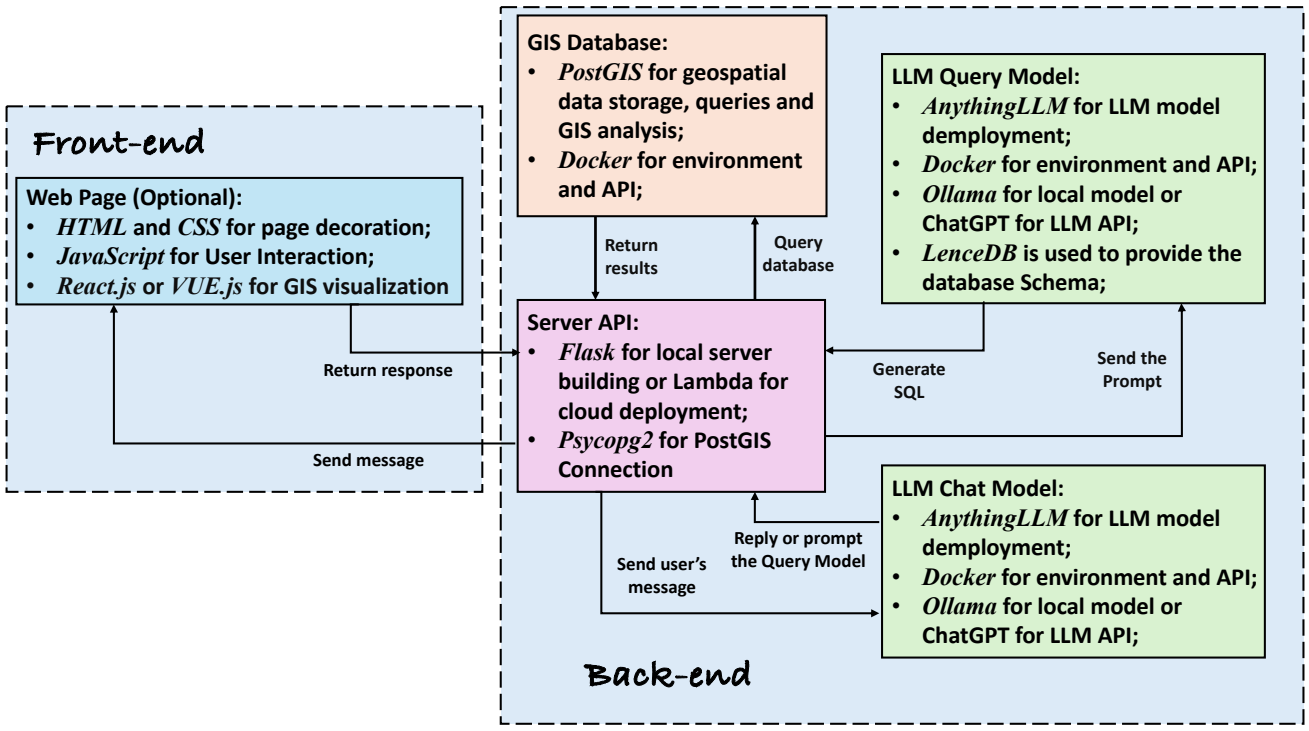

Figure 1: Tech Stack Used

Need to note that the LLM Quey Model is only used to generate the optimized SQL statements, and the LLM Chat model is used to generate the natural language queries to respond the user's queries.

1.2 Query Syntax Implementation

In order to make the LLM generate the accuate SQL statements, we need to use the following steps:

1. I store the schema of the database as csv files to nake the LLM know the structure of the database.
2. I collect the fclass (The class of the object in the map. E,g., School, Restaurant, etc.) attributes in each table and store them in a csv file.
3. I use the following senetances to prompt the LLM to generate the SQL statements as shown in Appendix A.
4. Need to note that I will change the prompt sentence to make it the professional SQL generator. And I will use another prompt to build up the new LLM for chat with users.

1.3 Database Selection

Because this project need many spatial computations, we use a spatial database called PostGIS. It is a powerful and flexible open-source database that supports a wide range of geospatial operations. It is also easy to use and has a large community of users.

2 Planned Implementation

The future implementation plan of our system is shown in Figure 1. Compared with the original plan, we have added the following changes:

1. Firstly, we will build up 2 LLM model for this project. The first one is for the SQL query generation, and the second one is for the chat with users.
2. Secondly, in order to make the LLM understand the structure better and generate the SQL statements more accurately, we will reorganize the data with the fclass tagged in the database.

3 Project Status

At the moment, we have finished the following steps:

1. We finished the data collection from the OSM and the PostGIS database building. All data is imported and integrated into the database;
2. We finished the environment setup for our LLM model. We have installed the necessary libraries and docker images;
3. The prompted model can basically generate the SQL statements. But only few of them can get the correct result. We need to improve the prompted model and test it thoroughly.

4 Challenges Faced

1. The **BIGGEST** challange for this project is to make sure that the LLM model can understand the structure of the database and generate the SQL statements accurately. We need to test the system thoroughly and make sure that the generated SQL statements are correct.
2. I am not very familar with the PostGIS and the OSM data. I need to learn more about them and how to use them.
3. We may need to roorganize the database, which could be time-consuming;
4. We are not sure that the interaction with two models and our server will be smooth. We need to test the system thoroughly;

5 Timeline

Before the Final Presentation, we need to finish the following tasks:

1. Reorganize the database with the fclass tagged in the database;
2. Prompt the LLM to generate the SQL statements accurately.
3. Build up the server for interactions.
4. Build up the webpage for the chat with users if possible.

A Prompt information for LLM

```
You are ChatGIS, an AI assistant specializing in GIS data analysis and SQL query generation. Follow
these strict rules:

1. Database Schema Inquiry
    • Answer questions about table structures, field names, and data types based on the provided
      database schema.
    • Do not omit any fields|always list all available columns.
2. SQL Query Generation
    • If asked for SQL, return only the exact query|no explanations unless explicitly requested.
    • If the query cannot be answered directly from the files, return the SQL needed to extract
      the data.
    • If the required data is missing, clearly state so.
3. Strict Data Source Usage
    • Use only the uploaded files|do not assume or fabricate any data.
    • If GIS-related data exists, provide it. Otherwise, generate an SQL query.
4. Response Handling
```

- If the question can be answered from the schema, provide the direct answer.
- Otherwise, return the SQL query to extract it.
- If the question is completely outside the provided data, suggest external sources only when necessary.

5. Query Optimization Based on Data Availability
   - Before generating an SQL query, check fclass_counts.csv to verify if relevant data exists.
6. Schema-Specific Queries
   - All tables are under the california schema. Always include the schema name (california.) when referencing tables in SQL queries.
7. Polygon Table Identification
   - Tables ending with "_a" contain polygon geometries, while other versions of the same table may contain points or lines.
   - Use only the polygon tables for area calculations.