

Project Title: Designing a Chain-of-Thought-Based LLM System for Solving Complex Spatial Analysis Tasks Through Intelligent Geoprocessing Orchestration

Geospatial challenges such as flood risk assessment, site suitability analysis, or land cover monitoring often demands sophisticated workflows involving various GIS tools and data sources. Traditionally, constructing these workflows requires expert knowledge and manual processes. This project involves building a system that uses reasoning capabilities of Large Language Models (LLMs) to automatically plan and execute geospatial workflows step-by-step much like a human expert.

Objective:

- Develop a reasoning-enabled framework combining LLMs and geoprocessing APIs to auto-generate multi-step geospatial workflows from natural language queries.
- Enable integration of heterogeneous spatial datasets and libraries for tool/resource selection.
- Build an interface translating user queries into sequential geoprocessing tasks with transparent Chain-of-Thought reasoning.
- Demonstrate with benchmark tasks like flood mapping and site selection including input/output, metrics, and visualizations.

Features:

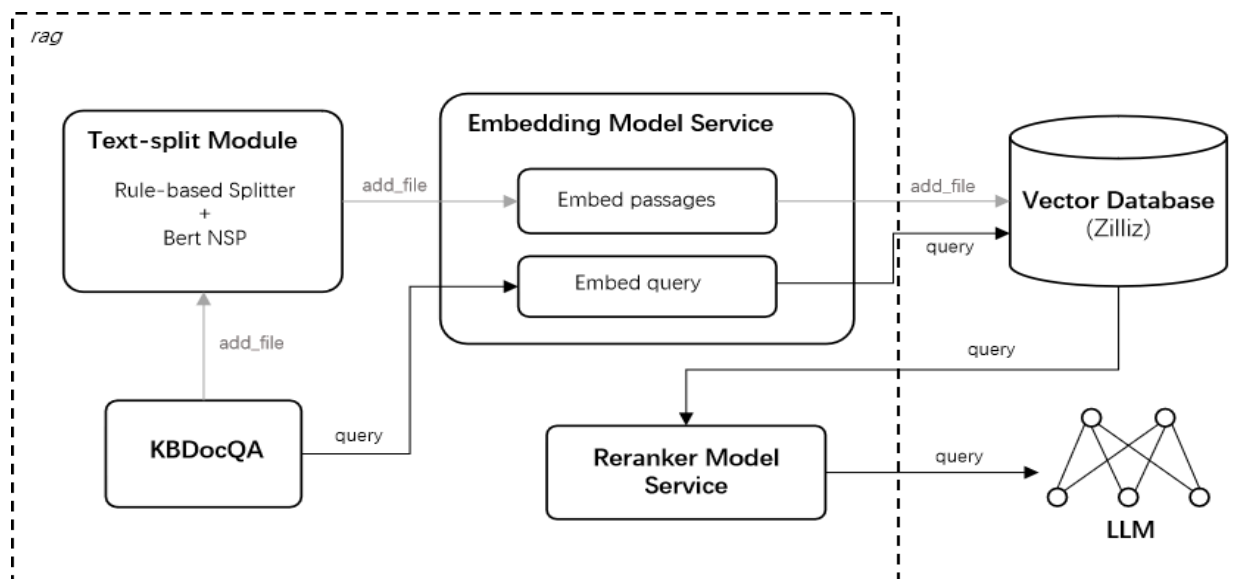
1. Knowledge Base Integration: According to the type, complexity, and real-time requirements of your questions, model (GeoGPT) collaboratively utilizes multiple data sources (such as web search, public databases, personal document libraries, and large model knowledge), efficiently integrates the advantages of all data sources, suitably provides you with the highest quality answers, and comprehensively improves the accuracy and efficiency of its responses.
2. My Document Library:
 - Knowledge Base Management: You can manage your personal collection of research materials by uploading, organizing, and deleting files within the Document Library. These operations enable you to build and maintain a customized knowledge base tailored to your specific research needs.
 - The system limits the file size of each document to 100MB. You can upload up to 50 documents in a single upload operation. Once a document is uploaded, pipeline will automatically start the format parsing process. This parsing process may take some time. You can track the parsing progress of each document in real time via the status bar.
 - You can edit, download, move, or delete a single document.
 - In My Documents, you can perform document retrieval through the search bar shown in the following figure. After you enter a document title and press the Enter

key, your target file will be located quickly. This search function helps you query relevant documents in your personal knowledge base that match the conditions.

- Knowledge Base Q&A: After uploading documents to My Documents and building a personal knowledge base, you can perform Q&A based on the whole knowledge base. Alternatively, you can choose to perform Q&A based only on specific folders. You can view historical questions and current answers, along with the relevant supporting references.
3. Code Execution: You can execute Python code that is compiled successfully within the chat and receive the related execution results.
 4. Deep Discovery: Click the Deep Discovery button in the dialog box to activate it, enter a question you want to explore, and click Send. Model (GeoGPT) will perform Deep Discovery according to your question. This question can be broad or specific. You can observe the progress of the current Deep Discovery process. At the same time, you can view the content explored and read by model during this process in the Activity and Sources window on the right. After a moment, model will generate a detailed Deep Discovery report.

To Do:

1. Deploy GeoGPT-R1-Preview model (open source) on AWS Sagemaker. It's currently on huggingface (<https://huggingface.co/GeoGPT-Research-Project/GeoGPT-R1-Preview>) and can be deployed using huggingface sdk and sagemaker.
2. Implement RAG pipeline, a mixture of GeoGPT-RAG and our own on AWS Infrastructure.
3. Implement Knowledge base Integration and Management, focusing on the RAG pipeline and parsing process. Pipeline includes the two open sourced models, GeoEmbedding and GeoReranker.



4. Create a prompt library for Chain-of-Thought reasoning and in-context learning to generate GIS workflows. Outputs include: JSON/YAML workflow definition, Chain-of-Thought reasoning logs, and GIS outputs.
5. Incorporate a reasoning-acting-observation loop to improve task outcomes iteratively.
6. Enable the model to execute Python code that is compiled successfully within the chat and receive the execution results in the form of an answer. This will be used to perform tool calling and enable the model to reason better based on the code execution results received in chat. This is a very crucial feature.
7. Use tools like PyQGIS, WhiteboxTools integrating with ability to execute Python code in chat to complete GIS workflow steps.
8. Incorporate a reasoning-acting-observation loop to improve task outcomes iteratively.
9. Develop a modern, simple and user-friendly UI for LLM chatbots and integrate features into the left-sidebar. One of the features on the left sidebar is "My Document Library" which will allow the user upload files, custom datasets and other material for contextual understanding, this is where RAG comes in. Other features will be added later. UI for input, output visualization, Chain-of-Thought tracking, and layer downloads
10. Comparative metrics for accuracy, runtime, resource use, and error management versus baseline/manual methods.