Kyle Klein
NASA Internship for
Goddard Space Flight Center
January-May 2025

Leveraging Graph Theory to Increase Utility of Large Language Models

Chainlit is a Python-oriented tool that enables developers to rapidly build a visual user interface (UI) for LLM/chatbot applications. Like many UI-focused tools, it is tightly coupled & implemented with the **JavaScript** language. Some custom JavaScript code was implemented in order to improve user experience with features like pop-up windows of graphs based on user data.

Azure is Microsoft's Cloud Services Platform. NASA has access to other cloud service providers such as AWS and Google Cloud Platform but Azure was used for this project for reasons such as its compatibility with OpenAI and excellent optical character recognition tools.

Optical Character Recognition (OCR)

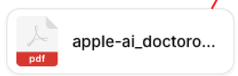
OCR is a technology that has proven extremely useful when working with LLMs. OCR can convert images of text into machine-readable text that an LLM can easily process. PDF files often contain images of text (i.e. scanned text) that has been historically difficult and laborious to extract. Modern OCR tools can convert large PDF files of text into machine readible formats like JSON or markdown extremely quickly.

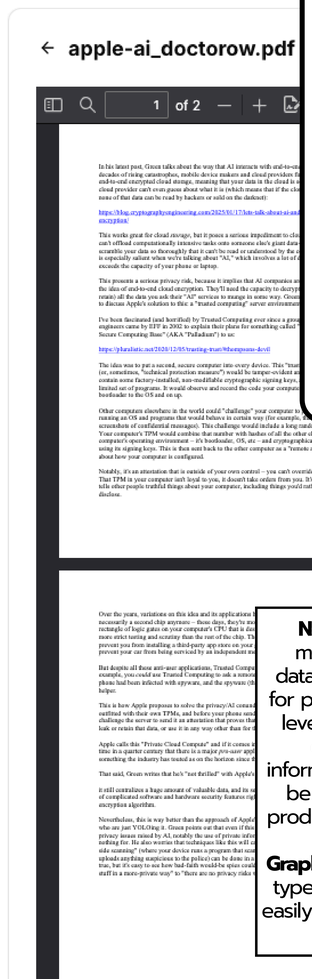
LightRAG

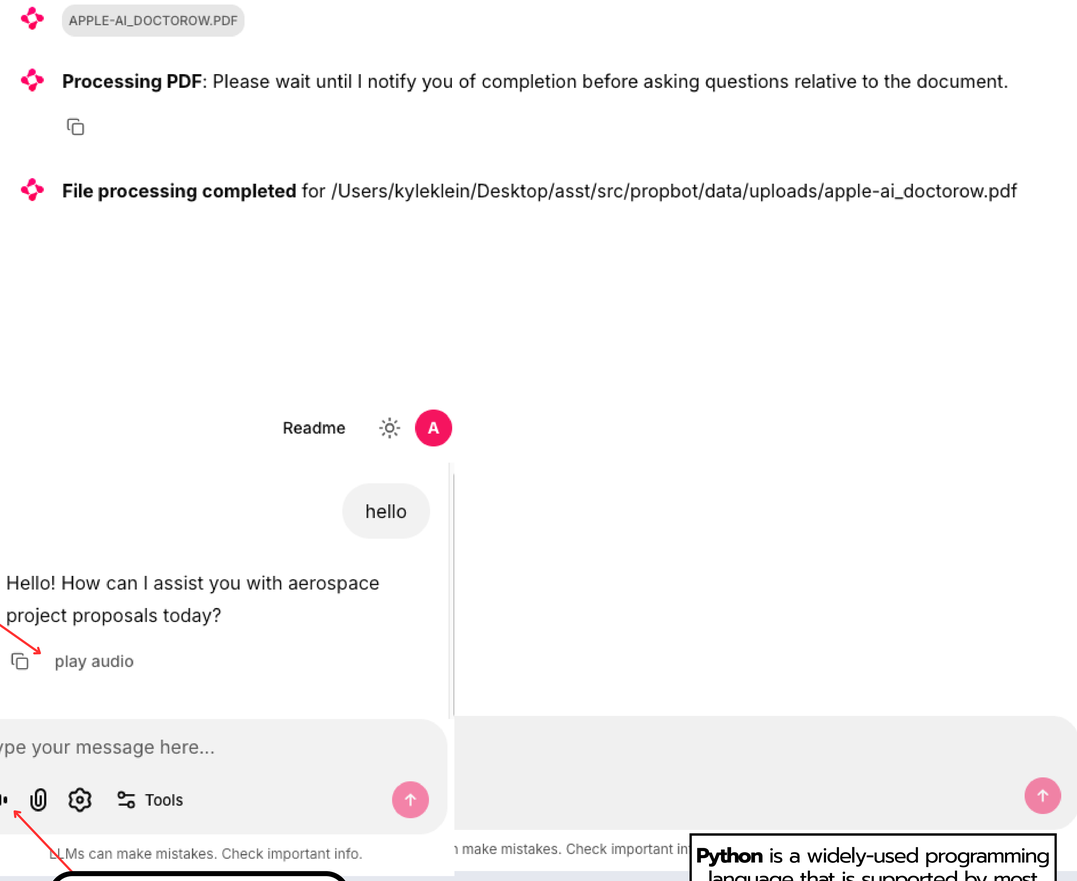
LightRAG is a system that greatly enhances the data and information retrieval of large language models and chatbots. Traditional RAG (retrieval augmented generation) systems enable LLM applications to recall information about user-supplied documents and data by leveraging the concepts of vector databases. Traditional RAG revolutionized the way we interact with LLMs and imbued them with even greater usability. However, the early iterations and paradigms have suffered from the inability to provide high accuracy and wide scope of reference for LLM responses.

LightRAG partially builds and improves upon the theory and tools found in Microsoft's GraphRAG system. GraphRAG introduced the idea of using knowledge graphs and graph database systems to provide more comprehensive and accurate LLM responses when compared to traditional RAG. However, GraphRAG can be quite computationally expensive and slow. In comparison, LightRAG is significantly faster and cheaper while offering comparable performance. LightRAG makes multiple LLM calls to examine and compile information the user is referencing before it passes that data to the actual chat LLM. This sophisticated process enables the chat LLM to provide the user with comprehensive and accurate information.

LightRAG codebase and original research paper can be found at:
github.com/HKUDS/LightRAG







Text-to-Speech

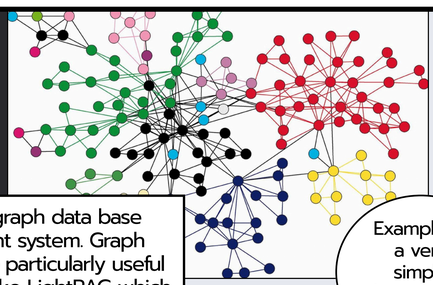
Text-to-speech was added as a feature by leveraging Google Text-to-Speech (**gTTS**). This is a technology that utilizes natural language processing and digital signal processing to generate audio, in this case audio from the chatbot's text responses.

Speech-to-Text

Speech-to-text was added as a feature by leveraging the **OpenAI Whisper** API. This is a technology that utilizes natural language processing and digital signal processing to generate text, in this case text from the user's audio input/ microphone.

While the application had the option to choose models from various providers for the chat UI, **OpenAI** was the preferred provider for general implementation. Reasons for this include high API compatibility (especially with Azure), popular options for smaller models such as gpt-4o-mini for LightRAG processing, and the Whisper API for speech-to-text processing.

Python is a widely-used programming language that is supported by most frameworks and tools. Python was the main programming language used for this project. **Lanchain** (and additionally LangGraph) is a popular platform for quickly creating LLM-centric programs and chatbots. Langchain was implemented in this project to create an LLM backend that could easily process user prompts and utilize tools to give useful and accurate responses.



Example of a very simple knowledge graph

Neo4j is a graph data base management system. Graph databases are particularly useful for programs like LightRAG which leverage knowledge graphs to create a comprehensive information center. Neo4j can also be used as a graph visualizer, producing interactive images akin to these 2 graphs.
GraphQL is a query language and type system that can be used to easily transfer graph data between programs and APIs.

Structured Data Visualizer

Microsoft Excel is a popular program to create spreadsheets or tables from large amounts of structured data. **SQL** is a well-established querying language that operates smoothly with relational databases. Since relational databases are also table-like in nature, it is relatively easy to transfer Excel data into a relational/SQL database. If the data is hierarchical in nature (i.e. tiers of components and sub-components of a space instrument), the data can then be transformed into a hierarchical graph like the one seen here.