Tech Note

This project uses a Retrieval-Augmented Generation (RAG) approach to build an interactive PDF chatbot using Streamlit and LangChain. The system allows users to upload multiple PDF documents, which are then parsed and chunked using RecursiveCharacterTextSplitter with a chunk size of 500 and an overlap of 50. These chunks are embedded via OpenAI's embedding model and stored in a FAISS vector store for fast similarity search.

The core model used is OpenAI’s gpt-4o (with an option to switch to gpt-3.5-turbo), accessed via the ChatOpenAI interface. A custom system prompt guides the assistant to respond only with content grounded in the uploaded documents, cite page numbers, and remain polite and professional. Retrieval is configured with adjustable settings for top-k chunks, candidate pool size (fetch\_k), and MMR diversity (lambda\_mult) to balance relevance and diversity.

One key insight was that overly restrictive prompts like “reply exactly: ‘Document does not mention this’” often led to unhelpful or generic responses. By softening the system prompt and allowing the model to express uncertainty more naturally, the assistant performed more helpfully while staying grounded.

Strengths of the model include reliable citation-based answers, configurable retrieval diversity, and real-time interaction through a user-friendly interface. Weaknesses involve occasional irrelevant retrievals due to embedding similarity limits and minor citation errors when text spans pages or context is fragmented.

In addition to the grounded chatbot that strictly answers based on the uploaded PDFs, we created a creative version that allows more open-ended responses by introducing an alternative system prompt. This creative prompt permits the assistant to speculate beyond the document content when necessary, encouraging plausible responses even if the exact information is not found in the PDFs. The underlying model remains GPT-4o, but the prompt behavior changes dynamically based on a toggle switch in the UI labeled “Allow creative speculation.” This setup enables flexible use cases: grounded answers for precise document review and creative reasoning when broader interpretation is desired, all without altering the rest of the architecture or vector store design.

Team Members and Roles

* Xu Song, Jiaxun Zhang, Yuming Zhang: Led testing configuration, parameter tuning, and validation.
* Jiarui Chen, Xueshan Lin, Jinxian Wu, Yaojie Ye, Xinyu Gong: Focused on development, code implementation, documentation, and presentation deliverables.