RAG CHATBOT

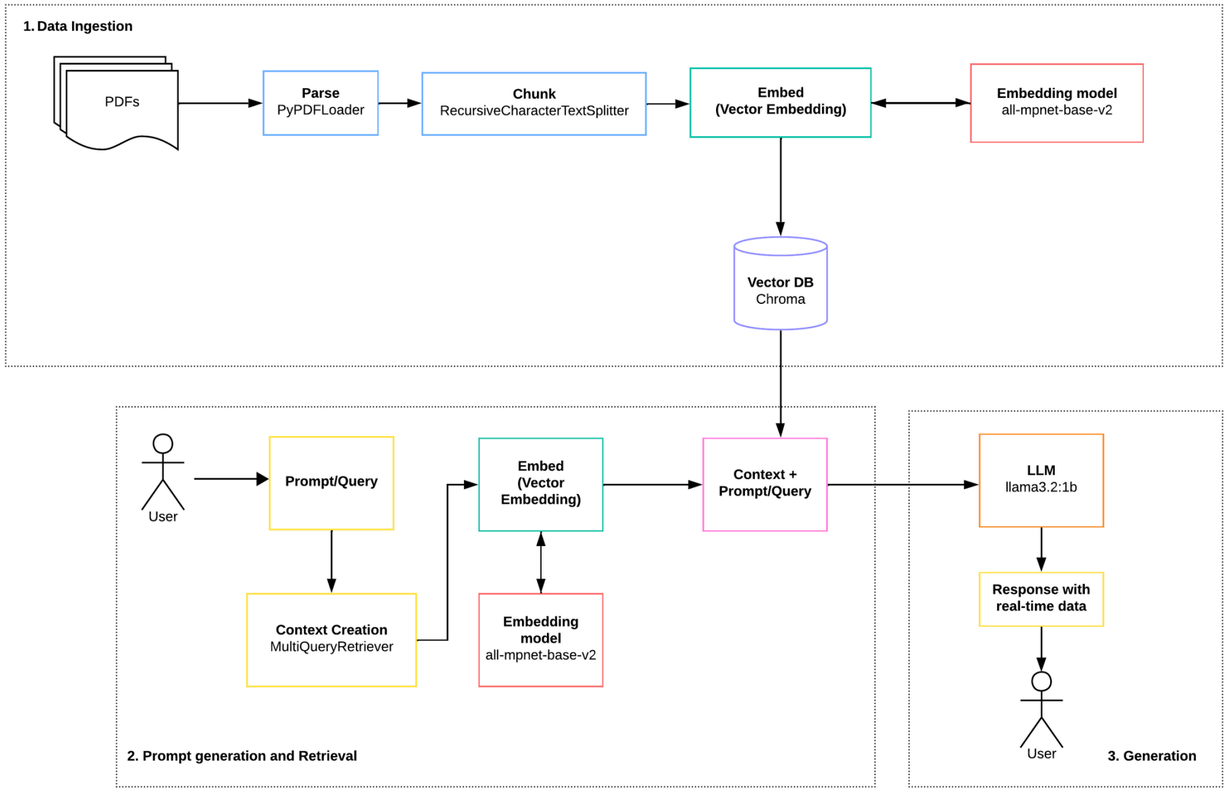
A simple, local, PDF based RAG chatbot using Langchain, Chroma and Ollama

Nidhi Girish

RAG PIPELINE

Divided into 3 major steps

1. **Data Ingestion:** The PDF chosen is parsed, chunked reasonably and added to the Chroma Vector Database after embedding the chunked data.
2. **Prompt Generation and Data Retrieval:** The prompt from user is enhanced by using the MultiQueryRetriever tool and context is created by retrieving data from the vector DB using these prompts
3. **Generation:** The context and prompt are sent to the llama3.2 model for generating a response.



KEY COMPONENTS

Knowledge base

- PyPDFLoader - Loads PDF documents that form your source knowledge
- Chroma - Vector database that stores your document embeddings

Semantic Layer

- all-mpnet-base-v2 - Embedding model that converts text into semantic vector representations
- RecursiveCharacterTextSplitter - Chunks documents into semantically meaningful pieces for embedding

KEY COMPONENTS

Retrieval System

- MultiQueryRetriever - Generates multiple query variations to improve retrieval coverage and finds relevant chunks from Chroma

Augmentation

- Retrieved chunks are formatted and combined with the modified user's query
- Creates enriched context with MultiQueryRetriever response for the LLM

Generation

- llama3.2:1b - Large Language Model that generates responses based on the augmented context

CHALLENGES

- Research for the right combination of tools
- Setting up packages and running locally (version issues)
- Choice between embedding models
- Choice between ollama LLM models

THANKYOU

Open to Q&A!