S. No	File Name	Retriever Model	Generator Model	Retriever Issues	Generator Issues
1	**RAG-LangChain-Base**	**HuggingFace embeddings (hkunlp/instructor-base)** were used to create semantic embeddings from the PDF document, which were subsequently stored in a **FAISS vector database** to build the retriever model.	The generator model used is **fastchat-t5-3b-v1.0**, a 3-billion-parameter **T5** variant from **Hugging Face**, fine-tuned specifically for conversational tasks and instruction-following.	The text splitting strategy (chunk size and overlap) may split meaningful context, resulting in incomplete retrieval, causing a miss in crucial context needed for accurate responses	FastChat-T5 typically has a relatively small token limit (~512 to 1024 tokens). Large contexts or extensive prompt data might exceed this limit, causing essential context information to get truncated, causing incomplete or irrelevant responses.
2	**RAG-LangChain-OpenAI**	The retriever model utilizes **OpenAI embeddings** to generate semantic embeddings from the PDF document, which are stored in a **FAISS vector database** for efficient semantic retrieval.	The generator model used is **OpenAI GPT-4o-mini**, integrated with a predefined **LangChain prompt template** designed to guide the model's responses for accurate and context-aware generation.	OpenAI embeddings, while effective, might generalize concepts excessively, resulting in loss of subtle context distinctions, causing retrieval of superficially similar but contextually irrelevant content.	GPT-4o-mini, being a smaller variant, may have tighter context length constraints compared to standard GPT-4, causing longer contexts or extensive retrieval data to be truncated, affecting response quality.