Technical Architecture – Local RAG System

---

1. System Overview

We implemented a Retrieval-Augmented Generation (RAG) architecture for document question answering using fully local open-source components. The system consists of four primary layers: Document Processing, Vectorization & Indexing, Retrieval, and Generation.

2. Architecture Flow

- PDF is loaded and parsed into raw text.
- Text is split into manageable chunks with overlap.
- Chunks are converted into semantic vectors using BGE-base embedding model (~110M parameters).
- Vectors are stored inside FAISS for fast similarity search.
- User query is embedded and matched against stored vectors.
- Top-K relevant chunks are retrieved.
- Flan-T5-small (~80M parameters) generates grounded answer using retrieved context.

3. Models & Parameters

- Embedding Model: BAAI/bge-base-en (~110M parameters) – Converts text into semantic vectors.
- Language Model: google/flan-t5-small (~80M parameters) – Generates context-aware answers.
- Total parameter footprint: ~190M parameters (CPU compatible).

4. What We Achieved

- Fully local AI-powered document question answering system.
- No external API usage (data privacy preserved).
- Semantic search via FAISS.
- Context-grounded answer generation.
- Modular and scalable architecture.

5. Future Directions

- Upgrade to larger LLM (Flan-T5-base or Mistral-7B) for improved reasoning.
- Add reranker for enhanced retrieval precision.
- Support multi-PDF knowledge base.
- Deploy as internal web application.
- Scale to GPU for higher performance.