Document Summarization with Retrieval-Augmented Generation (RAG)

Subhan Ahmed

22i-2497

Introduction

This project implements a Retrieval-Augmented Generation (RAG) system for document summarization, addressing the need for concise, coherent summaries of long documents. The system ingests documents in PDF, TXT, or Markdown formats, splits them into chunks, embeds them using SentenceTransformers, retrieves relevant chunks with FAISS, and generates summaries using the BART model. The objective is to produce fluent, accurate summaries while evaluating their quality. The system processes three sample documents: a generative AI paper, an ArXiv article, and a CNN/DailyMail article.

Methodology

The pipeline consists of four modules:
1. Document Ingestion: Loads documents using LangChain's PyPDFLoader and TextLoader, removes metadata and references with regex, and splits text into 2000-character chunks with 400-character overlap using RecursiveCharacterTextSplitter.
2. Embedding and Retrieval: Converts chunks to embeddings with all-MiniLM-L6-v2 (SentenceTransformers), normalizes them for cosine similarity, and stores them in a FAISS IndexFlatIP. Retrieves top-5 chunks for queries like "Summarize this document."
3. Summary Generation: Concatenates retrieved chunks and generates summaries using facebook/bart-base with beam search (4 beams, max length 150). Tracks input/output tokens and latency.
4. Evaluation and Output: Evaluates summaries for fluency (sentence structure), coverage (key term presence), and accuracy (no contradictions/metadata). Saves results and metrics to text files.

The system uses open-source tools (LangChain, FAISS, SentenceTransformers, Hugging Face Transformers) for accessibility and efficiency. BART was chosen over GPT or LLaMA for its lightweight, open-source nature and strong summarization performance

Challenges and Future Work

The challenges that I faced were mostly hardware related. Actually I have a ram of 8 GB out of which almost 7 GB continuously run. So , my laptop couldn’t bear larger models or anything else open while running the application. Future improvements include:

• Semantic chunking via sentence embedding clustering.
• Larger models (e.g., bart-large-cnn) for better summaries.
• Visualizations of similarity scores or evaluation metrics.

Conclusion

The RAG-based summarization system successfully processes and summarizes documents, achieving high fluency and reasonable coverage/accuracy. While the pipeline is modular and reproducible, enhancements in chunking and evaluation could further improve performance.