RAG Document Summarization Project Report

Author: Fasih

Date: June 15, 2025

Overview

This report details the implementation of a Retrieval-Augmented Generation (RAG) system for document summarization, developed as part of a coursework assignment. The system processes documents in PDF format, splits them into chunks, generates vector embeddings, retrieves relevant content using a FAISS vector database, and produces summaries with a pre-trained BART model. The project evaluates the pipeline's performance on three sample documents: an introduction to RAG (1.pdf), a wildfire incident (3.pdf), and a puppy rescue story (2.pdf).

Methodology

The RAG pipeline follows a structured approach:

- Document Parsing: Documents are loaded using PyPDF2, with text cleaned to remove excessive newlines and spaces. The RecursiveCharacterTextSplitter from LangChain divides text into chunks of approximately 500 characters with 100-character overlap.

- Embedding Generation: The all-MiniLM-L6-v2 model from SentenceTransformers creates 384-dimensional vector embeddings for each chunk.

- Storage: Embeddings are stored in a FAISS IndexFlatL2 vector database for efficient similarity search.

- Retrieval: A query ("Summarize this document") is embedded, and the top-k (up to 2-3) most relevant chunks are retrieved based on cosine similarity.

- Summarization: The facebook/bart-large-cnn model generates a summary from the retrieved chunks, with a maximum length of 200 tokens.

The system operates on CPU, with a custom cache directory (D:\HuggingFaceCache) to manage storage.

Results

The system was tested on three documents, with the following outputs:

1. 1.pdf (Retrieval-Augmented Generation)

- Summary: Retrieval-Augmented Generation (RAG) is a hybrid approach combining information retrieval and natural language generation. It enhances large language models (LLMs) by retrieving relevant document chunks from a knowledge base before generating responses. This method is particularly effective for tasks requiring factual accuracy, such as document summarization or question answering.

- Retrieved Context:

  - Chunk 1: Retrieval-Augmented Generation (RAG) is a hybrid approach combining information retrieval and natural language generation. It enhances large language models (LLMs) by retrieving relevant document chunks from a knowledge base before generating responses. This method is particularly effective for tasks requiring factual accuracy, such as document summarization or question answering. RAG systems typically involve two components: a retriever and a generator

  - Chunk 2: . RAG systems typically involve two components: a retriever and a generator. The retriever uses vector embeddings to identify relevant text chunks, while the generator, often a pre-trained LLM like GPT or BART, produces coherent outputs based on the retrieved context. This combination reduces hallucinations and improves response quality.

- Metadata: Similarity Scores: [1.3283215761184692, 1.7366697788238525], Latency: 3.16 seconds

2. 2.pdf (Puppy Rescue Story)

- Summary: Lena was walking home from school when she saw a small puppy trembling under a bench. She asked nearby people, but no one claimed the dog. A week later, a poster appeared — the puppy had a family. Though sad, Lena returned Milo, and they remained friends.

- Retrieved Context:

  - Chunk 1: Lena was walking home from school when she saw a small puppy trembling under a bench. It had no collar and looked hungry, so she gave it her sandwich. She asked nearby people, but no one claimed the dog. That evening, she brought it home and named him Milo. A week later, a poster appeared — the puppy had a family. Though sad, Lena returned Milo, and they remained friends.

- Metadata: Similarity Scores: [1.853224277496338], Latency: 2.65 seconds

3. 3.pdf (Wildfire Incident)

- Summary: lightning strike sparks a wildfire in the Pine Valley forest. Local firefighters rushed in, supported by helicopters dropping water. Wild animals fled to safety, and nearby towns evacuated. After two days of intense battle, the fire was finally contained. It left behind ashes but also united the community.

- Retrieved Context:

  - Chunk 1: A sudden lightning strike sparked a wildfire in the Pine Valley forest. The dry summer made the flames spread rapidly. Local firefighters rushed in, supported by helicopters dropping water. Wild animals fled to safety, and nearby towns evacuated. After two days of intense battle, the fire was finally contained. It left behind ashes but also united the community.

- Metadata: Similarity Scores: [1.8040711879730225], Latency: 2.51 seconds

The latencies range from 2.51 to 3.16 seconds, with similarity scores indicating reasonable relevance (1.33 to 1.85). The summaries vary in accuracy, with 1.pdf and 2.pdf capturing core ideas, while 3.pdf omits the dry summer context.

Challenges

Several challenges were encountered during development:

- Chunk Fragmentation: Initial runs produced fragmented chunks (e.g., individual words), addressed by improving text preprocessing with regular expressions to normalize newlines and spaces.

- Summary Accuracy: Early summaries were incoherent due to repeated or fragmented chunks; this was mitigated by fixing retrieval logic, though some inaccuracies persist (e.g., 3.pdf summary lacks full context).

- TensorFlow Warnings: Unnecessary TensorFlow logs were suppressed using environment variables (TF_ENABLE_ONEDNN_OPTS=0, TF_CPP_MIN_LOG_LEVEL=3), improving output clarity.

Future improvements could involve using pdfplumber for better PDF parsing, fine-tuning the BART model, or adjusting chunk size for optimal summarization.

Conclusion

The RAG summarizer successfully processes and summarizes three diverse documents, demonstrating a functional pipeline for automated summarization. The system handles document parsing, embedding, retrieval, and generation, with latencies under 3.2 seconds. However, challenges in chunk coherence and summary accuracy suggest areas for refinement. This project showcases the practical application of RAG, providing a foundation for further enhancement to meet full rubric expectations.