Task 2 – Optimizing RAG: Two Techniques

Name: Adavally Lokesh Reddy

Date: 05 July 2025

Introduction

In the previous task, I built a basic RAG (Retrieval-Augmented Generation) model that takes information from a PDF and answers questions using OpenAI and Pinecone. While it works, there are ways to improve how well it retrieves and answers. Below are two methods that I think can make the system work better.

1. Mixing Two Search Methods (Keyword and Semantic Search Together)

Usually, RAG uses something called "dense embeddings" to find the most related text. But sometimes, this misses important keywords or exact terms. So one way to improve is by using both:

- **Keyword-based search** like BM25 or TF-IDF
- And the usual **semantic search** (dense vectors)

We can first use keyword search to get some relevant chunks and then use vector-based search to refine those results. This can help make sure we're not missing anything important just because it doesn't "sound" similar.

2. Making the Context Smaller by Summarizing

When we pass the retrieved text into the GPT model, it might be too long or include extra information. Instead of giving it all, we can use a summarizer (even a small model) to make it shorter.

For example:

- After we get 3 chunks from Pinecone, we summarize them into 1 short paragraph.
- Then we send that as the context to GPT.

This makes the answer more focused and also helps avoid hitting the token limit.

Conclusion

These two changes – mixing search methods and summarizing the context – can make the RAG system smarter and more efficient. They are not too hard to implement but can make a noticeable difference in the quality of the answers.