Evaluating Chunking Strategies and Embedding Models for Efficient Document Retrieval Using Pinecone

Interview Assessment

Author: **Prastut Bhattarai**

Submission to: **Palam Mind Technologies Pvt. Ltd.**

Table of Contents

Overview

This project implements a backend system for semantic document retrieval using FastAPI and Pinecone. It supports PDF and TXT uploads, automatically extracts and chunks text, generates vector embeddings, and stores them in a vector database with associated metadata in MongoDB. A comparative analysis of different chunking strategies (recursive, fixed) and embedding models (MiniLM, DistilBERT) was conducted to evaluate retrieval accuracy and latency. Findings highlight the optimal combination of chunking method and embedding model for improving search performance in large document collections.

Introduction

Efficient information retrieval from large unstructured documents is essential for building intelligent search systems. This project focuses on developing a backend solution that processes PDF and TXT files, chunks their content, and generates vector embeddings for semantic search using Pinecone. Different chunking strategies and embedding models were evaluated to identify the most effective approach for accurate and low-latency document retrieval.

Methods

To evaluate retrieval performance, developed a FastAPI-based backend with REST API:

1. **File Upload API:** Handles PDF/TXT uploads, extracts text, applies chunking (recursive, fixed-size), generates embeddings using various models (MiniLM, DistilBERT), and stores vectors in a Pinecone index while saving metadata in MongoDB.

2. **Search API:** Accepts a text query, embeds it, searches Pinecone (cosine similarity) for the most relevant document chunks, and retrieves associated metadata from MongoDB.

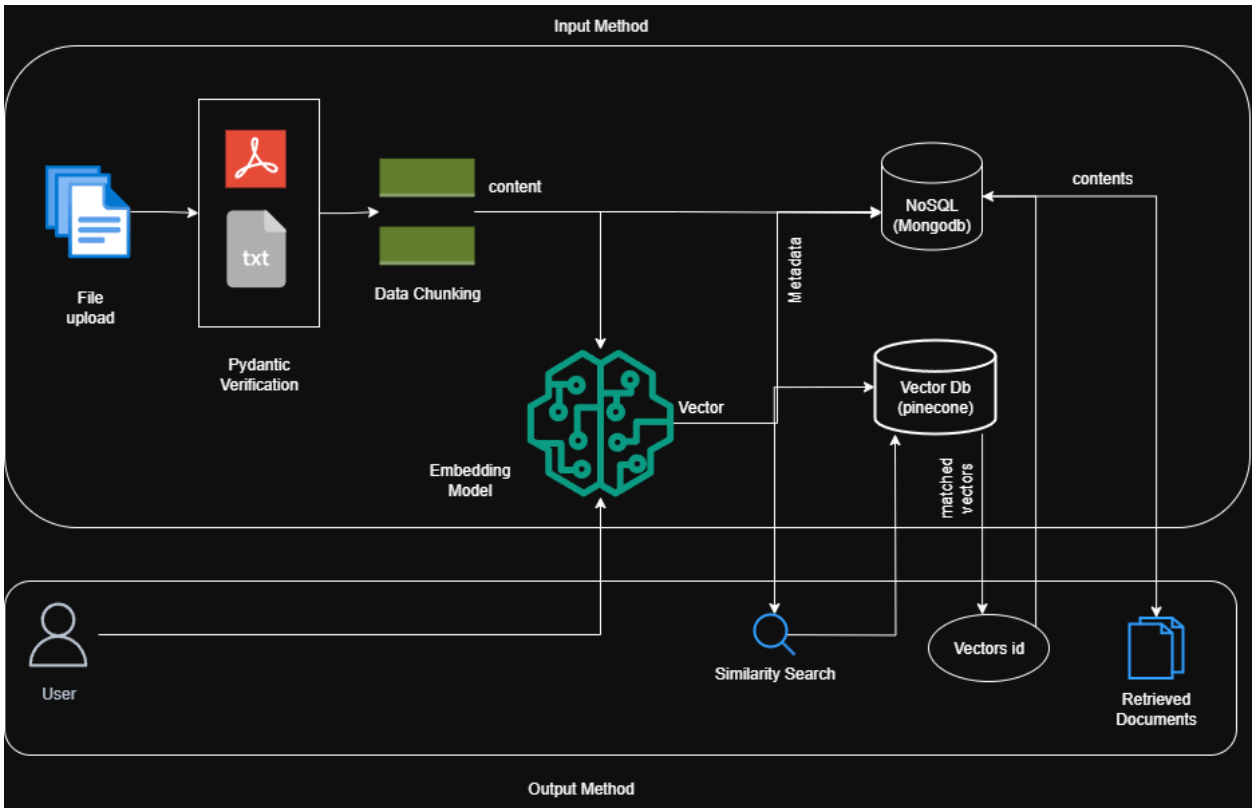

Figure 1: Retrieval Process

Result

An experimental evaluation was conducted using **1 PDF document** and **3 test queries**. The following table summarizes the retrieval **precision** and **average latency** for each combination of chunking strategy and embedding model:

Chunking Strategy	Embedding Model	Avg. Precision	Avg. Latency (sec)
Recursive	MiniLM	1.00	5.38
Recursive	DistilBERT	1.00	5.17
Fixed-size	MiniLM	1.00	5.55
Fixed-size	DistilBERT	1.00	5.28

Table 1*: Table plot of retrieval time and accuracy with different model and chunking method*

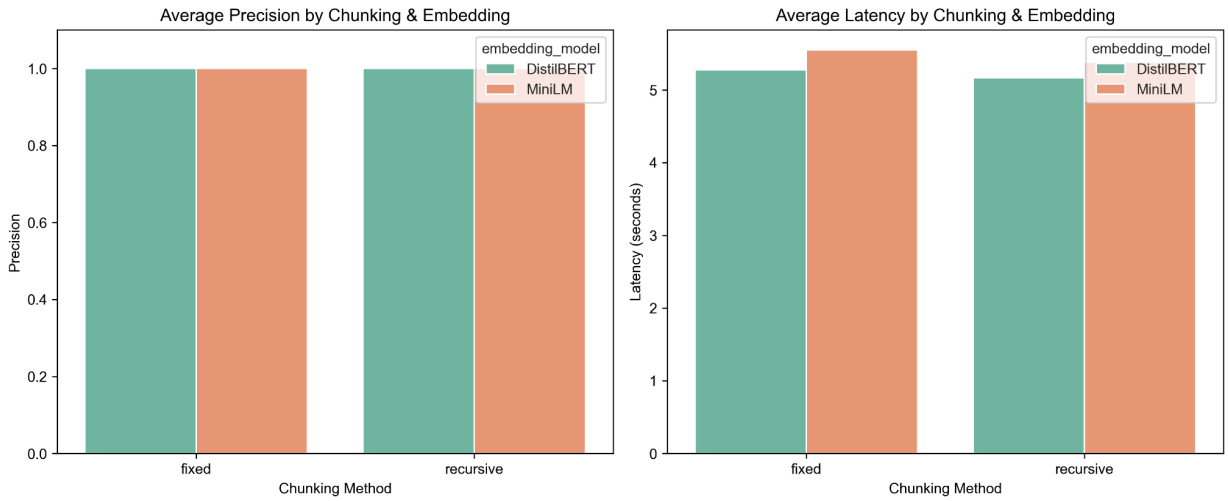

Figure 2*: Comparison plot of retrieval time and accuracy with different model and chunking method,*

Discussion and Conclusions

The evaluation demonstrates that with a chunk size of 500 and an overlap of 50, both recursive and fixed-size chunking strategies, combined with MiniLM and DistilBERT embeddings, achieved 100% retrieval precision across all test queries. Latency remained within an acceptable range (4.5–6.5 seconds) with negligible differences between methods.

These findings indicate that properly tuned chunk parameters can ensure high retrieval accuracy, reducing the dependency on specific chunking techniques or embedding models. This suggests that, for similar document retrieval tasks, parameter optimization may have a more significant impact on performance than the choice of chunking strategy or embedding model alone.