Project Overview

The project involves developing an app to extract and highlight key information from PDF documents using a local language model. This tool is particularly useful for quickly retrieving specific details from documents like resumes, legal papers, or research articles. The assignment was straightforward and focused on enhancing text extraction accuracy and efficiency.

1. Challenges Faced and Solutions

1.1. Text Extraction and Highlighting

Challenge: Extracting and accurately highlighting specific phrases in PDFs was problematic due to non-standard text encoding, hidden characters, and line breaks. The initial exact phrase matching approach failed when text was split across lines or included hidden characters.

Solution: The approach was revised to normalize the extracted text by removing extra spaces and converting it to lowercase. Instead of searching for entire phrases, each phrase was split into individual words, searching and highlighting each word separately. This method improved the accuracy of highlighted sections within PDFs.

1.2. Extracting Relevant Parts Based on Query

Challenge: Extracting relevant text from PDFs based on user queries proved challenging. The initial regex matching approach lacked accuracy, especially in identifying nuanced or contextually relevant information.

Solution: Transitioned to using a pre-trained sentence embedding model from the Sentence-Transformer library. This model converts both the query and PDF text into numerical embeddings capturing semantic meaning. By comparing these embeddings, the tool effectively identifies and retrieves sentences most similar to the query, significantly enhancing information extraction precision for complex or context-dependent queries.

2. Potential Improvements and Scalability

2.1. Processing Time Optimization

Current Limitation: The current CPU-based model setup results in slower processing times, particularly when handling multiple PDFs.

Future Improvements: Explore more efficient model architectures and implement GPU acceleration. Consider parallel processing and batching queries to enhance responsiveness and reduce processing time.

2.2. Scalability

Consideration: As the volume of PDFs and query complexity increase, efficient handling of larger datasets becomes essential.

Future Improvements: Optimize the app for large datasets by exploring distributed computing solutions and enhancing data pipelines. Upgrade infrastructure to support higher concurrency and load, ensuring consistent performance.