Documentation: Financial Statement Processing Pipeline

This project is designed to process financial statements from PDF documents, extract and structure the text, and generate accurate answers to financial queries using advanced natural language processing (NLP) techniques. Below is a detailed explanation of the workflow and its components:

1. Text Extraction:

- The first step in the pipeline is extracting text from the PDF document. Since financial statements can contain both text and scanned images, the system uses a dual approach:

- Optical Character Recognition (OCR): Converts PDF pages into images and extracts text using Tesseract, a powerful OCR engine.

- Fallback to PDF Text Extraction: If OCR fails or extracts insufficient text, the system falls back to pdfplumber, a library that directly extracts text from PDFs.

- This ensures robust text extraction, even for documents with mixed content.

2. Text Structuring

Once the text is extracted, it is structured into a more organized format for easier processing. The system identifies:

- Key-Value Pairs: Lines containing a colon separator (e.g., "Revenue: 1,000,000 crore") are detected and stored as key-value pairs.
- Tables: Rows with multiple columns (detected using spaces or pipe characters) are identified and stored as tables.
- Unstructured Text: Any remaining text is stored as plain text.
- This structured data is easier to analyze and process in subsequent steps.

3. Text Chunking

- To make the text more manageable for language models, the structured data is split into smaller chunks. This is done using a Recursive Character Text Splitter, which:
- Splits the text into chunks of a specified size (e.g., 1000 characters).
- Ensures overlap between chunks (e.g., 296 characters) to maintain context.
- Uses separators like paragraphs, lines, and pipe characters to create meaningful chunks.
- Chunking improves the efficiency and accuracy of language models by providing them with smaller, context-rich segments of text.

4. Vector Embedding and Storage

- The text chunks are converted into numerical representations called embeddings using a pre-trained sentence transformer model (all-MiniLM-L6-v2). These embeddings capture the semantic meaning of the text and are stored in a Pinecone vector database. Pinecone allows for:

- Fast and efficient retrieval of relevant text chunks based on similarity.
- Scalable storage of embeddings for large datasets.

- The system ensures that the embeddings are stored with metadata (e.g., truncated text) for easy reference during query processing.

5. Query Processing and Response Generation

When a financial query is received, the system:

- Converts the query into an embedding using the same sentence transformer model.
- Retrieves the most relevant text chunks from Pinecone based on similarity.
- Constructs a prompt using the retrieved context and the query.
- Uses OpenAI's GPT-3.5-turbo model to generate a concise and accurate answer.
- The system is designed to handle a wide range of financial queries, such as revenue, expenses, profit, and balance sheet details.

6. Output and Storage

The generated questions and answers are stored in a JSON file for future reference. This file contains:

- The original financial queries.
- The corresponding answers generated by the system.
- This output can be used for reporting, analysis, or further processing.

Key Features

- Robust Text Extraction: Combines OCR and PDF text extraction for reliable text retrieval.
- Structured Data: Organizes text into key-value pairs, tables, and plain text for easier analysis.
- Efficient Chunking: Splits text into smaller, context-rich chunks for better processing by language models.
- Semantic Search: Uses embeddings and Pinecone for fast and accurate retrieval of relevant text.
- Accurate Answers: Leverages OpenAI's GPT-3.5-turbo to generate precise and context-aware responses.

Use Cases

This pipeline is ideal for:

- Automating financial document analysis.
- Generating answers to financial queries in real-time.
- Extracting and organizing data from complex financial statements.
- Enhancing decision-making with accurate and timely financial insights.

How to Use

- Provide the path to the financial statement PDF.
- Run the pipeline to extract, structure, and process the text.
- Query the system with financial questions and retrieve accurate answers.
- Access the generated questions and answers in the output JSON file.