PDF Title and Outline Extractor - Technical Report

PDF Processor: Overview and Report

Overview

This report explains a Python-based PDF processor that extracts the title and headings (outline) from PDF documents. It is built for automated processing of multiple PDFs and saves the extracted metadata into structured JSON files. The system is ideal for bulk document processing, such as digital archiving or indexing educational, technical, or corporate PDFs.

Libraries Used

The script uses several libraries:
- PyMuPDF (imported as fitz): The core engine for PDF parsing, including reading text, blocks, and styles.
- pathlib and os: For clean file system path handling and folder creation.
- json: To store extracted data in JSON format.
- re (regular expressions): For cleaning and matching text patterns.
- concurrent.futures: To use multithreading and speed up processing across multiple pages and files.

What the Script Does

The script defines a class named PDFProcessor, which handles the logic of extracting a title and an outline (i.e., headings or major sections) from each PDF. It processes all PDF files in a specified input folder and outputs one JSON file per PDF to a specified output folder.

Title Extraction

The title is extracted from the first page of each PDF. The logic assumes that the title is one of the largest pieces of text on the page, located toward the top. It assigns a "weight" to each text span using a combination of its font size and vertical position, and picks the most prominent one as the document title. Short words, numbers, and decorative lines are filtered out to avoid false positives.

Outline Extraction

The script scans each page (up to 50 pages per document) and tries to find headings based on font size. Larger font sizes are treated as headings. The size threshold determines the heading level (H1, H2, H3, H4). The script ignores noisy text like URLs or fully capitalized lines and merges lines properly even if broken across spans. To avoid duplication, it maintains a set of previously seen lines.

Performance

To improve speed, the script uses ThreadPoolExecutor to process pages in parallel. This makes it scalable to handle dozens or hundreds of PDFs quickly. It also caches results internally so repeated operations aren't recomputed.

File Handling

The input PDFs are placed in a folder named /app/input. The script processes all PDFs in that folder and writes the output JSON files into /app/output. Each output file is named after the corresponding PDF.

Output Format

Each output JSON contains the extracted title and an outline which is a list of heading objects. Each heading includes its level (like H1), text, and the page number it was found on.

Error Handling

PDF Title and Outline Extractor - Technical Report

If a file or page cannot be processed, the script logs the error but continues processing the rest. This ensures that one corrupted file doesn't stop the entire batch process.

Conclusion

This script is a robust tool for automatically extracting key structural information from PDF documents. It is designed to be efficient, clean, and fault-tolerant, making it suitable for automated document analysis pipelines. Future improvements may include better hierarchy detection, OCR support for scanned PDFs, or a user interface for easier use.