PDF Table Extraction and Export to Excel

Overview

This script extracts tables from a PDF file and saves them as an Excel file. It uses `pdfplumber` to analyze the layout and text structure of the PDF to detect tables. The extracted tables are saved into an Excel workbook, with each table stored in a separate sheet.

Dependencies

- Python 3.x
- Required libraries:
  - `pdfplumber` (for extracting text and tables from PDFs)
  - `pandas` (for handling tabular data)
  - `openpyxl` (for writing data to Excel)
  - `tkinter` (for file selection dialogs)

Install the dependencies using:

```
pip install pdfplumber pandas openpyxl
```

How to Use

Run the script:

```
python script.py
```

A file selection dialog will appear:

1. Select a PDF file containing tables.
2. Choose an output location and filename for the Excel file.
3. The script will process the PDF and extract tables into the Excel file.

Functions Documentation

`detect_tables(page)`

Purpose

Detects tables on a given PDF page by analyzing the extracted words and their layout.

Parameters

- `page`: A `pdfplumber.page.Page` object representing a page in the PDF.

Returns

- `tables`: A list of detected tables, each table represented as a list of lists (rows of strings).

Logic

1. Extracts words from the page with `x_tolerance` and `y_tolerance` to handle different table structures.
2. Groups words by their `y` coordinates to form rows.
3. Sorts words within each row by `x` coordinate.
4. Ensures consistent column lengths by padding missing values.

`save_to_excel(tables, output_file)`

Purpose

Saves extracted tables into an Excel file, with each table stored as a separate sheet.

Parameters

- `tables`: A list of tables extracted from the PDF.
- `output_file`: Path to the output Excel file.

Logic

1. Uses `pandas.ExcelWriter` with `openpyxl` as the engine.
2. Converts each table into a Pandas DataFrame.
3. Writes each table to a separate sheet in the Excel file.

`extract_tables_from_pdf(pdf_path, output_excel_path)`

Purpose

Extracts tables from all pages of a given PDF file and saves them to an Excel file.

Parameters

- `pdf_path`: Path to the input PDF file.
- `output_excel_path`: Path to the output Excel file.

Logic

1. Opens the PDF file using `pdfplumber.open(pdf_path)`.
2. Iterates through all pages and applies `detect_tables()`.
3. Collects tables from all pages and calls `save_to_excel()` to write them to an Excel file.

`select_file()`

Purpose

Opens a file selection dialog to choose a PDF file.

Returns

- `file_path`: Path of the selected PDF file.

Logic

1. Uses `tkinter.filedialog.askopenfilename()` to open a file selection dialog.
2. Returns the selected file path.

`main()`

Purpose

Handles user interaction for selecting a PDF file and specifying the output Excel file.

Logic

1. Calls `select_file()` to get the PDF path.
2. If no file is selected, it exits.
3. Opens a save dialog using `asksaveasfilename()` to get the Excel file path.
4. Calls `extract_tables_from_pdf()` to process the PDF and save the extracted tables.

Example Output

If a PDF contains two tables, the output Excel file will have:

- `Table_1` in Sheet1
- `Table_2` in Sheet2

Limitations

- Might not handle highly irregular tables accurately.
- Tables without clear separations may be misidentified.
- Extraction quality depends on the PDF's formatting.

Future Improvements

- Improve detection for tables without clear borders.
- Support for multi-line cell content.
- Option to preview extracted tables before saving.

This script is a simple yet effective way to extract tabular data from PDFs and convert them into Excel for further processing.