**PDF Document Summarizer**

This project is a Python-based application designed to extract text from PDF documents, detect their language, and generate concise summaries. It supports both native (text-selectable) PDFs and scanned (image-based) PDFs by leveraging Optical Character Recognition (OCR) capabilities.

**Features**

* **Text Extraction:** Extracts text from PDF files.
* **OCR Support:** Automatically attempts OCR for scanned or image-only PDFs if native text extraction fails or is insufficient.
* **Language Detection:** Detects the language of the extracted text.
* **Summarization:** Generates a concise summary of the document using a pre-trained transformer model.
* **Multilingual (Experimental):** Attempts to translate non-English text to English before summarization (requires specific models and may have limitations depending on local setup).
* **Summary Saving:** Option to save the generated summary to a text file.

**Prerequisites**

Before running the application, ensure you have the following installed on your Windows system:

1. **Python 3.8+**: Download from [python.org](https://www.python.org/downloads/windows/).
2. **Git (Optional but Recommended)**: For cloning the repository. Download from [git-scm.com](https://git-scm.com/download/win).
3. **Poppler (for pdf2image)**: A PDF rendering utility.
   * Download the latest Poppler for Windows binaries from [oschwartz10612's GitHub releases](https://github.com/oschwartz10612/poppler-windows/releases). Look for a .zip file like Release-XX.YY.0-0.zip.
   * Extract the contents to a simple path, e.g., C:\Poppler\poppler-XX.YY.0\Library\bin. Note this path.
4. **Tesseract OCR (for pytesseract)**: An optical character recognition engine.
   * Download the Windows installer from [UB Mannheim's Tesseract page](https://digi.bib.uni-mannheim.de/tesseract/). Look for tesseract-ocr-w64-setup-X.X.X.exe.
   * Run the installer. **During installation, ensure you select "Add Tesseract to PATH"** and note the installation directory (e.g., C:\Program Files\Tesseract-OCR). Ensure English language data is selected.

**Setup Instructions**

1. **Clone the Repository (if applicable) or Download the Project Files:**
2. git clone <your-repository-url>
3. cd pdf\_summarizer\_project

(If you downloaded files, navigate to your pdf\_summarizer\_project directory in PowerShell.)

1. **Create a Python Virtual Environment:**
2. python -m venv venv
3. **Activate the Virtual Environment:**
4. .\venv\Scripts\activate

Your PowerShell prompt should now start with (venv).

1. **Install Required Python Libraries:**
2. pip install -r requirements.txt

(Ensure you have a requirements.txt file in your project containing PyPDF2, langdetect, transformers, torch, nltk, pytesseract, pdf2image, sentencepiece if you're attempting multilingual, and accelerate). If you don't have requirements.txt, run:

pip install PyPDF2 langdetect transformers torch nltk pytesseract pdf2image accelerate

(If you intend to use multilingual translation, you might also need sentencepiece, but be aware of the persistent environment issues we faced with it.)

1. **Download NLTK Data:** After installing nltk, you need to download the punkt tokenizer data. Add this to your main.py or run it once in Python:
2. import nltk
3. try:
4. nltk.data.find('tokenizers/punkt')
5. except nltk.downloader.DownloadError:
6. nltk.download('punkt')

**Configuration**

**Crucially, you need to update main.py with the paths to your Poppler and Tesseract installations.**

1. Open your main.py file in a text editor.
2. Locate the --- Configuration for Poppler and Tesseract --- section at the top.
3. **Update GLOBAL\_POPPLER\_PATH**: Replace the placeholder with the exact path to Poppler's bin folder you noted during installation (e.g., C:\Poppler\poppler-24.08.0\Library\bin).
4. GLOBAL\_POPPLER\_PATH = r"C:\Users\shaha\Downloads\Release-24.08.0-0\poppler-24.08.0\Library\bin" # YOUR ACTUAL PATH
5. **Update GLOBAL\_TESSERACT\_CMD**: Replace the placeholder with the exact full path to the tesseract.exe file (e.g., C:\Program Files\Tesseract-OCR\tesseract.exe).
6. GLOBAL\_TESSERACT\_CMD = r"C:\Program Files\Tesseract-OCR\tesseract.exe" # YOUR ACTUAL PATH
7. **Save the main.py file.**

**Usage**

1. **Open PowerShell** and navigate to your pdf\_summarizer\_project directory.
2. **Activate your virtual environment:**
3. .\venv\Scripts\activate
4. **Set the Tesseract data path for the session:** (This is needed because pytesseract needs to find the language data.)
5. $env:TESSDATA\_PREFIX="C:\Program Files\Tesseract-OCR\tessdata"

(Adjust path if your tessdata folder is not in C:\Program Files\Tesseract-OCR\tessdata).

1. **Run the application:**
2. python main.py
3. Follow the prompts to enter the path to your PDF file. The application will process it and display the summary. You will then be asked if you want to save the summary to a file.

**Notes on Multilingual Summarization**

The application includes a DocumentTranslator class intended for translating non-English documents to English before summarization. However, due to persistent environment issues related to the sentencepiece library in this specific setup, this functionality may not work reliably. If the translation model fails to load, the application will proceed to summarize the original (non-English) text, which may result in less accurate or meaningful summaries.

For robust multilingual support without external APIs, a Linux environment (e.g., via WSL2 on Windows) is generally recommended due to better compatibility with certain NLP libraries.

**Future Improvements**

* **Robust Multilingual Support:** Address the sentencepiece issue for more reliable non-English document summarization.
* **GUI (Graphical User Interface):** Develop a user-friendly interface for easier interaction, replacing the command-line prompts.
* **Batch Processing:** Add functionality to process multiple PDF files at once.
* **Customizable Summary Length:** Allow users to specify desired summary length more precisely.
* **Error Handling:** Enhance error reporting and graceful failure for various PDF issues.
* **Advanced OCR Preprocessing:** Implement image preprocessing steps (e.g., deskewing, denoising) to improve OCR accuracy on challenging scanned documents.