

Intelligent Bibliography Creation System

Introduction

The Intelligent Bibliography Creation System automates the process of creating bibliographies, reducing manual effort and enhancing efficiency. It reads bibliographic data from a PDF file, writes it to an Excel sheet, and inputs the data into the library's backend system.

Features

- **PDF Data Extraction:** Automatically extracts bibliographic information from specified PDF files.
- **Excel Generation:** Writes extracted data into an Excel sheet, with each bibliography as a row.
- **Automated Input:** Opens a browser and inputs data from Excel into the library's backend system.

Usage

1. **Place PDF:** Put the PDF file in the designated folder (e.g., input/).
2. **Run the Program:** Execute the main program in the command line.
3. **Input Bibliography Numbers:** The system will prompt for bibliography numbers or ranges (e.g., 1 for the first bibliography or 1-10 for multiple entries).
4. **Automatic Input:** The system opens the browser and fills in the bibliographic data.
5. **Verify:** Check the library's backend to confirm successful entries. Rerun the program if corrections are needed.

System Architecture

- **Frontend:** Command Line Interface (CLI) for user inputs.
- **Backend:** Python for data extraction and automation.

- **Data Processing:** Uses pdfplumber for PDF parsing, openpyxl for Excel operations, and Selenium for browser automation.

Technical Details

- **Python:** Main programming language for development.
- **pdfplumber:** Extracts text from PDFs.
- **openpyxl:** Reads and writes Excel files.
- **pandas:** Processes and cleans data.
- **Selenium:** Automates browser interactions.

Workflow

1. **Extract Data:** Reads the PDF and cleans data.
2. **Generate Excel:** Creates an Excel file with bibliographic data.
3. **Validate Data:** Ensures data completeness and prompts for corrections if needed.
4. **Automated Input:** Inputs bibliographic data into the backend.
5. **Error Handling:** Logs failures and prompts for retries.

Future Plans

- Support for multiple PDF formats.
- Automatic data validation enhancements.
- Integration with more input platforms.

References

- Selenium Documentation
- pandas Tutorials
- openpyxl Documentation
- pdfplumber Documentation

Contributors

- **Developer:** Shiuan-Jen Yang