Phase 2 Report

Insightify

Group F(original assigned group)

Bingchen Yang

249552780

Table of Contents

Table of Contents....2
1. Introduction....3
2. Features Implemented....3
3. Class Diagram....4
4. Sequence Diagram....4
5. Individual Contribution....5
Bingchen Yang....5
6. Team Leader Role....6
7. Bugs, Issues, and Future Implementation....7
8. Presentation Video and Slides....9

1. Introduction

- Project Overview: Insightify is a tool designed to extract and organize valuable information from PDF documents efficiently.
- Current Phase: This phase focuses on implementing core functionalities, resolving bugs, and planning for future enhancements.

2. Features Implemented

2.1 PDF Read and Parsing:

- Open and read PDF files using Fitz (PyMuPDF).
- Parse and extract meaningful content such as text and images from PDF pages.

2.2 Text Extraction:

- Extract text from PDF pages using Fitz (PyMuPDF).

2.3 Image Extraction:

- Extract images from PDF pages using Fitz (PyMuPDF).

2.4 Table Extraction:

- Extract tables from PDF documents using Camelot and save them as CSV files.

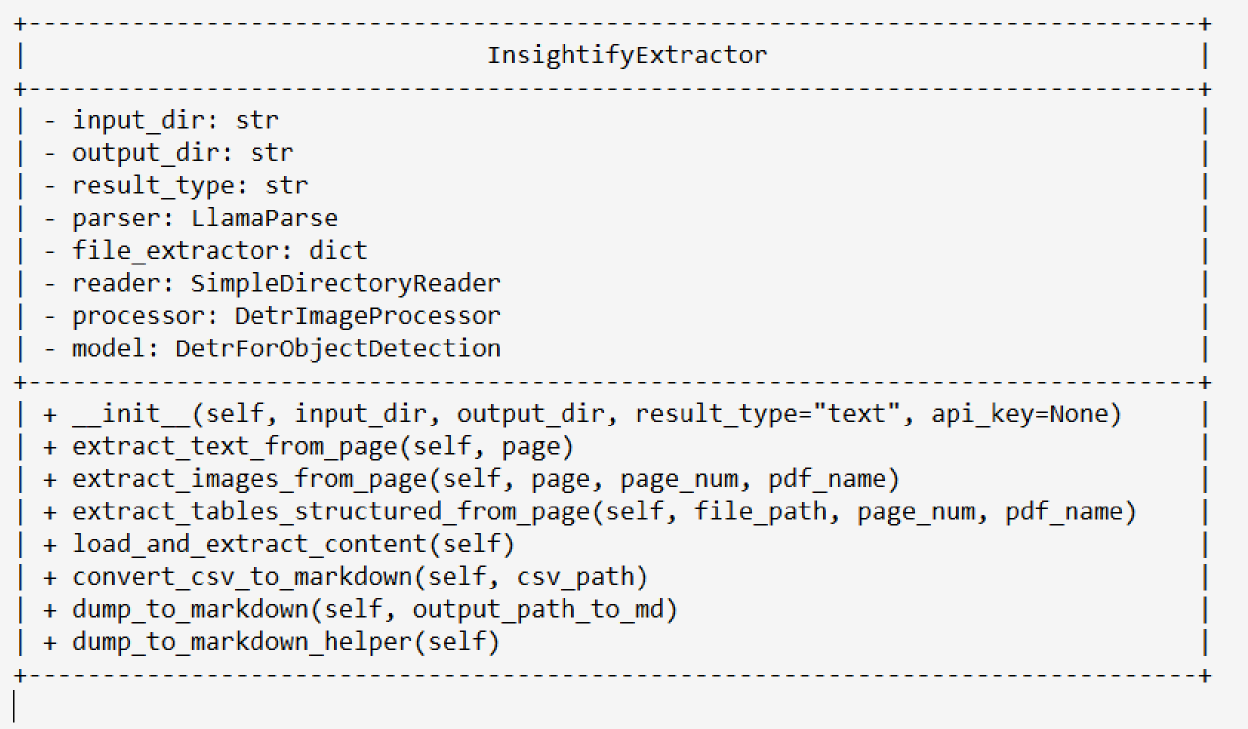
2.5 Object Detection:

- Utilize the DEtection TRansformers (DETR) model from the Transformers library for object detection within images.

2.6 Markdown File Generation:

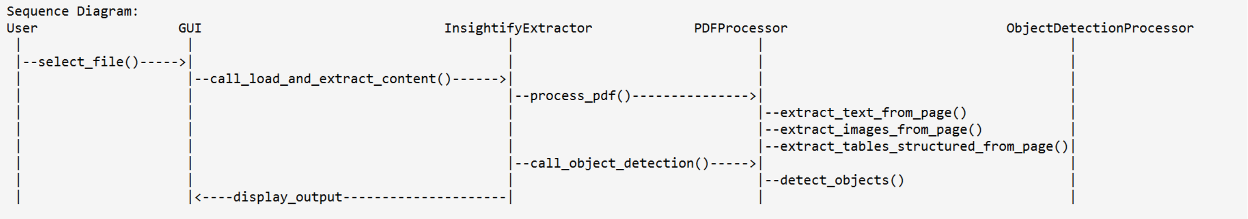
- Convert extracted content into a well-structured markdown file for easy access and readability.

3. Class Diagram



```
+-------------------------------------------------------------------------------+
|                               InsightifyExtractor                             |
+-------------------------------------------------------------------------------+
| - input_dir: str                                                              |
| - output_dir: str                                                             |
| - result_type: str                                                            |
| - parser: LlamaParse                                                          |
| - file_extractor: dict                                                        |
| - reader: SimpleDirectoryReader                                               |
| - processor: DetrImageProcessor                                               |
| - model: DetrForObjectDetection                                               |
+-------------------------------------------------------------------------------+
| + __init__(self, input_dir, output_dir, result_type="text", api_key=None)     |
| + extract_text_from_page(self, page)                                          |
| + extract_images_from_page(self, page, page_num, pdf_name)                    |
| + extract_tables_structured_from_page(self, file_path, page_num, pdf_name)    |
| + load_and_extract_content(self)                                              |
| + convert_csv_to_markdown(self, csv_path)                                     |
| + dump_to_markdown(self, output_path_to_md)                                   |
| + dump_to_markdown_helper(self)                                               |
+-------------------------------------------------------------------------------+
|
```

4. Sequence Diagram



```
Sequence Diagram:
User                  GUI                                InsightifyExtractor          PDFProcessor                             ObjectDetectionProcessor
 |                     |                                        |                               |                                       |
 |--select_file()----->|                                        |                               |                                       |
 |                     |--call_load_and_extract_content()------>|                               |                                       |
 |                     |                                        |--process_pdf()---------------->|                                       |
 |                     |                                        |                               |--extract_text_from_page()             |
 |                     |                                        |                               |--extract_images_from_page()           |
 |                     |                                        |                               |--extract_tables_structured_from_page()|
 |                     |                                        |--call_object_detection()----->|                                       |
 |                     |                                        |                               |--detect_objects()                     |
 |                     |<----display_output---------------------|                               |                                       |
```

5. Individual Contribution

Bingchen Yang

- Project Initiation:
  - Set up the initial project structure and GitHub repository.
  - Configured the PyCharm environment for seamless development and version control integration.
- Core Development:
  - Implemented the InsightifyExtractor class to handle text extraction, image extraction, table extraction, and object detection.
  - Integrated various libraries including Fitz, Camelot, Transformers, and LlamaParse to facilitate robust PDF parsing and content extraction.
- Bug Fixes:
  - Identified and resolved issues related to temporary memory handling, ensuring that PDF pages are properly closed to avoid PermissionError.
  - Debugged and optimized the interaction between Fitz and Transformers to prevent access conflicts.
- Feature Development:
  - Developed a user-friendly GUI for file selection using tkinter.
  - Created functionality to convert extracted content into markdown files for easy readability and accessibility.
- Future Planning:
  - Outlined plans for OCR integration using Tesseract for handling scanned PDFs.
  - Proposed enhancements to use AI models for interpreting structured data and images.
  - Planned the transition from markdown output to PDF format, leveraging libraries like ReportLab.
- Documentation:
  - Documented the entire development process, including code comments, usage instructions, and detailed reports on progress and future goals.

6. Team Leader Role

Task Assignment and Coordination:

- Assigned tasks to team members both during class and after class to ensure continuous progress.
- Set up the GitHub repository and invited team members to collaborate.
- Taught team members how to use PyCharm and Git for version control.
- Created a Google Drive folder for non-code-related tasks such as documentation and presentations, and invited team members to collaborate.

Support and Mentorship:

- Offered assistance to team members facing challenges with setting up their workflows and working environments.
- Provided guidance on coding-related issues to ensure effective contributions from all team members.

Weekly Check-Ins:

- Organized and led weekly check-in meetings with the team to discuss progress, challenges, and next steps.
- Used these meetings to provide updates, gather feedback, and ensure that all team members were aligned with the project's objectives.

7. Bugs, Issues, and Future Implementation

7.1 Object Detection Integration:

- Description: Integration of Hugging Face's DEtection TRansformers (DETR) model for object detection tasks within images.
- Solution: Implemented object detection using DETR model and ensured compatibility with other extraction methods.
- Status: Fixed

7.2 Structured Data Extraction:

- Attempts: We attempted to resolve issues related to extracting structured data, specifically tables, from PDF documents.
    - Attempt 1: Tried using LlamaParse for structured data extraction but found it unsuitable as it does not handle structured data well.
    - Attempt 2: Moved to using Camelot to extract tables directly from the PDF.
- Status: Fixed

7.3 Rendering Extracted Structured Data to Markdown:

- Description: Rendered extracted structured data in CSV format to a markdown file.
- Solution: Implemented a function to convert CSV data into markdown table format.
- Status: Resolved

7.4 Temporary Memory Issue:

- Description: PDF pages were not being closed properly by the time the program terminated, resulting in permission errors.
- Solution: Ensured that PDF documents are explicitly closed using the .close() method.
- Status: Unresolved

7.5 Conflict with Hugging Face Transformer:

- Description: A conflict arose when both Fitz and the Transformer model tried to access the same resources in temporary memory, causing a PermissionError.
- Solution: Proper file handling and separation of tasks were suggested to avoid conflicts. However, further debugging and implementation are required.
- Status: Unresolved

7.6 Scanned Content Handling:

- Description: Planned implementation of features for scanned content using the Hugging Face transformer. Integration is planned for OCR using Tesseract to handle scanned PDFs.
- Solution: Use Tesseract for OCR to handle scanned PDFs separately from Fitz and Transformers.

- Status: Planned for Future Implementation

7.7 Generating PDF Output:

- Description: Transitioning from markdown file generation to PDF format while maintaining readability and accessibility.
- Solution: Utilize libraries like ReportLab to generate formatted PDF reports.
- Status: Planned for Future Implementation

8. Presentation Video and Slides

Presentation Slides: Deliverable 1 Presentation
Presentation Video: Presentation_Feb_12th.mp4
Github Repository: Insightify