University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2019

PDF Extraction and Cleanup

Sarah Bondurant, Nathan Davis, Richard Mays, Keegan Riley, Hayden Willeford

Abstract

Due to their consistency across software and devices, PDF files are one of the most used file formats in today's academic world. PDF files are easily readable to humans, but machines can struggle to interpret the text. This project will use Natural Language Processing (NLP) and computer vision to restructure the text into a more readable format for the machine making PDFs even easier to use.

1.0 Problem

PDF files are one of the most used file formats in the modern world. Most online academic resources come in PDF form due to their consistency across different devices and software. They are easily readable by the user, and overall they look clean and professional. However, unlike their user-friendliness with humans, machines struggle to interpret them. This causes issues when machines try to read PDFs and creates inconsistencies in machine's interpretation of them.

Parsing a PDF leads to unexpected results, ranging from mis-ordering text in columns to incorrect formatting. Correct parsing of columns would read down one column then down the next. However, with PDFs, sometimes text is read across the columns, garbling the data up. With math formulas in PDFs, equations that involve subscripts and superscripts might see them moved around to an incorrect position. Also, tables or other graphics can be placed in completely incorrect sections of text. All of these issues lead to incorrect documents in a corpus, causing untold complications for customers and the provider alike.

2.0 Objective

The objective of this project is to use Natural Language Processing (NLP) and computer vision to restructure a PDF's text into a more easily readable structure for machines to interpret.

3.0 Background

3.1 Key Concepts

Two key technologies that are related to the problem and are essential to the development of the solution are NLP and computer vision. NLP, or Natural Language Processing, was developed to aid computers to understand the user's natural language. This being said, it is not an easy task to teach machines to be able to do this. NLP is a branch of AI that uses machine learning to take in input and learn how a language is used then being able to replicate it. Computer vision is a field that focuses on how computers are able to gain a high-level understanding from digital images or videos. Using both of these technologies, we could use machine learning to "teach" a computer to use its computer vision to run through documents and then use its NLP to then be able to extract and clean up the PDF.

Some tools that we would like to include would be modeled after the PyPDF2 tool that Python has for manipulating and using PDFs. These tools we wish to implement are the extraction of the document’s information (which is the main goal), rotating pages, merging PDFs, splitting PDFs, adding watermarks and encrypting a PDF if one so chooses. While we wish to include all of these tools, our main goal would be to make sure that we are able to extract the data from the document and therefore store it in the XML or other compatible document file to then output it to the desired file type.

3.2 Related Work

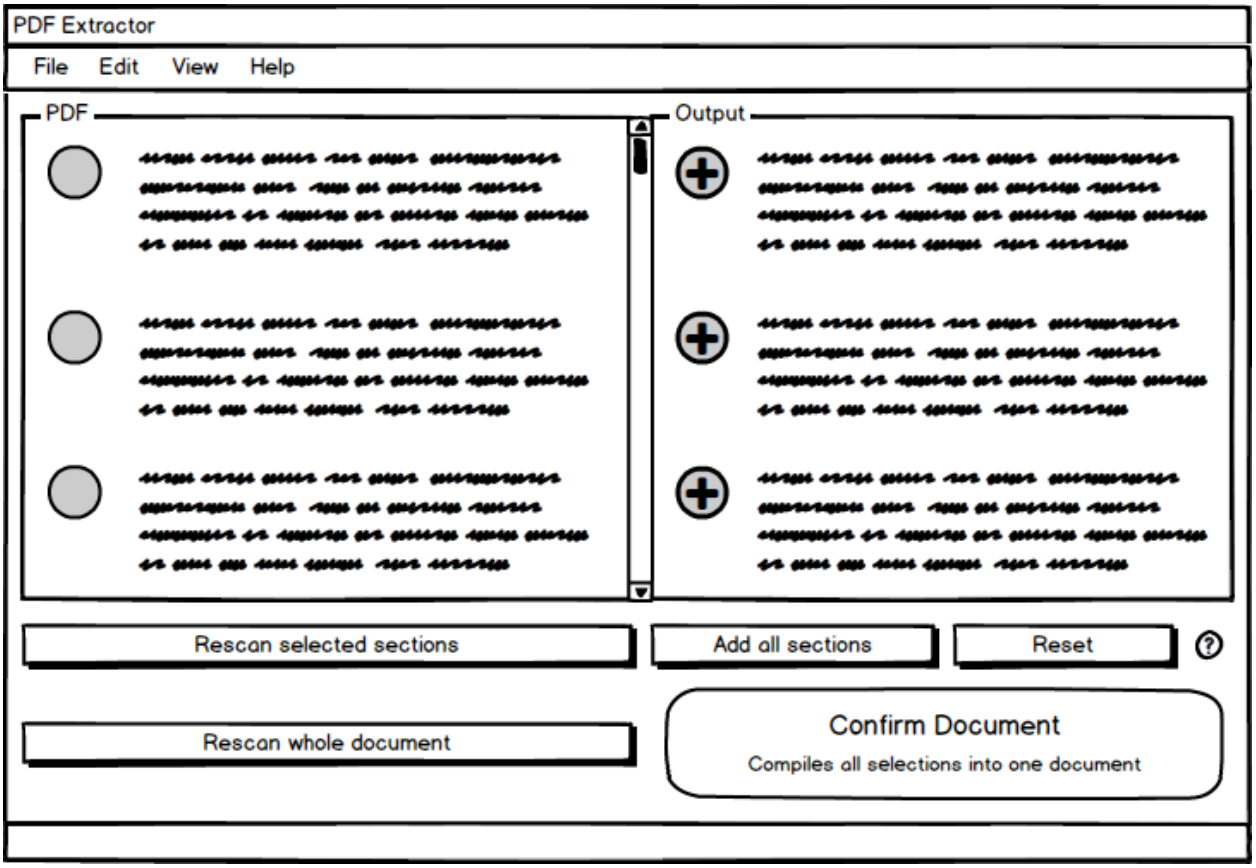
One source that has accomplished some aspects of this project is called ScienceBeam created by elifesciences. In this example, they used computer vision to teach their program to understand and create XML files from a PDF. The issues with their implementation is that the program can only convert the PDF to an XML and that XML is the only file type that the software was taught through machine learning. We hope to incorporate more options for the output file when our final project is completed. The ScienceBeam project also is still collaborating with other publishers to gather more valid PDF/XML pairs to help train their neural networks since it is hard to find raw author-submitted PDFs with complete XML. [1]

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

- Given a corpus, our program will reconstruct said corpus into machine-readable text
- No data is lost in document translation
- Formatting remains consistent with original design
- Execution takes a reasonable amount of time
- Program can take in multiple files to produce output
- User interface is clean and easy to use

4.2 High Level Architecture



Our project will consist of a PDF extractor program.

- UI Design:
    - Upon opening the program, the user will be prompted to upload a PDF.
    - The PDF will be scanned both for words as well as any images that may be present in the PDF and contain text. Words in images will be read using OCR. Text will attempt to be read as-is, if possible.
    - Once the document is scanned, both the original PDF and the output document will be displayed side-by-side.
    - The user will be able to select the sections to be included in the final document. The user can also select the selections to be rescanned, or rescan the whole document. On these rescans, the program will attempt different weights for combining text scraping, OCR output, and NLP.
    - Once the user has selected everything for the final document, the user can export the document. Output format will default to .docx but may be applied to other formats, such as .txt.

- PDF Scanning Code:
    - For simple text, the program will extract the text and attempt to maintain accurate formatting to the source material. For images of text, the program will use OCR to extract the text.
    - We may attempt multiple solutions involving different mixes of text scraping, OCR, NLP, and pre-existing utilities created for this problem.
        - One approach to NLP is implementing word sequence tasks that will attempt to generate and predict text representation

4.3 Risks

Risk	Risk Reduction
Potential data loss in conversion	Extensive use testing with a wide variety of corpus. Account for all formatting design structures.
Potential output containing incorrect data	Extensive use testing with a wide variety of corpus. Ensure language processing is accurate.
PDF input files not accepted/error	Ensure file type inclusion, and allow for multiple files

4.4 Tasks

1. Gain background knowledge in NLP/Computer Vision and how it relates to PDF's.
2. Understand how project fits into larger Sorcero workflow.
3. Brainstorm plans of attack for pulling text from PDFs.
4. Start working on 1 solution as a time
5. UI Design of program
6. Testing with various documents from exported PDFs to printed and scanned documents
7. Write documentation for the source code
8. Write user guide for the end user

4.5 Schedule

Tasks	Dates
1. Background knowledge of NLP/ Computer Vision	1/13 - 1/18
2. Sorcero workflow information	1/15 - 1/20
3. Brainstorm possible plans of attack	1/20 - 2/1
4. Begin work on 1st solution	2/3 - 2/15
5. Begin work on 2nd solution	2/17 - 2/29
6. Continue work on later solutions	3/2 - 3/14 (Variable)
7. UI Design & Useability	3/16 - 3/28
8. Testing	3/30 - 4/11
9. Documentation Write-Up	4/13 - 4/18
10. Create User Guide	4/20 - 4/ 25

4.6 Deliverables

- Design Document: Documentation for usage and properties of converter. Features and considerations will be added into this document.
- Web site: Information page for that houses source code and documentation for the project.
- Source code for converter program
- Final Report

5.0 Key Personnel

Sarah Bondurant – Bondurant is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed Programming Paradigms and Software Engineering. Over the past summer, she worked in web development. She will be responsible for UI design and the user guide.

Nathan Davis – Davis is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms and Software Engineering. He will be responsible for testing and documentation.

Richard Mays – Mays is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms, Software Engineering, and Computer Architecture. He will be responsible for creation of the initial prototype and the first prototype update.

Keegan Riley – Riley is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Programming Paradigms and Software Engineering. He will be responsible for the second and third prototype updates.

Hayden Willeford – Willeford is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering. He will be responsible for generalizing the prototype for different fonts.

Sorcero – Sorcero is building an Enterprise NLP suite that supports the Life Sciences & Insurance industries in handling complex information, policies, rules, and regulations, automating the building of incredibly smart knowledge bases and workflows that rely on technical language.

6.0 Facilities and Equipment

Some form of vision AI will be required for this project. Google's Cloud Vision API is able to perform Optical Character Recognition (OCR) on text within an image, which will be useful for scanned PDFs. Their API charges per image, where each page of a PDF is counted as an individual image. For less than 1000 units, their services are free.

7.0 References

[1] ScienceBeam - using computer vision to extract PDF data, https://elifesciences.org/labs/5b56aff6/sciencebeam-using-computer-vision-to-extract-pdf-data