Eric Crawford
CS410
Text Information Systems Technology Review

When we want to share rich text information with others, we have many choices of programs to use and formats to save to. One of the most prevailing formats that is supported by many text editors is Adobe's PDF. The PDF format was created by Adobe in 1991, and within a couple of years became the defacto standard for sharing immutable documents. Its success was further acknowledged by being incorporated into the International Organization for Standardization. Even though PDFs were originally meant to be used to create documents that could not be changed or converted, the recent rise in fields such as text mining and text analytics have created a high demand for software and services that can extract text information out of PDFs. Today there are a myriad of tools and services that can extract plain text from PDFs, but they have all pretty much failed at retaining any rich information such as text size, font family, text location in the document, etc. I plan to explore two brand new services that advertises their ability to pull out rich textual information, along with the text that is extracted from PDFs. The first of these services will be Textract by AWS and the second is PDF Extract API by Adobe.

One of the successes of PDFs is its ability to encapsulate text from another program, such as Microsoft Word, and its ability to also encapsulate existing written documents via being scanned from a printer or scanner. I will be testing with two different PDF documents that are representative of the two use cases. The first document is a PDF version of the Affordable HealthCare Act (AHCA) (see appendix), which represents a document that was created from another program and has been encapsulated into a PDF. The second is a Crew Training Manual from NASA (NASA), which represents a document that was scanned into digital form and converted to PDF (see appendix). This will allow me to conduct an accurate analysis of the results on potential real pdf text extraction situations. I will compare the costs, experience, and results of both services and give my analysis on pros and cons for each service.

Textract was released as a developer preview in 2018 by AWS. AWS is the number one cloud service provider and has one of the biggest suite of services that companies, and individual developers can use to move or expand their backend to the cloud. Up until 2018, there was very little machine learning services offered by AWS. At their "RE-Invent" conference in 2018, that changed with the release of a suite of machine learning tools which included AWS Textract. The service has a basic cost of $0.0015 for each page of text extraction, but also offers other extraction types like forms, tables, and expenses from representative documents, at a higher cost per page. The service can be interacted with via an SDK for almost all the of the popular programming languages (java, python C# … etc.), or the AWS CLI interface, or within the AWS console for quick exploration of small documents.

Since this was exploratory work, I used the AWS console. Even though the service can handle thousands of pages, the console has a restriction of 11 pages max. This is because it's an interactive interface that shows you the results of the extraction in real time displays bound

boxes around objects it believes are words, sentences, or forms. Because of the page limit, I could only submit a smaller subset of both documents. The experience of analyzing a document in the console was as simple as uploading the file and clicking the analyze button. The preview of the results seemed to be quite accurate. I was then able to download a zip containing all the info that was extracted. The zip contained a file with the raw extracted text, several csv files containing any data that was in a tabular format, and a json formatted file with the additional pdf specific metadata. This metadata contained important things like:

- IDs for each text item it found
- Page number and geometry where the text was located
- Whether it was a whole line of text or a single word
- Confidence score on how accurate it thinks of the bullet point above
- A search key that could be used to find this word in a text editor

Due to the max limit, I used pages 11 – 20 of the AHCA, the first 10 pages of the NASA crew training manual, and pages 59-60(pdf pages) of the NASA document. The last set of pages was tested because it had images and columnar layout. All the raw text was successfully extracted from each document, but what was impressive was that Textract was also able to extract the handwritten words from the NASA documents. For example, the word "Parker" at the top right corner of the first page, and the handwritten text under the "Index Data" section on the first page. Both entries showed up perfectly in the raw text document. The big difference between the two extractions was Textract was pretty good at keeping the raw text of the AHCA in the same structure, while it had a little trouble doing the same for the NASA document. Textract also provided a separate file that contained the PDF metadata for each extraction. As expected, it had pretty much high confidence on each word and line it found and correctly tagged the words I checked. I had successfully done the same for the NASA document, even going as far as to tag the "Parker" word as "HANDWRITING" text. Another surprise was how well it detected text that was in a columnar format. It had created several csv files detailing these.

Adobe PDF Extract API is a relatively new service. Many will find it weird that it took so long for the creators of the PDF format to come out with a service this sophisticated. While Adobe has always had tools users could use to extract text from PDFs, that was the limit of it, only text. With this new service, Adobe has incorporated machine learning, which they call Sensi, to help extract additional information and context about the text itself. This service is provided as part of the Adobe I/O cloud suite of tools. It's priced at $0.05 per Document transaction, which seems cheaper than AWS Textract. However, the service is only offered as SDK's to be added into an existing codebase, and the number of supported programming languages is limited to only javascript, .Net, Java, Python.

Getting set up with a project for Adobe PDF Extract is simple and the process generates an entire project in your selected programming language, with examples of the many services Adobe offers. I used the Java examples to test out the services. This service has a 100-page limitation during the trial I was in, so I also did not use the entire AHCA or NASA document here either. The process of doing an extraction was straight forward and just a few lines of code.

After running the program, a json formatted file is created with all the text and meta data found for each extracted text. This file contained very rich information such as:

- Text Size
- The language of the text (English in this case)
- Page number where text is located
- Font family and style of the text

Most of the results were of entire sentences on one line. Very rarely did it give a result that was just one word. One interesting observation is that this service did not extract the handwritten text at all from the NASA documents, which is surprising. Adobe should have been entirely capable of performing this function.

Both services performed really well and have their own strengths and weaknesses. AWS Textract was good at extracting text from multiple formats (not just PDFs), it could extract handwritten text and it had several ways to use the service. However, it could not extract contextual information about the text such as color, size, and style. Adobe PDF Extract was able to give us that contextual information but that was pretty much its only benefit over AWS Textract. It did not offer columnar data awareness, could not detect handwritten text, and you could only use the service if you were a developer and knew one of the supported languages. While both services are good at extracting text, at this time AWS Textract would give more information that would be valuable in many text mining and text analytics operations.

Appendix:

Affordable HealthCare Act (aka: AHCA)
http://www.gpo.gov/fdsys/pkg/PLAW-111publ148/pdf/PLAW-111publ148.pdf

NASA Crew Training Manual (aka: NASA)
https://www.hq.nasa.gov/alsj/HSI-481184-LCRU.pdf