Challenges of Optical Character Recognition

Orville “El” Anderson
and10393@umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris Minnesota USA

Abstract

Optical Character Recognition (OCR) is technology used to extract text from images. OCR has three main weaknesses when applied to scanned documents stemming from page layouts, the alphabet used, and visual noise. By intentionally expanding the documents we use to train modern OCR models, we can increase [usability] of this technology. This paper looks at the Tesseract, Amazon Textract, and Google Document AI models, as applied to English and Arabic documents.

Keywords: optical character recognition, scanned documents, layout, languages, visual noise, datasets

1 Introduction

Over time, American Institutions, have accumulated a ginormous amount of scanned documents. In April of 2024, the Department of Justice, updated the Americans with Disabilities Act to include access to digital content such as scanned documents. Among other things, all scanned documents made publicly accessible by state and local governments must now be usable by a screen reader. When a document is scanned, it becomes an image and looses all information about the text present. The first step to make a scanned document screen-readable is to recognize the text lost when the document was scanned. This process can be done manually, but that isn’t well suited for large numbers of documents. Instead we look to Optical Character Recognition (OCR), a technology made to extract text from images.

the first step is to consider if the page is necessary, the second is to try to find the source document

This paper looks at how OCR works, specifically for scanned documents, and looks at some of the specific weaknesses in applying OCR in the context of Public Universities.

2 Background

The process of OCR starts with an existing scanned document. These documents can come in many file types, some common ones are .tiff, .pdf, and .jpg. This file is then input to an OCR model, such as Tesseract, Amazon’s Textract, or Google’s Document AI, where the model will go through the three stages of OCR. The model will then return the text identified in the document as plain text, sometimes with meta-information. [1]

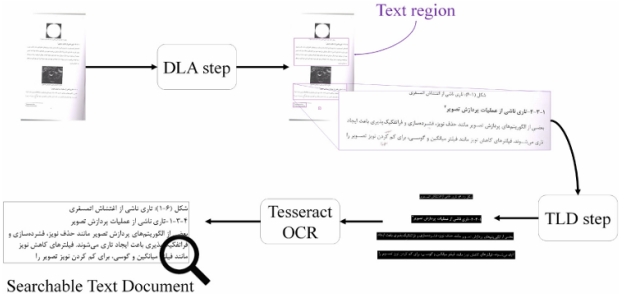

Figure 1: stages of OCR

Pages to Reference[1, 5]

2.1 Document Layout Analysis

The first step in OCR is Document Layout Analysis (DLA), and is a general pre-processing step. The purpose is to identify what part of the image is text and what is not. This effectively draws a box around each paragraph and table on the page. This step frequently outputs the result as a binary image, where each pixel is marked as a text or non-text pixel, to reduce computation costs.

2.2 Text Line Detection

Text Line Detection (TLD) is the second step in OCR. TLD takes the blocks of text from the previous step and further breaks them down, into lines, words, and then individual characters. The output of this step is each identified character in its own defined box of pixels.

A common technique used in this step includes rotating the individual lines of text to create a baseline, so counter any rotated text. This extra action improves the accuracy of identified characters. Two methods employed by Fateh et al[2] used font size and correcting line rotation, to improve overall accuracy.

2.3 Recognition

The final step in the OCR process is referred to as Classification or Recognition. This step takes the boxes of individual

[1]Modern OCR models can return use this information to make differnt output formats, but they all use the text and meta-information to do that.

characters from the last step and tries to identify the character inside of them.

One common technique is to compare the unknown character to a set of known characters, overlaying them, and seeing which ones are the most similar. Another technique is to identify features from the character to make an educated guess.[5][2]

2.4 Comparison

The main considerations when judging OCR models are accuracy and speed.[1]

A popular way to measure the accuracy of OCR output is to run the OCR program on a document where the page content is know. The OCR output is then compared to the known content and is measured by [the formula below], where x is a unit of measurement, like a line, word, or character.

$$\frac{\sum_{i=1}^{\text{num of pages}} \text{number of x correctly indentified}}{\sum_{i=1}^{\text{num of pages}} \text{number of x on the page}}$$

3 Challenges

There are three main categories of things that make scanned documents harder to digitize. Each of these challenges tie back to the key concept that OCR models work best when applied to what they were made to recognize.

3.1 Layout

Documents come in many different layouts. Things like images, figures, and number of columns add a layer of complexity to documents. In the Document Layout Analysis step of OCR the model must know which parts of the page to ignore, but also to know what order the sections of text should go in. A paper formatted with two columns, such as this, is meant to be read left column, then right column. Unless otherwise instructed, an OCR model will take the first line from each column and treat them as one line.

There exists OCR specifically trained to handle documents like forms. For example a model can be trained to specifically digitize job applications for a specific company, or one specific tax form. By making a specialized model, there is an increased accuracy when working with documents of that specific layout, but a decrease in accuracy for other layouts.

One related challenge to OCR accuracy is curved lines of text. The DLA and TLD stages of OCR[3] use rectangles to section off portions of text and do not automatically rotates lines/words/characters to be [horizontal?]. Identifying a character when it is rotated is less likely to be accurate then if the character was horizontal. [2]

[2]Objection, how is this relevant?? Wouldn't it be more important to talk about training for recognition?

[3]don't you dare use another acronym in this paragraph

3.2 Alphabet

The English alphabet consists of 26 letters, each with a lowercase and uppercase variant. English is written left-to-right and is primarily non-cursive. Arabic, in comparison is unicase, uses contextual forms, is written right-to-left, and is cursive. Arabic also uses diacritics, dots and marks, which can appear above or below the main text and influence the meaning.

These two languages are important when discussing OCR, both because of their popularity, but also because of how different they are. As this paper[2] says about Arabic, "Finally, character and TLD approaches cannot be universally applied to scripts with different languages. This challenge is particularly pronounced in languages such as Persian and Arabic, where text characters can take the form of connectors or non-connectors. Connector letters are affixed to both pre- and post-letters to form words, and diacritic marks are commonly used in Arabic text—both of which can introduce complexity to TLD techniques." This paper explored the effects of changing the size of boxes when isolating characters in Arabic documents. They found that by increasing the size of the boxes, they had a higher accuracy.

I introduce a lot of concepts here (like reading direction) that I don't do anything with. I need to touch on them, or remove them

This weakness in OCR models is most easily seen in non-Latin language documents, but can also be seen when using these models on documents with a variety of fonts, or documents with handwritten text.

3.3 Visual Noise

One big factor in the accuracy of OCR is the quality of the initial image. Marks on the physical document, book spines, and low image resolution all add additional complexity to the process.

4 Results

To evaluate accuracy and compare OCR models we use bench-marking datasets, collections of images where the expected output is known. Because of the inherit variety of documents needed to target these issues there is not really a reasonable way to make one. Some specialized data-sets exist [2, 3]

Fateh et Al[2] looks at TLD for Arabic text and found that increasing the size of the boxes drawn around each character increased OCR output accuracy(for the model(s) they tested)

Thomas Hegghamer created an English and Arabic dataset specifically to test OCR accuracy of documents with artificial noise applied. "functions to generate six ideal types of image noise: "blur," "weak ink," "salt and pepper," "watermark," "scribbles," and "ink stains" (see Fig. 2c-h). While not an exhaustive list of possible noise types, they represent several of the most common ones found in historical document

scans.”[3] Hegghamer’s ”Noisy OCR Dataset”, consists of 422 original documents with 43 variations of each, for a total of 18,568 documents.

5 Conclusion

If the end goal of Optical Character Recognition for documents is to make one model that can be used on all documents to ever exist, it makes sense to have an all-encompassing weakness-addressing data set. As it stands, that data set can not exist. Ignoring the obvious storage and resource considerations, for the reasons outlined in this paper, we can not begin to comprehend the number of variations in documents, let alone truly test for them all.

A universal data set would not be widely accepted because it would actively go against many popular forms of OCR, specialized layout OCR models. While not generally applicable, due to their specialized nature, these models have a place in OCR conversations.

All that said, I think that ideal data set is worth perusing. Progress towards a goal we may never meet is still progress. By pushing the bounds of what OCR models can do, we can strengthen their current capabilities.

dataset or data set?

Acknowledgments

Thanks.

References

[1] Ridvy Avyodri, Samuel Lukas, and Hendra Tjahyadi. 2022. Optical Character Recognition (OCR) for Text Recognition and its Post-Processing Method: A Literature Review. In *2022 1st International Conference on Technology Innovation and Its Applications (ICTIIA)*. 1–6. doi:10.1109/ICTIIA54654.2022.9935961

[2] Amirreza Fateh, Mansoor Fateh, and Vahid Abolghasemi. 2024. Enhancing optical character recognition: Efficient techniques for document layout analysis and text line detection. *Engineering Reports* 6, 9 (2024), e12832. doi:10.1002/eng2.12832

[3] Thomas Hegghammer. 2022. OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment. *Journal of Computational Social Science* 5 (05 2022). doi:10.1007/s42001-021-00149-1

[4] Ravi Raj and Andrzej Kos. 2022. A Comprehensive Study of Optical Character Recognition. In *2022 29th International Conference on Mixed Design of Integrated Circuits and System (MIXDES)*. 151–154. doi:10.23919/MIXDES55591.2022.9837974

[5] Chhanam Thorat, Aishwarya Bhat, Padmaja Sawant, Isha Bartakke, and Swati Shirsath. 2022. A Detailed Review on Text Extraction Using Optical Character Recognition. In *ICT Analysis and Applications*, Simon Fong, Nilanjan Dey, and Amit Joshi (Eds.). Springer Nature Singapore, Singapore, 719–728.