Challenges of Optical Character Recognition

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

Abstract

This paper is about Optical Character Recognition(OCR) on scanned documents. The focus is on how the technology works, the weaknesses in current OCR programs, and how we can start to address the identified weaknesses. This paper uses Tesseract, Amazon Textract, and Google Document AI as example models, and looks at English and Arabic documents specifically.

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

1 Introduction

There is a wide variety of documents that have been photographed[1]. These images are popular because they are easy to create and share. These images are inferior to traditional digital documents in the sense that they [are not search-able] or editable without fundamentally changing the structure.[2] Optical Character Recognition is a [technology] made to extract text from images. This paper looks at how OCR works, specifically for scanned documents, and looks at some of the specific weaknesses in applying OCR to these documents.

2 Background

The first step in using OCR on a document is to acquire an image of the document. This can be done using a camera or a scanner. These images can be saved as a variety of formats, such as a .pdf, .jpg, or .png. This file is then input to an OCR model, such as Tesseract, Textract, or Document AI[3] 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.[4]

Pages to Reference[1, 4, 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 step frequently includes converting the input image to 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 characters. The output of this step is each identified character in its own defined box of pixels.

2.3 Recognition

The final step in the OCR process is referred to as Classification or Recognition. This step takes the boxes of individual 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][5]

2.4 Comparison

The main considerations when judging OCR models are accuracy and speed.[4]

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.

3.1 Layout

[There exists many ways to format a paper.]

[1] passive voice...

[2] garbage

[3] Document AI can mean many things, even within the topic of OCR, here in this paper it specifically means the OCR model Google Document AI

[4] There are some models which also return information about where they found each character and with this they can output the text in a variety of formats.

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

3.2 Alphabet

The Latin alphabet has about 25 letters, each with a lowercase and uppercase variant, is written left-to-right and is primarily non-cursive. OCR is so accurate when applied to [Latin-alphabet texts] because of the simplicity of the alphabet and because they are generally trained on English text. [6]

This training does not automatically transfer to other alphabets.

Arabic, in comparison to Latin is a significantly more complex alphabet. The alphabet consists of 28 characters, uses contextual forms, and is unicase. Arabic is cursive, where each character is connected. The alphabet is written right to left. The final bit of complexity to note about this alphabet, is the use of diacritics, dots and marks, which can appear above or below the main text and influence the meaning.

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)

5 Conclusion

While it's ideal to have one golden bench-marking set, that's hard (because of the reasons outlined above), so now we have specialized ones. Honestly I don't have a full stance on if I think these specialized sets are good. Yes, they highlight weaknesses of general OCR models, and push the development of OCR further, but blind acceptance and promotion of them can neglect some of the specialties of OCR, like layout-specific models. Data sets also don't really encourage reducing time and resource complexity for models, which I would like to see. I am interested in this topic of OCR because of the role I think it could play in digital accessibility, but for that we would need to push for more free models with graphic user interfaces.[7]

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.

[6] you wanna cite this?

[7] for context, Document AI and Textract are both paid models, and Tesseract only has 3rd party GUIs