TRANSLATING OPTICAL CHARACTER RECOGNITION TEXT TO WORDS WITH THE READING TIME MACHINE

David Zhu, Prof. Jill Naiman

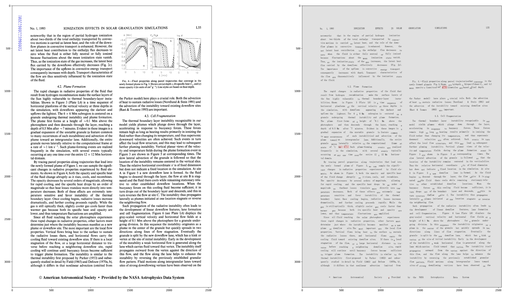
National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign

INTRODUCTION

What is OCR?

- OCR stands for **optical character recognition**, the technique to convert a pixel image into words.
- Some ancient books are only available in image so some words may be hard to see or digitize

- Current issues we found with OCR data are missing characters, punctuations/line feeds, and grammar errors.
- For the OCR engine, we choose the google OCR engine Tesseract.



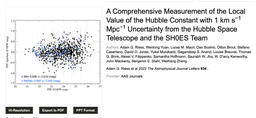
▲ *fig 1 - left side is the image of an article and right side is the OCR output of it*

AIM

- In this study with Prof. Naiman Jill, we explored different ways to examine how to increase the readability and correctness of OCR data so they can read better to humans. An example of an OCR'd page is in the figure.

- Overall goal: digitize pre-digital (published prior to 1997) astrophysical literature, and probably to host figures + captions at AIE (Astronomy Image Explorer).

- My project: The aim of this project is to optimize the OCR process in order to minimize errors in extracted text.



▲ *fig 2 - AIE figure with captions*



▲ *fig 3 - AIE figure without captions*

METHOD

Error Rates

- The word error rate (WER) and the character error rate (CER) are quantifiers to show how many errors occur in an OCR result.

- The formula is the sum of edit operations (substitution, deletion, insertion) divided by the sum of all words/ characters. WER looks at word-level scope while CER looks at character scope.

Substitution Deletion Insertion

$$CER = \frac{S + D + I}{N}$$

$$WER = \frac{S_w + D_w + I_w}{N_w}$$

Number of chars/words

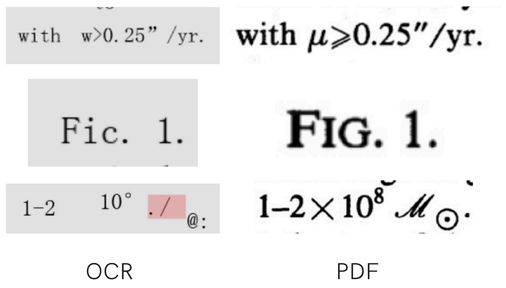
▲ *fig 4 - the formula of WER and CER; Sw, Dw, Iw are substitutions, deletions, insertions in word level.*

Confidence Scores

- We tried pytesseract on several pdf pages to get the word output and its confidence score.

- Tesseract confidences can be low for objects like math formulas (red in *fig 5*) and errors can occur even for high confidence scores (such as "Fic" and "Fig" in *fig 5*).

- We highlighted in red some textboxes that pytesserct does not feel so confident so we have an overview of what are common errors. There are also text boxes where OCR feels confident but an actual error, such as "Fic" and "Fig" *fig 5*.



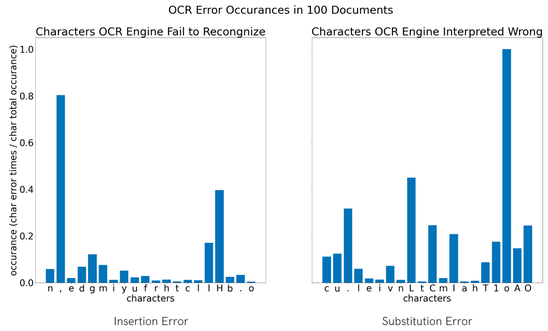
▲ *fig 5 - OCR errors with their ground truth*

RESULTS

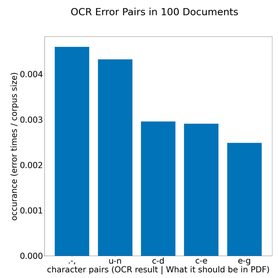
Error Frequencies for Each Letter

- We tried to look at each character to see which characters have the highest probability of being OCR'd incorrectly.

- I used the Levenshtein Distance metric to sort each edit operation and count the occurrence of each wrong character divided by the total occurrence of it.



▲ *fig 6 - The character level error frequencies; left side are characrers OCR fail to recongnize, and right side are characters OCR engine interpreted wrong*



◀ *fig 7 - top 5 frequently OCR error pairs; left part in a pair is the OCR result, and right part in a pair is what it should be as ground truth*

Possible Configurations

- Based on the discovery of error rates and our dataset, we can try different approach such as training data, Sentence fragements, configurations, and language models *(fig 8)*.

Page segmentation modes:

0. Orientation and script detection (OSD) only.
1. Automatic page segmentation with OSD.
2. Automatic page segmentation, but no OSD, or OCR. (not implemented)
3. Fully automatic page segmentation, but no OSD. (Default)
4. Assume a single column of text of variable sizes.
5. Assume a single uniform block of vertically aligned text.
6. Assume a single uniform block of text.
7. Treat the image as a single text line.
8. Treat the image as a single word.
9. Treat the image as a single word in a circle.
10. Treat the image as a single character.
11. Sparse text. Find as much text as possible in no particular order.
12. Sparse text with OSD.
13. Raw line. Treat the image as a single text line, bypassing hacks that are Tesseract-specific.

OCR Engine modes:

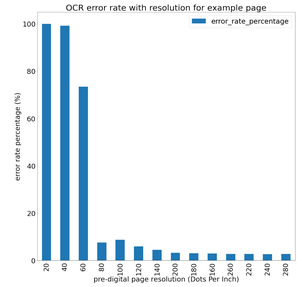
0. Legacy engine only.
1. Neural nets LSTM engine only.
2. Legacy + LSTM engines.
3. Default, based on what is available.

◀ *fig 8 - different tesserect configurations and language models; Entries in grey would be first considered common configurations*

CONCLUSION

- The focus of my work has been to build a pipeline for quantifying the OCR errors of Tesseract on astronomical literature. My future work will explore the parameter space of Tessearct's many inputs in order to optimize performance on our dataset."

- The next step is to compare how different resolutions and file formats affect the performance of the OCR engine *(fig 9)*.



◀ *fig 9 - the OCR error rate in relation with file size (in MBs) with jpg format*

REFERENCES

- Astronomy Image explorer. Astronomy Image Explorer. (n.d.). Retrieved July 22, 2022, from http://www.astroexplorer.org/
- Astrophysics data system. NASA/ADS. (n.d.). Retrieved July 22, 2022, from https://ui.adsabs.harvard.edu/
- B., S. V., Majumder, B., Gupta, A., & Surana, H. (2020). Practical natural language processing: A comprehensive guide to building real-world Nlp systems. O'Reilly Media.
- Filyp. (n.d.). Filyp/autocorrect: Spelling Corrector in python. GitHub. Retrieved July 22, 2022, from https://github.com/filyp/autocorrect
- Hládek, D., Staš, J., & Pleva, M. (2020). Survey of automatic spelling correction. Electronics, 9(10), 1670. https://doi.org/10.3390/electronics9101670
- Jiang, M., Dubnicek, R. C., Worthey, G., Underwood, T., & Downie, J. S. (2022). A prototype Gutenberg-hathitrust sentence-level parallel corpus for OCR error analysis. Proceedings of the 22nd ACM/IEEE Joint Conference on Digital Libraries. https://doi.org/10.1145/3529372.3533298
- Leung, K. (2021, September 23). Evaluate OCR output quality with character error rate (CER) and word error rate (WER). Medium. Retrieved July 22, 2022, from https://towardsdatascience.com/evaluating-ocr-output-quality-with-character-error-rate-cer-and-word-error-rate-wer-853175297510
- Pytesseract. PyPI. (n.d.). Retrieved July 22, 2022, from https://pypi.org/project/pytesseract/
- Rijhwani, S., Anastasopoulos, A., & Neubig, G. (2020). OCR post correction for endangered language texts. Proceedings of the 2020 Conference on Empirical Methods in Natural

ACKNOWLEDGEMENTS

Thank to prof. Jill Naiman and the help of NCSA SPIN program.

ILLINOIS