Proceedings of the International Conference IMCS-60, MSU
October 10–13, 2024, Chisinau, Republic of Moldova

Comparison of Large Language Models and Traditional Neural Networks in Optical Character Recognition for Old Alphabets

Marius Cerescu, Tudor Bumbu

Abstract

This study compares large language models (LLMs) and traditional neural networks (TNNs) in Optical Character Recognition (OCR) for historical alphabets. While deep learning has advanced OCR technology, recognizing old scripts remains challenging due to their complexity. LLMs, with vision capabilities, offer a novel approach by integrating visual and linguistic understanding. This research evaluates the accuracy and robustness of both models on a dataset of an ancient alphabet, highlighting the potential of LLMs to improve OCR in historical linguistics and digital preservation. The findings provide valuable insights for applying modern AI to the preservation of historical texts.

Keywords: Large Language Models (LLMs), Traditional Neural Networks (TNNs), Optical Character Recognition (OCR), Historical Scripts, Digital Preservation, Text Recognition.

1 Introduction

Optical Character Recognition (OCR) is crucial for digitizing and preserving historical texts, enabling easier access to ancient manuscripts. However, adapting OCR models to recognize new scripts, particularly in historical documents, remains challenging due to the need for specialized datasets, complex training, and significant computational costs.

Traditional neural networks (TNNs), like those used in ABBYY FineReader [1], are widely applied for OCR but require extensive effort

©2024 by Marius Cerescu, Tudor Bumbu

to achieve accuracy with rare or complex scripts. The "Bratul Codex" [2], a 16th-century Romanian manuscript in semi-uncial Cyrillic, exemplifies these challenges due to its historical script and document degradation.

Recent advancements in large language models (LLMs), such as GPT-4o [3], offer new possibilities for OCR by utilizing pre-existing knowledge to interpret complex scripts with minimal training. This study compares LLMs and TNNs in OCR for the "Bratul Codex," focusing on accuracy, aiming to show the potential of LLMs in simplifying the recognition of complex scripts and contributing to digital preservation.

2 Methodology

In this chapter, we describe the methods and tools used to evaluate the performance of Large Language Models (LLMs) and Traditional Neural Networks (TNNs) in Optical Character Recognition (OCR) tasks. The dataset for this study was derived from the "Bratul Codex", a 16th-century manuscript comprising 458 pages. For the analysis, 15 well-preserved pages were selected, and the images were preprocessed using "ScanTailor" [4] software. Preprocessing included correcting page orientation, deskewing, setting margins, and converting the images to black and white to enhance contrast and improve the quality of character recognition.

Two models were evaluated in this study: a Large Language Model (LLM) using GPT-4o with vision capabilities and a Traditional Neural Network (TNN) implemented via ABBYY FineReader PDF 15. GPT-4o was accessed via both the OpenAI API and the web interface [5], allowing for a flexible application of the model in recognizing characters from the preprocessed images. The GPT-4o model was used in its pre-trained state without further fine-tuning.

For ABBYY FineReader, a custom OCR project was created, with a focus on training the OCR model using a custom alphabet specific to the characters in the "Bratul Codex." The training process involved

manually assigning characters from the scanned images to corresponding characters in the custom alphabet, improving the model's recognition accuracy over time.

To compare the systems, accuracy was selected as the primary metric. Both models were tested on the same set of 15 preprocessed pages to ensure consistency in the evaluation. This comparative approach aimed to highlight the strengths and limitations of LLMs and TNN-based OCR software in handling complex historical scripts.

3 Results

This section presents a comparative analysis of Large Language Models (LLMs) and Traditional Neural Networks (TNNs) in Optical Character Recognition (OCR) for the "Bratul Codex." Several configurations were tested to assess both accuracy and efficiency under varying conditions. The study first compares the overall accuracy between the LLM and TNN models. It then examines the impact of different training data sizes on TNN performance, followed by an analysis of how batch sizes and system prompts affect LLM accuracy. Finally, feedback training for the LLM is explored to assess its potential for improving accuracy.

3.1 TNN Accuracy and Training Data Size

The TNN model, implemented using ABBYY FineReader PDF 15, was first trained on a dataset of 5 pages (2,520 characters) from the "Bratul Codex". Testing on a separate set of 504 characters resulted in an accuracy of 75%, with 129 unrecognized characters. When the training dataset was expanded to 10 pages (4,877 characters), the accuracy improved significantly to 90%, with only 45 errors in a test set of 495 characters. This demonstrates that increasing the training data volume enhances TNN performance, particularly when handling complex historical scripts like semi-uncial Cyrillic.

3.2 LLM Accuracy with different Batch Sizes and System Prompts

The GPT-4o model was evaluated using different batch sizes and interfaces. Initially, with 9 characters sent per request via the API and using a predefined system prompt [6], the model achieved 50% accuracy on a set of 495 characters. When characters were processed one at a time, accuracy increased to 64%, suggesting that the model benefits from focusing on individual characters. However, using the ChatGPT interface, with the same initial prompt, yielded different results: batch processing of 9 characters resulted in 58% accuracy, while individual character recognition dropped to 43%. This indicates that the interface used can significantly influence the model's performance, likely due to how context is managed in each scenario.

3.3 LLM with Feedback Training

An additional experiment explored the potential for improving LLM accuracy through feedback. Using the ChatGPT interface, the model was initially taught how each character appeared. Then, in an iterative process involving 25 feedback messages, the model's accuracy for single-character recognition reached 100%. However, for images containing multiple characters, further feedback training was required. For images with up to 5 characters, the model maintained high accuracy after 40 training iterations with feedback, but its performance diminished substantially with increasing text complexity. For words with more than 5 letters, the model struggled, often failing to return the correct number of letters, which significantly contributed to the decrease in accuracy, making it not viable to train the model for such cases.

3.4 Summary of Findings

Overall, the results indicate that while TNNs achieve higher accuracy with larger training datasets, LLMs like GPT-4o can deliver competitive performance under certain conditions, particularly when recogniz-

ing individual characters. The LLM's accuracy is influenced by batch size and the interface used, with the best results obtained from individual character processing. Feedback training is a promising method for enhancing LLM performance, especially for single-character recognition, though its effectiveness decreases as text complexity increases. Further research is needed to optimize LLMs for multi-character OCR tasks in historical documents.

4 Discussion

One of the primary challenges with using LLMs via the API is its stateless nature. Since each API call is independent, it is not possible to train the model through ongoing conversations or iterative feedback. This necessitates the creation of a single, highly effective prompt that can instruct the model on how to recognize each character in the script. Crafting such a prompt is particularly challenging, as it must be both comprehensive enough to cover the nuances of the script and concise enough to be practical for repeated use in API calls. The prompt needs to be optimized for clarity and brevity since it has to be included in every request, adding to the computational and financial costs of processing large volumes of text.

In contrast, using the ChatGPT interface allows for a more adaptive and conversational approach. Here, the ability to engage in a dialogue with the model makes it easier to teach the model how to recognize specific characters or words. This iterative process can rapidly improve the model's accuracy in recognizing individual characters or small words, making it a viable solution for tasks where quick adaptation is necessary. The interface's conversational nature also makes it possible to provide immediate feedback, leading to a more dynamic and flexible OCR process.

However, when dealing with large amounts of text, the API is more practical despite its limitations. It allows for the processing of extensive documents without the need for continuous human interaction. The key to maximizing its effectiveness lies in finding the ideal prompt

that balances detail with efficiency. This prompt must be capable of guiding the model to accurately recognize a wide variety of characters with minimal input, which is a non-trivial task given the complexity of historical scripts.

On the other hand, the TNN approach, as implemented in ABBYY FineReader, is highly effective for achieving high accuracy, particularly when a significant amount of training data is available. The process of training the OCR model, while time-consuming, results in a robust system capable of processing large texts with high precision. This makes TNNs the preferred choice for projects that require the digitization of extensive documents with a focus on accuracy. However, the time and effort required to train the model are substantial, which may not be ideal for projects with tight deadlines or limited resources.

5 Conclusion

The choice between using an LLM via API, the ChatGPT interface, or a TNN model depends largely on the specific requirements of the task. For large-scale text digitization where high accuracy is paramount, TNNs are the best option, despite the extensive training required. For more dynamic, small-scale tasks where quick adaptation is needed, the ChatGPT interface offers a flexible and efficient solution. Meanwhile, the API, with an optimized prompt, provides a middle ground, capable of handling large volumes of text efficiently but with the challenge of crafting the ideal prompt to achieve consistent accuracy.

This discussion highlights the importance of selecting the right tool for the job, taking into consideration factors such as the volume of text, the desired accuracy, and the available resources. Future work could explore further optimizations of prompts for LLMs and more efficient training processes for TNNs to enhance their applicability in various OCR tasks.

Acknowledgments. SIBIA - 011301, Information systems based on Artificial Intelligence has supported part of the research for this paper.

References

[1] *ABBYY FineReader PDF website*, 2024. Retrieved from https://pdf.abbyy.com/finereader-pdf/.

[2] *Bratul Codex website*, 2024. Retrieved from http://media.lit.uaic.ro/gafton/txt/text.php?page=20.

[3] *Hello GPT-4o – OpenAI website*, 2024. Retrieved from https://openai.com/index/hello-gpt-4o/.

[4] *ScanTailor website*, 2024. Retrieved from https://scantailor.org/.

[5] *ChatGPT website*, 2024. Retrieved from https://chatgpt.com/.

[6] M. Cerescu. *Bratul codex character description prompt,* 2024. Retrieved from https://docs.google.com/document/d/18LfDOu5NHMgTn_osReJLQb0iIt63pJeVgtD9i19-EYw.

Marius Cerescu[1], Tudor Bumbu[2]

[1,2]Moldova State University,
Vladimir Andrunachievici Institute of Mathematics and Computer Science
[1]E–mail: marius.cerescu@gmail.com
ORCID: https://orcid.org/0009-0000-5704-6577
[2]E–mail: bumbutudor10@gmail.com
ORCID: https://orcid.org/0000-0001-5311-4464