SHORT-PAPER

Post-OCR Correction with OpenAI's GPT Models on Challenging English Prosody Texts

JAMES ZHANG

WOUTER HAVERALS

MARY NAYDAN

BRIAN W. KERNIGHAN

Published: 18 September 2024

DocEng '24: ACM Symposium on Document Engineering 2024
August 20 - 23, 2024
CA, San Jose, USA

Conference Sponsors:
SIGWEB

https://doi.org/10.1145/3685650.3685669
ISBN: 9798400711695

Post-OCR Correction with OpenAI's GPT Models on Challenging English Prosody Texts

James Zhang
Department of Computer Science
Princeton, New Jersey, USA
james.zhang@princeton.edu

Wouter Haverals
Center for Digital Humanities
Princeton, New Jersey, USA
wouter.haverals@princeton.edu

Mary Naydan
Center for Digital Humanities
Princeton, New Jersey, USA
mnaydan@princeton.edu

Brian W. Kernighan
Department of Computer Science
Princeton, New Jersey, USA
bwk@cs.princeton.edu

ABSTRACT

The digitization of historical documents faces challenges with the accuracy of Optical Character Recognition (OCR). Noting the success of large language models (LLMs) on many text-based tasks, this paper explores the potential of OpenAI's GPT models (3.5-turbo, 4, 4-turbo) on the post-OCR correction task using works from the Princeton Prosody Archive (PPA), a full-text searchable database containing English texts published between 1559 and 1928 on versification and pronunciation. We conduct a comparative analysis across different model configurations and prompt strategies. Our results indicate that tailoring prompts with work metadata is less effective than anticipated, though adjusting the temperature parameter can be beneficial. The models tend to overcorrect works with already good OCR quality but perform well overall, with the best model setup improving the Character Error Rate (CER) by a mean of 18.92%. Additionally, after introducing a preliminary quality estimation step to process texts differently based on their original OCR quality, the best mean improvement increases to 38.83%.

CCS CONCEPTS

• **Applied computing** → **Optical character recognition**; • **Computing methodologies** → *Natural language processing*; • **General and reference** → *Evaluation.*

KEYWORDS

Post-OCR Correction, Large Language Models, LLMs, Optical Character Recognition, Error Correction, GPT, Historical Documents

ACM Reference Format:
James Zhang, Wouter Haverals, Mary Naydan, and Brian W. Kernighan. 2024. Post-OCR Correction with OpenAI's GPT Models on Challenging English Prosody Texts. In *ACM Symposium on Document Engineering 2024 (DocEng '24), August 20–23, 2024, San Jose, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3685650.3685669

DocEng '24, August 20–23, 2024, San Jose, CA, USA

ACM ISBN 979-8-4007-1169-5/24/08
https://doi.org/10.1145/3685650.3685669

1 INTRODUCTION

OCR has been transformative in broadening access to archival materials that have been previously locked in their physical formats. However, OCR technologies still struggle with older historical texts, as they frequently contain unique typography, nonuniform layout, and degraded printing quality, making digitization challenging for archives and libraries. The PPA presents an additional challenge because its prosodic works often feature musical notation and idiosyncratic markings, which are exceptionally difficult to parse into text. This layer of difficulty makes the OCR outputs of works from the PPA particularly interesting to work with.

We prioritize correcting OCR outputs to enhance retrieval applications. Poor OCR performance is troublesome for collections like the PPA, which relies on the OCR outputs to power searches in its database. The ability to effectively query a database with specific keywords to identify works is crucial for productive research in the digital humanities [3]. Without reliable OCR, many archives resort to post-processing procedures to clean the text outputted by OCR. While manual correction is ideal for accuracy and trustworthiness, it is infeasible due to sheer corpus size and associated costs.

Recent advances in LLMs offer a promising angle for improving OCR output quality. In particular, OpenAI's GPT models have shown impressive performance on problems close to the post-OCR correction task, such as machine translation [6] and text correction [12]. LLMs perform well even in zero-shot settings (i.e., when the model is asked to perform a task it has not seen before) [12]. This robustness is especially beneficial for collections that cover niche topics and lack the data for resource-intensive solutions.

In this work, we assess three recent LLMs released by OpenAI (GPT 3.5-turbo, 4, and 4-turbo) on their ability to perform post-OCR correction on the textual OCR outputs provided by the PPA. We experiment with a mix of six different prompts and parameter changes to find which information and settings are best to offer the models when making corrections. To be able to conduct the evaluation, we manually transcribe some pages for ground truth.

We find that providing contextual information (e.g., author name, publication year, etc.) does not lead to significant improvements over the vanilla prompt, which itself does well. We also observe that the models struggle with OCR texts that are either extremely erroneous or already good. This analysis leads us to incorporate a recently proposed LLM-based solution to the quality estimation problem to flag these two classes ahead of time. Synthesizing our

findings, we propose a pipeline that the PPA and other corpora can implement to immediately enhance their OCR output quality.

2 RELATED WORKS

Currently, it is common for archives to employ lightweight yet domain-specific methods. These solutions are designed to run fast and target the most common and notable errors via heuristic rulesets. For example, in the PPA, post-OCR correction is now performed using a Python script that scans for OCR artifacts and applies rule-based corrections. The key corrections included (1) processing running headers, (2) rejoining linebreaks, (3) changing the historic long "s" (ſ) to the modern "s", and (4) checking words containing the letter "f" as the OCR might mistakenly interpret the long "s" as "f." Cleaned texts are then checked against a predefined dictionary for spelling mistakes.

While these approaches address straightforward errors, they struggle with more important and complex issues. These limitations have driven the exploration of more encompassing alternatives, with neural language models being a common choice. Popular architectures like Gated Recurrent Unit (GRU) [10], Long Short-Term Memory (LSTM), and transformers are continually revisited.

The prevailing strategy involves training — or fine-tuning when data or funding is limited — a transformer, using domain-specific data. This effectively turns these models into specialized experts tailored for precise tasks that are usually not generalizable (e.g., cursive Urdu text [11], Cyrillic handwriting based on Bézier curves [4], and 19-20th century Swedish newspapers [8]). While these models boast great performance scores and improvement on previous state-of-the-art methods, they lack versatility and require significant training data. For collections like the PPA where training data are scarce, these solutions are practically unattainable.

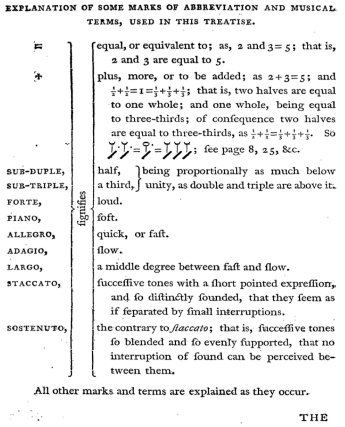
Despite the ease of access to LLMs, research on their application to post-OCR correction has been limited. Boros et al. [1] benchmarked a few LLM series (GPT, BLOOM(Z), OPT, Llama) and found that all LLMs performed poorly, with most worsening the input text with hallucinations and others leaving it relatively unchanged. More recently, however, Thomas, Gaizauskas, and Lu [9] fine-tuned Meta's Llama 2 to correct 19th-century British newspapers' OCR outputs and reported major improvements over a traditional transformer-based approach. They propose using a prompt-based framework to integrate LLMs into this problem.

Following a prompt-based approach, we test new and tailored methods, assessing the GPT models without additional fine-tuning to ensure our method remains low-resource while leveraging domain expertise. We also propose a solution inspired by the quality estimation problem to reduce LLM-produced hallucinations when generating corrections. To the best of our knowledge, this is the first time quality estimation has been applied in this manner.

3 DATASET

Works from the PPA cover many topics, from phonology to grammar to history. The archive stresses that its characterizing term *prosody* is reflected in the broadest sense to include as many works as possible. At the time of our research, its 6,754 works are classified into six non-disjoint collections: dictionaries, linguistic, literary, original bibliography, typographically unique, and word lists.

We focus on documents tagged *typographically unique* (TU) because they encompass a variety of unique textual elements such as musical notation, invented diacritical marks, and phonetic scripts, which often present significant challenges for accurate OCR [5]. A total of 693 works are tagged TU. Fig. 1 presents a typical page (left) and its OCR output (right).



EXPLANATION OF SOME MARKS OF ABBREVIATION AND MUSICAL TERMS, USED IN THIS TREATISE.

= equal, or equivalent to; as, 2 and 3 = 5; that is, 2 and 3 are equal to 5.

+ plus, more, or to be added; as 2+3=5; and ½+½=1=¼+¼+¼+¼; that is, two halves are equal to one whole; and one whole, being equal to three-thirds; of consequence two halves are equal to three-thirds, as ⅓+⅓=⅔+⅓; So 𝅗𝅥.𝅗𝅥.=𝅗𝅥.=𝅗𝅥.𝅗𝅥.𝅗𝅥.; see page 8, 25, &c.

figures

SUB-DUPLE, SUB-TRIPLE, half, being proportionally as much below a third, unity, as double and triple are above it.

FORTE, loud.

PIANO, soft.

ALLEGRO, quick, or fast.

ADAGIO, slow.

LARGO, a middle degree between fast and slow.

STACCATO, successive tones with a short pointed expression, and so distinctly sounded, that they seem as if separated by small interruptions.

SOSTENUTO, the contrary to *staccato*; that is, successive tones so blended and so evenly supported, that no interruption of sound can be perceived between them.

All other marks and terms are explained as they occur.

THE

EXPLANATION\n. . SUB-DUPLE, SU]-TRIPLE, FORTE, PIANO, ALLEGROS ADAGIO, LARGO, STACCATO, SOSTENIJTO,-\n.: J\nOF SOME MARKS OF ABBREVIATION AND MUSICALU TERMS, USED IN THIS TREATISE.\n\"equal, or equivalent to; as, z and 3= 5; that is,\nz and 3 are equal to 5. plus, more, or to be added; as 2+3=5; and\n2+=I ="+3+ ; that is, two halves are equal to one whole; and one whole, being equal to three-thirds; of consequence two halves are equal to three-thirds, as z-+-=j -+j-. So „ - „= \'=J3;; fee page 8, 25, &-c. half, I being proportionally as much below a third,J unity, as. double and triple are above it. m lloud. b- soft.\nquick, or fast. flow. a middle degree between fasi and flow. successive tones with a. flort pointed expreflionr,\nand so diflin\'tly founded, that they seem as if separated by finall interruptions. the contrary tojlaccato; that is, ficceflive tones\nso blended and f6 evenly supported, that no interruption of found can be perceived be- tween them. All other marks. and terms are explained as they occur.-\nTHE

Figure 1: From *Prosodia rationalis: or, An essay towards establishing the melody and measure of speech, to be expressed and perpetuated by peculiar symbols* by Joshua Steele (1779) via *Eighteenth Century Collections Online*, Gale Primary Sources.

For all its works, the PPA holds images of the pages, their OCR outputs, and metadata, which are sourced from Gale Cengage's *Eighteenth Century Collections Online* and HathiTrust Digital Library. Key metadata elements include the title, subtitle, author, publication year and place, and collection type. Due to the setup by PPA's providers, no reliable mechanisms are in place to verify OCR accuracy, and some works underwent OCR long ago (circa 2008).

4 METHODOLOGY

4.1 Manual Transcriptions

Because there is no one "correct" transcription, we handpick and transcribe 21 pages from various TU works that we anticipate would be challenging for OCR. Selection criteria included poor scan quality (e.g., background noise and page warping) and uncommon typography or notation systems. Thus, we run corrections (at the page level) on the OCR outputs of the selected pages and compare them against our transcriptions, serving as the gold standard.

We acknowledge that our transcriptions are not perfect, but we ensure consistency and that they faithfully portray and preserve the content by adhering to the following four principles:

(1) Record all text that can be expressed in Unicode
(2) Reflect the author's intended meaning and reading
(3) Keep grammatical and typographic errors
(4) Transcribe all non-Unicode symbols as @SPECIAL_CHAR@

While we do not instruct the LLMs to apply the last rule, we include it to penalize the corrections. We hope future work addresses this capability, as it is highly desired by researchers and archivists.

4.2 Model Configurations and Prompts

We used a single Python script to run the corrections with all our approaches. Prompts were stored as strings and rotated into the API call when necessary. The prompts themselves were written in a plug-and-play style for scalability. To get means, we ran a single setup ten times for each of the three GPT models, saving the result in its own file. We consider this a single execution, which took about 3.5 hours and $11 USD in OpenAI API credits.

The GPT API accepts two prompt types: system and user. The former instructs the model on how it should act and respond to user commands; the latter are the direct user inputs that elicit specific actions from the model. We experiment with various user prompts while maintaining the same system prompt throughout. We provide the system and baseline user prompts below for context. All of our prompts can be found in the GitHub repository, linked in Section 6.

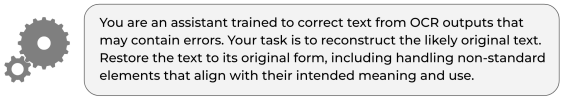


Figure 2: Our universal system prompt.

We aimed to convey two key points concisely when crafting this system prompt. First, only certain parts of the given OCR text may be erroneous. This is to curb overcorrection, which is a problematic tendency [1]. Second, to align the model's approach with the perspective we adopted in our manual transcriptions, corrections should be based on the author's intent. In the following subsections, we detail each modification and provide the underlying intuition.

4.2.1 Vanilla. To be concise, we omitted polite terms (e.g., "please") and used directives. We addressed overcorrection further by verbalizing a penalty. Additionally, we used headers to separate different sections. These strategies have been found to work well [2].

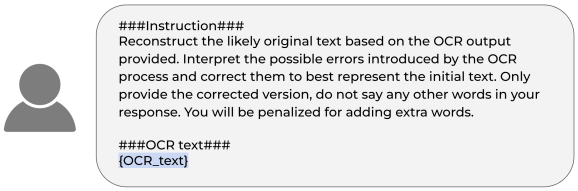


Figure 3: Our vanilla prompt.

4.2.2 Explaining "Typographically Unique". To provide the models with better domain-specific context, we adapted the definition of *typographically unique* from the official PPA website and prepended it to the vanilla prompt. We also added *###Context###* as a header.

4.2.3 Work-Specific Metadata. Using key metadata described in Section 3, we tailored the prompts to the work of each page. Notably, the title and subtitle of a work had a mean of a substantial 36 words, which we assumed would reveal salient content and insight into the topics of the pages to be corrected.

4.2.4 Temperature. Using our vanilla prompt, we varied the temperature parameter at intervals of 0.2, from 0 to 1.2.

4.2.5 Correctness Aware. We test a hypothetical scenario by providing the CER score and other error details. Detailed later in Section 5, CER measures the percentage of characters incorrectly recognized by the OCR. Though this setup is typically unavailable since it requires the gold standard, we wonder if the models could make better edits if informed of how erroneous the OCR was beforehand.

4.2.6 LLM as Second Reader. We emulate the notion of a second reader, who is used to verify and refine work done by a previous reviewer. To do so, we take the output of a single correction pass and use it as input for a second pass. With this iterative process, we aim to catch and fix as many errors as possible.

4.3 Quality Estimation (QE) Based Classifier

After an initial correction pass of all the pages, we noticed that the models underperformed when the original OCR output quality was either very poor or good. To address these cases, we adopted a three-pronged approach. For the former case, which likely consisted of works OCR'ed long ago, we reran OCR on the page images with the modern open-source Tesseract library and corrected those new outputs. For the latter, we left the outputs as they were. For all else, we proceeded with correction as normal.

To assess OCR output quality, we used a modified version of the GPT Estimation Metric Based Assessment (GEMBA) [7]. A prompt-based approach, GEMBA estimates the quality of outputs in machine translation tasks without needing the gold standard. Adapting it to our task, we treated the OCR outputs as translations and adjusted the best-performing gold-free Direct Assessment prompt. We chose GEMBA specifically for its simple output form, a number between 0 and 100. We regarded scores below 30 as poor and above 80 as good. The left image in Figure 5 details our full procedure.

5 RESULTS AND EVALUATION

We evaluated performance with the standard CER metric, which is based on the Levenshtein distance between the hypothesis and gold standard texts. We express CER as a nominal value (e.g., 0.46 as 46), where lower values signify better accuracy and fewer errors.

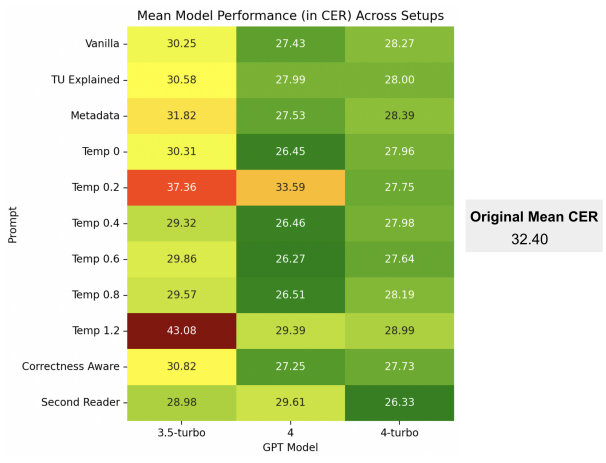


Figure 4: We use CER as a means to measure the searchability of documents in OCR-based systems.

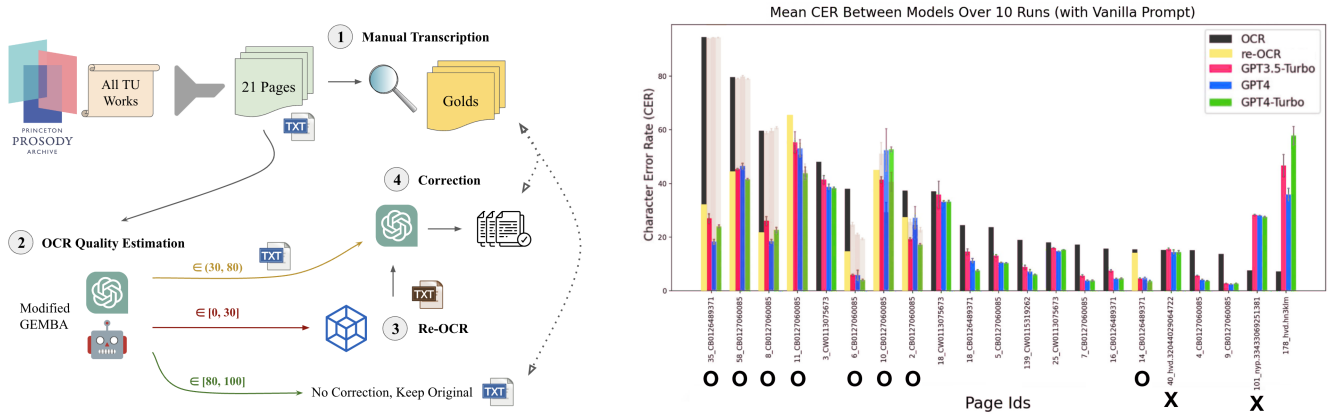


Figure 5: Our complete pipeline (left) and model performance improvements with the QE step (right). Pages with "O"s and "X"s were re-OCR'ed and untouched, respectively. The shaded bars represent correction performance with the original OCR texts.

Figure 4 shows the LLMs' performance across all configurations without the preliminary quality estimation step. A row for *Temp 1* is not included, as it is identical to the baseline *Vanilla* (the default temperature setting in the API is 1). Taking the mean over all setups, GPT4 (26.81) performs the best despite GPT4-turbo (27.18) being the more recent model. GPT3.5-turbo (31.08) does notably worse.

The various customizations do not seem to improve the LLMs' correction abilities. This may be because the information included was less relevant than anticipated. Lengthening the prompt could have distracted the LLMs from focusing on the crucial parts. The subpar result of the *Second Reader* approach might be explained by the text it was given to correct. The input to the second correction pass may have been too altered from the original. If the first run made mistakes, the second run likely exacerbated them.

The best overall setup, GPT-4 with a temperature of 0.6, was a simple parameter change, an 18.92% improvement over the given outputs. Nonetheless, the baseline itself worked well. These context-free performances suggest that effective corrections can still be achieved without domain-specific knowledge.

The right image in Figure 5 demonstrates the benefit of introducing the quality estimation (QE) step. We used the baseline because the tailored configurations were not significant. The modified GEMBA solution [7] recognizes the vast majority of texts with very poor or good OCR quality, leading to better final corrections. The mean CERs for GPT3.5-turbo, 4, and 4-turbo were 21.23, 19.84, and 19.82 respectively. These results are a huge jump (GPT4-turbo yielded an improvement of 38.83%) and suggest that implementing the full pipeline is worth pursuing, especially for retrieval purposes.

6 CONCLUSION

In this work, we proposed a novel prompt-based low-resource framework for post-OCR correction with LLMs. We found it particularly useful to handle works differently based on their original OCR quality with quality estimation techniques. While our approach shows promise, we acknowledge the limitations in the number of samples and tests used. Due to the PPA's sharing permissions, we are currently unable to share the data. Our code and prompts are available at https://github.com/jzhang512/post-ocr-correction.

ACKNOWLEDGMENTS

We thank the Princeton Prosody Archive (PPA) for providing access to their database and resolving our inquiries. We also thank the Princeton SEAS and Mikki Hornstein for helping fund this research.

REFERENCES

[1] Emanuela Boros, Maud Ehrmann, Matteo Romanello, Sven Najem-Meyer, and Frédéric Kaplan. 2024. Post-correction of Historical Text Transcripts with Large Language Models: An Exploratory Study. *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)* (March 2024). https://aclanthology.org/2024.latechclfl-1.14/

[2] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. 2024. Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4. (Jan. 2024). https://doi.org/10.48550/arXiv.2312.16171

[3] Ryan Cordell. 2017. "Q i-jtb the Raven": Taking Dirty OCR Seriously. *Book History* 20 (Oct. 2017). https://doi.org/10.1353/bh.2017.0006

[4] Evgenii Davydkin, Aleksandr Markelov, Egor Iuldashev, Anton Dudkin, and Ivan Krivorotov. 2023. Data Generation for Post-OCR correction of Cyrillic handwriting. (Nov. 2023). https://doi.org/10.48550/arXiv.2311.15896

[5] Selena Hostetler. 2023. A Typographically Unique Tour of the PPA. *Princeton Prosody Archive Editorial* (April 2023). https://prosody.princeton.edu/editorial/2023/04/a-typographically-unique-tour-of-the-ppa/

[6] Wenxiang Jiao, Wenxuan Wang, Jentse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? Yes With GPT-4 As The Engine. (Nov. 2023). https://doi.org/10.48550/arXiv.2301.08745

[7] Tom Kocmi and Christian Federmann. 2023. Large Language Models Are State-of-the-Art Evaluators of Translation Quality. (May 2023). https://doi.org/10.48550/arXiv.2302.14520

[8] Viktoria Löfgren and Dana Dannélls. 2024. Post-OCR Correction of Digitized Swedish Newspapers with ByT5. *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)* (March 2024). https://aclanthology.org/2024.latechclfl-1.23/

[9] Alan Thomas, Robert Gaizauskas, and Haiping Lu. 2024. Leveraging LLMs for Post-OCR Correction of Historical Newspapers. *The 2024 Joint International Conference on Computation Linguistics, Language Resources and Evaluation* (May 2024). https://aclanthology.org/2024.lt4hala-1.14

[10] Konstantin Todorova and Giovanni Colavizza. 2020. Transfer Learning for Historical Corpora: An Assessment on Post-OCR Correction and Named Entity Recognition. *CHR 2020: Workshop on Computational Humanities Research* (Nov. 2020). https://ceur-ws.org/Vol-2723/long32.pdf

[11] Nehal Yasin, Imran Siddiqi, Momina Moetesum, and Sadaf Abdul Rauf. 2023. Transformer-Based Neural Machine Translation for Post-OCR Error Correction in Cursive Text. *Document Analysis and Recognition – ICDAR 2023 Workshops* (Aug. 2023). https://doi.org/10.1007/978-3-031-41501-2_6

[12] Xiaowu Zhang, Xiaotian Zhang, Cheng Yang, Hang Yan, and Xipeng Qiu. 2023. Does Correction Remain A Problem For Large Language Models? (Aug. 2023). https://doi.org/10.48550/arXiv.2308.01776