Zero-shot Methods for Historical Text Restoration

Kiara M. H. Liu[1], Martin Mueller[2], and Matthew Wilkens[1]

[1] Department of Information Science, Cornell University, Ithaca, U.S.A.
[2] Department of English, Northwestern University, Cook County, U.S.A.

Abstract

The EarlyPrint corpus is a uniquely high-value resource, comprising over 60,000 digitized Early Modern English texts published between 1473 and the early 1700s. Despite having been created by hand-keying from scans of original documents, transcription defects remain a problem due to the limitations of early scanning technologies. Specifically, unrecognizable letters are denoted by the "blackdot" character ("•"). Previous methods, including both human review and an LSTM-based approach, had moderate success in correcting these transcription errors. This paper expands on previous work by exploring zero-shot techniques using historically adapted large language models. We identify two groups of blackdot words – we use lexical matching combined with zero-shot evaluation for the less challenging instances, and direct zero-shot prediction for the more complex cases. We achieve 95% accuracy on valid instances in the first group and 78.6% accuracy across the majority of blackdot words in the second. In total, we recommend 2.8 million missing-letter predictions and implement over 700,000 high-confidence corrections within the corpus, substantially improving data quality for scholarly use.

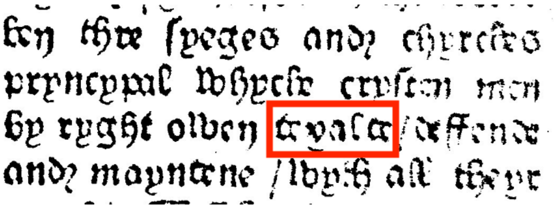
Keywords: historical text, zero-shot learning, language models, computational humanities

1 Background

The EarlyPrint corpus is a collection of over 60,000 digitized English-language books published between 1473 and the early 1700s, containing tokenized and part-of-speech (POS)-tagged texts from the EEBO (Early English Books Online) [21], ECCO (Eighteenth Century Collections Online) [10], and Evans (Evans Early American Imprints) TCP (Text Creation Partnership) projects [17]. The TCP projects themselves were collaborations between the University of Michigan Library, the Bodleian Libraries at the University of Oxford, ProQuest, and the Council on Library and Information Resources, aiming to create "standardized, machine-readable, searchable texts of early English print" from the image libraries of EEBO, ECCO, and Evans [15].

However, the texts contain transcription errors, especially in words that appear at the outer margins of the physical pages in the source texts, due to the limitations of early scanning technology. For example, EEBO's digitization process first began as a microfilm project in the 1930s. Not only did the ink in the physical books vary in readability, but the microfilm images themselves were also difficult to read at the margins. The interior margins posed a problem due to the slanted angle of the page near the spine of the book and were difficult to read without current-day "deskewing" technology; outer margins often contained smaller print, which was likely to fade and might even be accidentally cropped out of the image (see Figure 1) [19]. In cases where the transcribers could not recognize a letter, they marked the missing text using special characters, indicating "known unknowns." The most frequent type of known unknown is missing characters, marked by one or multiple "blackdot" characters ("•").

Kiara M. H. Liu, Martin Mueller, and Matthew Wilkens. "Zero-shot Methods for Historical Text Restoration." In: *Computational Humanities Research 2025*, ed. by Taylor Arnold, Margherita Fantoli, and Ruben Ros. Vol. 3. Anthology of Computers and the Humanities. 2025, 949–960. https://doi.org/10.63744/gz3Wm6kr19yr.



ben thre ſpeces and chyrches
prynncypal whyche crysten men
by ryght owen tyaſte / deffende
and mayntene / wyth all theyr

Figure 1: Example of text becoming less legible towards the outer margin on the right. The blackdot word is “t•ya••e” [1].

Although the majority of texts in the corpus contain relatively few errors, transcription accuracy remains an important area of improvement. In a 2013 study of around 200 EEBO-TCP users, most of whom were academics or students, 70% of respondents ranked “accuracy of transcription” as the most important factor to their work, with repeated requests for transcription accuracy as an area of future improvement [24]. This demonstrates that continued effort in correcting errors in the corpus is important to scholars in the field.

In 2016, an LSTM-based approach was developed to correct some of the blackdot words in the corpus [22]. This method involved two passes: one computing unigram word frequencies, and the other using an LSTM model trained on texts from the EarlyPrint corpus. The probabilities from both were then averaged to determine the final prediction. The performance of this approach was not fully measured, but a domain expert estimated that, for approximately 80% of corrupt tokens, the approach achieved around 80% accuracy. Over 3 million predictions were produced, but their accuracy was judged to be insufficient for bulk deployment. With the improved capabilities of transformer-based language models, we should be able to address errors that the 2016 approach could not fix, as well as better assess the performance of our new approach. We aim to use zero-shot prompting to demonstrate the feasibility of our approach in low-resource settings, particularly for humanities scholars working with smaller or sparsely labeled corpora.

2 Related Work

2.1 Text Restoration

Text restoration broadly refers to the restoration of texts where the writing or inscription is faded or otherwise difficult to discern; our project falls under this area as well. However, even though NLP research has proven helpful for this task, most of the research in this area focuses on truly ancient texts. For example, Assael et al. trained an LSTM-based seq2seq model on ancient Greek inscriptions first written between the 7th century BCE and the 5th century CE [2]. Later, Assael et al. trained an updated transformer-based model on the same dataset, resulting in an improvement in performance compared to the earlier approach [3]. Kang et al. also trained a transformer-based model to restore ancient Korean texts [12].

Most of the work in this area focuses on the restoration of older texts. As the language of our corpus is much closer to modern language, it is likely that pretrained LLMs, with minimal additional finetuning, could prove useful for correcting errors in our corpus.

2.2 Post-Transcription Correction

Much of the research in post-transcription correction has focused on Optical Character Recognition (OCR), particularly for correcting “unknown unknowns” (that is, errors that are not marked with special symbols). However, the development of the Text Creation Partnership (TCP) project began in the 1990s, long before OCR was commonly used in transcription workflows. At the time, transcriptions were performed manually from microfilm copies of early printed texts.

Nevertheless, the underlying principles of post-transcription correction remain consistent, whether the text in question originates from manual or OCR-based transcription. Both methods involve addressing errors where the transcription deviates from the original text, be they due to legibility issues, typographical mistakes, or inconsistencies in interpretation.

In particular, frameworks and techniques developed in this area remain relevant for the correction of manually transcribed texts. As summarized in Nguyen et al., approaches to post-OCR correction can largely be separated into three categories: manual, semi-automatic, and automatic [20]. The (semi-)automatic approaches can then be further categorized as isolated-word and context-dependent approaches.

1. Manual approaches are often crowdsourced, as also seen in the EarlyPrint project, which encourages users to "help [their] fellow scholar-readers by filling in gaps and correcting small errors in the corpus" [8].

2. Isolated-word approaches range from lexical approaches to simple models such as topic-based language models. Lexical approaches involve selecting corrections from candidate spellings based on some combination of dictionaries, word frequencies, and edit distances [9; 11; 13]. The initial steps of our process fall under this category, and involve the selection of candidate spellings based on a list of past corrections and word frequency.

3. Context-dependent approaches are purely language-model based. Recently, Thomas et al. and Boros et al. evaluated multiple large language models on post-OCR correction of historical texts, although their datasets mostly comprised newspaper text in the 18th–19th centuries [5; 25]. The later steps of our process fall under this category, and involve using a language model to decide between candidate spellings, given a context window.

Despite a large body of literature on post-OCR correction, there is rarely a focus on using language models to improve upon human transcriptions. Whereas a scholar transcribing from a high-quality digital image or the original text may produce very accurate results, non-scholarly professional and anonymous transcribers working from microfilm images of dubious quality may not. In our case, we dedicate effort towards fixing errors in human transcriptions themselves, and focus on "known unknowns" instead of "unknown unknowns."

2.3 Zero-shot Learning for Historical Text

Zero-shot learning refers to the ability of a model to perform tasks without having been explicitly trained on data for those tasks. It allows a model to generalize from previously learned knowledge to novel tasks or domains, even when it has not seen task-specific annotated examples during training [23]. This capability is particularly important to historical text processing, where annotated datasets are often scarce or absent, and domain-specific challenges – such as non-standardized spelling and historical grammar – further complicate the task.

One example of the application of zero-shot learning in historical text processing is Bollmann et al. which used both few-shot and zero-shot learning for normalizing historical texts [4]. They demonstrated that even the zero-shot approach often improved upon the identity baseline, albeit only slightly. Similarly, Toni et al. explored the capacity of T0, a transformer-based model, to perform zero-shot named entity recognition (NER), as well as document-level tasks such as identifying the date and language of a document [6]. However, they did not meet baseline performance for the NER task, recognizing noisy OCR as one possible reason.

Using pre-trained language models, zero-shot learning enables the application of state-of-the-art NLP techniques to historical text processing, even when labeled data is scarce or unavailable.

These approaches allow for the efficient processing of historical texts, without the need for extensive task-specific training and annotation. However, historical texts are also particularly prone to transcription or OCR errors, which present a significant challenge. In our study, we explore using zero-shot methods for the task of historical text correction in itself.

2.4 Early English Language Models

Models of historical languages provide a foundation for downstream tasks (such as text restoration, as outlined above) and future digital humanities research.

Pertaining to Early Modern English, MacBERTh is a transformer-based language model pretrained on multiple datasets of historical English, including EEBO, Evans, and ECCO, as well as the Corpus of Late Modern English Texts, the Corpus of Historical American English, and the Hansard corpus [16]. More recently, MonadGPT is a conversational finetune of Mistral-Hermes 2 on excerpts of English and French early modern texts, mostly from EEBO and Gallica, along with synthetic questions generated by Mistral-Hermes 2 [14].

As the focus of our study is on zero-shot methodology, we chose to incorporate MonadGPT into our comparison.

3 The EarlyPrint Project

The EarlyPrint project aims to "transform the early English print record, from 1473 to the early 1700s, into a linguistically annotated and deeply searchable text archive" [7]. The process of transforming TCP texts into EarlyPrint versions involves several key steps:

1. Standardization of certain characters and punctuation, along with the expansion of macrons and other abbreviations.

2. Refinement and adjustment of tags.

3. Tokenization, part-of-speech tagging, and regularization of spellings and lemmas in some cases.

Each token in an EarlyPrint text is assigned an ID, a part-of-speech tag, a lemma, and, where applicable, a "regular" attribute, which represents the standardized version of the word.

However, the process of digitizing historical texts – especially prior to the advent of modern scanning technologies – introduced challenges that persist in the data today. Early scans, particularly those created from microfilm copies of the original works, often suffered from issues such as poor readability, fading ink, and difficulties in capturing text from the outer margins of the page [18]. As a result, transcribers marked these missing or unidentifiable sections using specific symbols to indicate "known unknowns."

In the EarlyPrint corpus, the most common known unknowns are represented by the blackdot character "•," which indicates one or a few unidentifiable characters. Other markings include squares to denote unrecognizable punctuation and diamonds for whole words that were unidentifiable. Our project specifically focuses on words containing one or more blackdot characters, which we refer to as "blackdot words." These blackdot words form the basis of our study, as we aim to improve their transcription and correct errors introduced during the digitization process.

4 Processing

In our project, we use zero-shot evaluation to predict candidate corrections for blackdot words in the EarlyPrint corpus. Specifically, after we filter for blackdot words and collect candidate spellings, the model computes a loss value for each candidate. By ranking these loss values, we can identify the most likely corrections without the need for training a model specifically for blackdot word correction.

4.1 Preprocessing

For each text, we read the XML file and unwrap the nested text structures (e.g., pages, paragraphs, footnotes, tables) using a manually compiled list of tags. This process ensures that the text is simplified and organized, making it easier to obtain context windows in string format for later analysis.

4.2 Base Case: Lexical Matching & Zero-shot Evaluation

4.2.1 Lexical Matching

In order to improve upon our performance and processing time in the previous step, we use two sets of vocabulary – the vocabulary of the individual text from which the blackdot word is drawn and the vocabulary of all texts published during the same decade as the source text – to first find plausible candidate spellings, and then use the model to evaluate the most likely next spelling.

We use a vocabulary-based approach because, for any blackdot word, the chance that the correct transcription appears somewhere else in the same text or temporally similar texts is quite high. For each blackdot word, we first look for matches in the current vocabulary where the match's count is greater than one, to reduce the likelihood that the matched spelling is itself an error. If no matches are found in the current vocabulary, we look for matches among the recorded previous corrections. If, still, no matches are found, we look for matches in the vocabulary of the entire decade.

With this approach, we inevitably miss blackdot words whose correct spellings are so rare that they do not appear in any of these three vocabularies. Our aim, however, is not to correct every error. Rather, we seek to generate high-confidence corrections for the ”lowest-hanging fruits,” allowing human scholars to focus their effort on a smaller set of more complex – and potentially more interesting – instances.

4.2.2 Zero-shot Evaluation

In the zero-shot evaluation step, we evaluate three models' ability to predict possible candidate spellings of each blackdot word, obtained from the matches in the previous step.

For each candidate of each blackdot word, we pass into the model the tokenized context, with a maximum of 64 tokens on the left side and 16 tokens on the right side. Although the model is able to handle longer context sizes, we find that additional context does not consistently improve performance.

Each candidate match is tokenized separately and appended between the left and right side contexts. Completed excerpts are passed into the model, and cross-entropy loss is calculated over the entire sequence, indicating the likelihood of the excerpt given the candidate word being in its particular position. The candidate with the smallest loss value is taken to be the model's evaluation of the most likely spelling of the word.

For this method, we focus on blackdot words that match the following, moderately restrictive criteria:

1. The blackdot word must have fewer than 8 candidates. Evaluating over too many candidates would both take excessive compute time and decrease the meaningfulness of any differences in predicted likelihood.

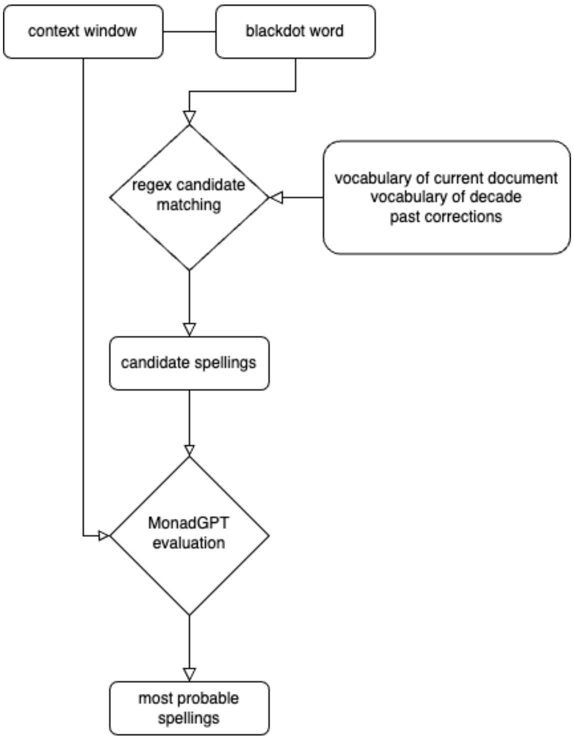


Figure 2: Pipeline structure for each blackdot word

2. The blackdot word must have 5 or more characters. Words with fewer than 5 characters tend to be function words with many candidate spellings, which pose greater difficulty.
3. The blackdot word must have 2 or fewer word-initial blackdots. Word-initial letters tend to have greater variation than word-terminal letters, which are often suffixes.

4.2.3 Model Comparison

We choose three models for our comparison – Llama-2-7b, T5, and MonadGPT 7B-GGUF. Llama-2 is a strong general decoder-only baseline, while T5 is an encoder-decoder baseline. MonadGPT, as introduced in 2.4, is a historically adapted instruction-tuned model. We expect it to retain representations well-suited to our corpus, even if instruction tuning modified some of its behavior. It should also be noted that, because EarlyPrint includes material from EEBO, and MonadGPT's training also drew on EEBO pages, some overlap between the model's prior exposure and our evaluation data is unavoidable. However, any overlapping text the model may have encountered during training would likely contain uncorrected blackdot words, which are not informative for the present correction task and therefore would not artificially inflate performance. Further, our goal is not to test the model's general ability to handle unseen early modern English, but to test in-corpus performance. Overlap is not a problem but an expected feature of the task.

We manually evaluate the performance of the three models on a sample of 90 blackdot words that met the criteria outlined in 4.2.2, selected via a multi-stage sampling pipeline. The samples are drawn from 50 EarlyPrint texts published before 1560. In total, these texts contain 37,510 blackdot words. We select approximately 4% of this set (1,450 instances) for initial consideration, of which 200 ultimately meet our inclusion criteria. Among the 1,250 excluded cases, roughly 20% contain more than eight candidate spellings, with an average of 68 candidates per blackdot word. Because our current implementation evaluates each candidate individually, such cases would require disproportionately large computational resources and are therefore omitted from this round

Model	MonadGPT	Llama 2	T5
Accuracy	0.78	0.79	0.42

Table 1: Comparing MonadGPT, Llama 2, and T5's performance on sampled blackdot words from pre-1560 texts

of evaluation. We further exclude 110 instances in which the correct spelling could not be confidently determined, or where meaningful evaluation is not possible – such as when fewer than two candidates existed, or when all candidates represent equally acceptable spelling variants of the same word. The resulting sample is intended to represent not the EarlyPrint corpus as a whole, but its more difficult instances. As texts from before 1560 exhibit the greatest orthographic variation relative to modern English, they are particularly suitable for evaluating model performance on the more challenging historical features of this corpus.

We compare the model's predictions with our gold labels as follows:

- gestur•/gesture: a prediction of "e" would be considered correct
- a•owing/avowing: a prediction of either "v" or "u" would be considered correct, as the two letters are interchangeable in earlier forms of Early Modern English
- conformiti•/conformity: a prediction of "e" would be considered correct, as "conformitie" is a valid spelling in Early Modern English

We find that MonadGPT and Llama 2 have similarly high performance, at 0.78 and 0.79 accuracy, respectively, whereas T5 performs significantly worse on this task (see Table 1). As MonadGPT and Llama 2's performance is quite similar, we choose to use MonadGPT for the remainder of our work, as it was fine-tuned on Early Modern texts.

4.2.4 Results

We then extend our evaluation to a larger sample of 386 blackdot words, taken from the full corpus. In this setting, we include cases with a single candidate and cases where multiple candidates are orthographic variants of the same word, in order to measure the accuracy of the entire pipeline rather than the model alone. We obtain an accuracy of 86% over all blackdot words that we labeled (Table 2).

We find that instances of transcription error are especially difficult for our system to solve:

- Confusion with punctuation: Some word-terminal blackdots represent missing punctuation instead of a missing letter. For example, in the case of "vniti••," the correction should be "vnitie" ("unity"), followed by a colon or period in the original text. However, during the transcription process, the punctuation was mistaken for a part of the word and included in the word as a blackdot character. Unrecognizable punctuation is supposed to be indicated in the corpus with a different symbol (a black square), so we do not consider these edge cases in our system.
- Confusion of letters: Certain letters that looked similar in Early Modern English print were commonly confused with each other, such as s/f, n/u, r/t, and c/e. For example, for the blackdot word "•ogie," our system produces the candidates "logie" and "bogie," but upon examination, we find that the correct spelling here should be "logic," and the final letter "c" of the text has been confused with an "e."

	Labeled BWs	Valid and labeled BWs
Top 1 prediction is correct	0.86	0.95
Correct prediction in top 3	0.90	1.00

Table 2: MonadGPT's accuracy over labeled blackdot words vs. valid and labeled blackdot words in the sample

- Word division: Blackdot words are sometimes divided unusually. For example, where the blackdot word should have been "discre••on" (discretion), it was instead broken into two words and written as "discre•• on." A few texts follow this kind of format for all their blackdot words. This becomes difficult, although not entirely impossible, for our system to predict.

- Length mismatch: In rare cases, the number of blackdot characters in a word is wrong. For example, for "p••wer," our system produces the candidates "prower" and "plower," but the correct spelling is actually "power," which would not be matched as a candidate due to having a different length. However, as these cases are quite rare, we have opted not to use "softer" regex matching (which would in most cases significantly broaden the set of low-probability candidates).

After we remove cases in which the blackdot word itself is invalid, the model's accuracy increases significantly, to 95.1%.

We run this method over the entire corpus, accumulating over 1.5 million predictions. On average, a batch of 2,300 words took one A6000 GPU 71.5 minutes to process, although variance is high. The total processing time across the entire corpus was approximately 780 hours on our campus cluster at a total imputed cost of about USD 780. The predictions will be incorporated into the EarlyPrint corpus after a more detailed review by one of the researchers.

4.3 Alternate Case (High-frequency Words): Zero-shot Prediction

Of the top 1,000 most common blackdot words in the EarlyPrint corpus, 95.1% are shorter than five characters in length. As such, most of them were excluded from our previous batch. They pose unique difficulties, not only because of their shorter length, but also because they tend to have a very large number of candidate corrections. On average, each of the top 100 most common blackdot words has around 20 candidate spellings, with a few, less frequent outliers in the hundreds ("•••e", for example, which ranks among the top 200 most common blackdot words, has 900 matches). This is much higher than our filtering criterion of eight candidates in the previous approach, and would likely result in much lower accuracy.

In a second run, we focus on these top 1,000 most common blackdot words. As noted, 95.1% have fewer than five characters; 1.4% have more than three word-initial blackdots, and 52% have more than eight candidates. For words with eight or fewer candidates, we still use our first approach, which has proven to perform at a high accuracy. For words with more than eight candidates, we iteratively predict the most likely next token that maps onto the blackdot word.

As MonadGPT uses SentencePiece tokenization, its smallest unit is often not a whole word or single character. Therefore, for each of the blackdot words in this sample, we provide the model with a left-hand-side context window of up to 512 tokens. The model then computes the most likely next token, iterating over the tokens until it reaches a complete word. For example:

Blackdot	Generative Process	Gold Label	Result
mar•yrs	_mart → _mart+yr → _mart+yr+s = martyrs	martyrs	correct
visco••t	_vis → _vis+count = viscount	viscount	correct
••sgrace	_his → _his+g → _his+g+ra → _his+g+ra+ce = hisgrace	disgrace	incorrect

Table 3: Comparing the characteristics of words processed with zero-shot evaluation and words processed with zero-shot prediction

4.3.1 Results

We evaluate this approach, along with our first approach, on 3,826 randomly sampled instances of the top 1,000 most common blackdot words from pre-1620 decades. Our first approach is applied to blackdot words with fewer than eight candidates, while our second approach is applied to those with more than eight candidates.

Of the 3,826 instances, 1,821 pass the criteria for the zero-shot evaluation approach, while 2,105 pass the criteria for the zero-shot prediction approach.

The accuracy of the first approach is 78.7%. The accuracy of the new approach is 36.7%. However, when two specific blackdot words, "••d" and "y•," are removed, the accuracy increases to 78.5%, suggesting that these are outliers that the model (like many human scholars) finds particularly difficult. For example, y• not only stands for "y" followed by some other letter, but also functions as a 16th century abbreviation for an initial "th," and with a superscript "e," "t," or "u" it can stand for "the," "that," and "thou." As such, it might require a separate approach.

We run this method on all instances of the top 1,000 most common blackdot words, to obtain over 1.3 million additional predictions. On average, each batch of around 20,000 blackdot words took 6 hours to complete. The total runtime was approximately 402 hours at an imputed cost of USD 400.

5 Conclusion

Through our experimentation, we have improved upon the 2016 LSTM-based approach by producing 1.5 million predictions at an average accuracy of 86% and an additional 1.3 million predictions at an accuracy of 79%. Of these predictions, 700,000 that matched the results of the LSTM-based approach were considered sufficiently high-confidence to incorporate directly into the working edition of the texts. The remaining predictions serve as references for scholars. We have also brought light to other errors in the corpus, such as transcription errors and incorrect "regular" attributes, during our evaluation process, further improving upon the EarlyPrint library's annotation accuracy. Improvements such as these are crucial for not only enhancing corpus quality, but also reducing barriers to scholarly engagement. Reliable digital texts significantly ease scholars' workflows, especially when original copies are difficult to consult in person. Ensuring that digitized materials meet high standards of textual accuracy improves corpus usability and builds scholarly trust.

Moreover, our experimentation shows the limitations of using the conversational MonadGPT model for zero-shot evaluation. One challenge that we encountered in our experiments was correcting whole-word errors, which are denoted by diamond symbols or sequences of blackdots in the corpus. We approached this problem by iteratively predicting the most likely next token-piece until we reached a starting token, indicating that we had arrived at a complete word. However, the method did not yield high accuracy, suggesting that it was not effective enough for reliable whole-word correction. One possible reason for this is that the conversational format of MonadGPT may not be well-suited for whole-word correction, as it was fine-tuned on conversation-like excerpts from EEBO texts. Going forward, we plan to explore alternative strategies, such as using a MonadGPT-style model that is not instruction tuned, or employing models specifically trained to handle word-level corrections, to evaluate how model performance could improve in higher-resource situations. We hypothesize that fine-tuning (or continuing to pretrain) a base version of MonadGPT's underlying Mistral (or similar) model on entire EEBO texts, rather than just conversational excerpts, could lead to improved performance for correcting whole-word errors. Additionally, using vocabulary from a narrower, domain-specific context (e.g., medical texts, sermons, poetry) might prove useful for subsets of the EarlyPrint corpus.

For certain tasks in this study, we relied on relatively small test sets due to the difficulty of manual labeling, which limits the statistical confidence of our findings. In addition, model performance might also have been affected by partial overlap between MonadGPT's training data and the EarlyPrint collections. If the model encountered blackdot words or other transcription defects during training, such exposure could influence its behavior, potentially reducing its effectiveness on the correction task rather than inflating performance. Future work could expand both the scale and the scope of evaluation to test these methods on non-overlapping data.

Another significant challenge to improving the accuracy of the EarlyPrint library is the handling of "unknown unknowns." While we have shown that "known unknowns" can be corrected in many cases with good accuracy through zero-shot methods, "unknown unknowns" present a more complex issue, as their presence is not immediately obvious even to human annotators. Future work could focus on developing more advanced error detection and correction methods for these cases.

Acknowledgements

The research was supported in part by the National Endowment for the Humanities (#HAA-290374-23, "AI for Humanists," to MW).

References

[1] Anonymous. *[Thystorye and lyf of the noble and crysten prynce Charles the grete kynge of Frauuce [sic]]*. English. Keyboarded and encoded full text © 2003-2004 Early English Books Online Text Creation Partnership. All Rights Reserved; Analyte descriptor - Translated by William Caxton from the French. London: Printed by William Caxton, 1485, 192 p. URL: https://www.proquest.com/books/thystorye-lyf-noble-crysten-prynce-charles-grete/docview/2240929674/se-2.

[2] Assael, Yannis, Sommerschield, Thea, and Prag, Jonathan. "Restoring ancient text using deep learning: a case study on Greek epigraphy". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 6368–6375. DOI: 10.18653/v1/D19-1668.

[3] Assael, Yannis, Sommerschield, Thea, Shillingford, Brendan, Bordbar, Mahyar, Pavlopoulos, John, Chatzipanagiotou, Marita, Androutsopoulos, Ion, Prag, Jonathan, and Freitas, Nando de. "Restoring and attributing ancient texts using deep neural networks". In: *Nature 603*, no. 7900 (2022), pp. 280–283. DOI: 10.1038/s41586-022-04448-z.

[4] Bollmann, Marcel, Korchagina, Natalia, and Søgaard, Anders. “Few-Shot and Zero-Shot Learning for Historical Text Normalization”. In: *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019),* ed. by Colin Cherry, Greg Durrett, George Foster, Reza Haffari, Shahram Khadivi, Nanyun Peng, Xiang Ren, and Swabha Swayamdipta. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 104–114. DOI: `10.18653/v1/D19-6112`.

[5] Boros, Emanuela, Ehrmann, Maud, Romanello, Matteo, Najem-Meyer, Sven, and Kaplan, Frédéric. “Post-Correction of Historical Text Transcripts with Large Language Models: An Exploratory Study”. In: *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024),* ed. by Yuri Bizzoni, Stefania Degaetano-Ortlieb, Anna Kazantseva, and Stan Szpakowicz. St. Julians, Malta: Association for Computational Linguistics, 2024, pp. 133–159.

[6] De Toni, Francesco, Akiki, Christopher, De La Rosa, Javier, Fourrier, Clémentine, Manjavacas, Enrique, Schweter, Stefan, and Van Strien, Daniel. “Entities, Dates, and Languages: Zero-Shot on Historical Texts with T0”. In: *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models,* ed. by Angela Fan, Suzana Ilic, Thomas Wolf, and Matthias Gallé. virtual+Dublin: Association for Computational Linguistics, 2022, pp. 75–83. DOI: `10.18653/v1/2022.bigscience-1.7`.

[7] EarlyPrint. “About”. `https://earlyprint.org/about/`. Accessed October 2025. 2025.

[8] EarlyPrint. “EarlyPrint Library”. `https://texts.earlyprint.org/exist/apps/shc/home.html`. Accessed October 2025. 2025.

[9] Estrella, Paula and Paliza, Pablo. “OCR correction of documents generated during Argentina’s national reorganization process”. In: *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*. DATeCH ’14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 119–123. DOI: `10.1145/2595188.2595194`.

[10] Gale. “Eighteenth Century Collections Online”. `https://www.gale.com/primary-sources/eighteenth-century-collections-online`. Accessed October 2025. 2025.

[11] Gotscharek, Annette, Reffle, Ulrich, Ringlstetter, Christoph, and Schulz, Klaus U. “On lexical resources for digitization of historical documents”. In: *Proceedings of the 9th ACM symposium on Document engineering*. Munich Germany: ACM, 2009, pp. 193–200. DOI: `10.1145/1600193.1600236`.

[12] Kang, Kyeongpil, Jin, Kyohoon, Yang, Soyoung, Jang, Soojin, Choo, Jaegul, and Kim, Youngbin. “Restoring and Mining the Records of the Joseon Dynasty via Neural Language Modeling and Machine Translation”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* ed. by Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou. Online: Association for Computational Linguistics, 2021, pp. 4031–4042. DOI: `10.18653/v1/2021.naacl-main.317`.

[13] Kettunen, Kimmo. “Keep, Change or Delete? Setting up a Low Resource OCR Post-correction Framework for a Digitized Old Finnish Newspaper Collection”. In: *Digital Libraries on the Move,* ed. by Diego Calvanese, Dario De Nart, and Carlo Tasso. Cham: Springer International Publishing, 2016, pp. 95–103. DOI: `10.1007/978-3-319-41938-1_11`.

[14] Langlais, Pierre-Carl. “Pclanglais/MonadGPT · Hugging Face”. https://huggingface.co/Pclanglais/MonadGPT. 2023.

[15] Loewenstein, Joseph and Araghi, Alireza Taheri. “EEBO and EEBO-TCP: A Brief Introduction”. https://earlyprint.org/intros/intro-to-eebo-and-eebo-tcp.html. 2021. (Visited on 01/27/2025).

[16] Manjavacas Arevalo, Enrique and Fonteyn, Lauren. “MacBERTh: Development and Evaluation of a Historically Pre-trained Language Model for English (1450-1950)”. In: *Proceedings of the Workshop on Natural Language Processing for Digital Humanities*, ed. by Mika Hämäläinen, Khalid Alnajjar, Niko Partanen, and Jack Rueter. NIT Silchar, India: NLP Association of India (NLPAI), 2021, pp. 23–36.

[17] Michigan Library, University of. “Evans Early American Imprint Collection”. https://quod.lib.umich.edu/e/evans/. Accessed October 2025. 2025.

[18] Mueller, Martin. “Digital Images for EarlyPrint Texts”. https://earlyprint.org/posts/digital-images-for-earlyprint-texts.html. 2020.

[19] Mueller, Martin. “The TCP Texts and their Shortcomings”. https://earlyprint.org/intros/about-ep-texts.html. 2019.

[20] Nguyen, Thi Tuyet Hai, Jatowt, Adam, Coustaty, Mickael, and Doucet, Antoine. “Survey of Post-OCR Processing Approaches”. In: *ACM Computing Surveys 54*, no. 6 (2022), pp. 1–37. DOI: 10.1145/3453476.

[21] Petricca, Francesca. “LibGuides: Early English Books Online (EEBO) on the ProQuest Platform: Content”. https://proquest.libguides.com/eebopqp/content. Accessed October 2025. 2025.

[22] SangrinLee. “SangrinLee/Oldbook_Project”. https://github.com/SangrinLee/Oldbook_Project. 2023.

[23] Sanh, Victor et al. “Multitask Prompted Training Enables Zero-Shot Task Generalization”. In: *Proceedings of the International Conference on Learning Representations (ICLR 2022)*. Online, 2022.

[24] Siefring, Judith and Meyer, Eric T. “Sustaining the EEBO-TCP Corpus in Transition: Report on the TIDSR Benchmarking Study”. SSRN, https://ssrn.com/abstract=2236202. Rochester, NY, 2013. DOI: 10.2139/ssrn.2236202.

[25] Thomas, Alan, Gaizauskas, Robert, and Lu, Haiping. “Leveraging LLMs for Post-OCR Correction of Historical Newspapers”. In: *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024*, ed. by Rachele Sprugnoli and Marco Passarotti. Torino, Italia: ELRA and ICCL, 2024, pp. 116–121.