Logion: Machine-Learning Based Detection and Correction of Textual Errors in Greek Philology

Charlie Cowen-Breen[1]**, Creston Brooks**[2]**, Johannes Haubold**[3]**, Barbara Graziosi**[3]
[1]Department of Pure Mathematics and Mathematical Statistics, University of Cambridge
[2]Department of Computer Science, Princeton University
[3]Department of Classics, Princeton University

Abstract

We present statistical and machine-learning based techniques for detecting and correcting errors in text and apply them to the challenge of textual corruption in Greek philology. Most ancient Greek texts reach us through a long process of copying, in relay, from earlier manuscripts (now lost). In this process of textual transmission, copying errors tend to accrue. After training a BERT model on the largest premodern Greek dataset used for this purpose to date, we identify and correct previously undetected errors made by scribes in the process of textual transmission, in what is, to our knowledge, the first successful identification of such errors via machine learning. The premodern Greek BERT model we train is available for use at [X].

1 Introduction

Ancient texts have been passed down by scribes over hundreds of years, in a process known as textual transmission. Scribes occasionally make mistakes, some of which lie undiscovered to this day. As unchecked errors have the potential to change the meaning of a text, finding and correcting scribal errors is a central aim in Greek philology.

In a proof-of-concept paper, we presented the first scribal mistakes detected by contextual language models (Anonymous, 2023). In this paper, we describe and study the approaches used to arrive at those results and evaluate the algorithm's effectiveness on artificially generated errors.

Prior to Anonymous (2023), to the best of our knowledge, scribal errors were found only by hand—that is, with domain experts carefully reading the text until they find potential errors, and then using database searches to assess textual problems and propose solutions. These errors include simplifying difficult expressions, omissions, replacing one word for another with a similar sound, shape, or function, etc. Discovery of such errors typically requires a sophisticated understanding of an author's writing tendencies and the context of a particular text.

This motivates the use of contextual language models for the detection of scribal errors. In this paper, we propose *Logion*, a framework for detecting scribal errors based on contextual language models.[1] Logion consists of three stages: in the first stage, a contextual language model learns conditional word distributions for a selected corpus; in the second stage, potential errors are identified according to statistics derived from the learned distribution; lastly, in the third stage, corrections are proposed for the words identified as potentially erroneous. While not all words flagged by the algorithm will be genuine scribal errors, a "shortlist" of potential scribal errors can point philologists to previously undetected errors which, after being corrected, restore the original meaning.

To summarize, our main contributions are as follows:

- We present a premodern Greek BERT trained with what we believe to be the largest database used for this purpose to date.
- We propose Logion, a framework for scribal error detection and emendation based on contextual language models.
- We study the effectiveness of Logion at detecting artificially generated scribal errors, and showcase real errors which it has already discovered.

In this paper, we concentrate on the discovery of scribal errors in the works of the Byzantine author Michael Psellus, who is a convenient choice at a proof-of-concept stage for philological reasons. However, we remark that these methods may be applied to any premodern text passed down by scribes, provided sufficient data is available.

[1] The name "Logion" derives from an ancient Greek word meaning "oracle."

2 Related Work

In a study also related to the restoration of premodern Greek, Assael et al. (2022) train a multi-task transformer-based model to date, place, and fill gaps in ancient Greek inscriptions. Inscriptions display the original text on stone, pottery, or other media, whereas most of what survives from antiquity reaches us via a long tradition of hand-copying from earlier exemplars. For this reason, Assael et al. (2022) focus on gaps in inscriptions caused by physical damage but not on copying errors in texts.

In English and other modern languages, previous work on textual error detection has typically focused on spelling and grammar checking (Etoori et al., 2018) (Ganiz et al., 2020) (Naber, 2003), while textual errors introduced by scribes are often more complex (e.g. Figure 4). For this reason, identifying scribal errors more closely aligns with *out-of-distribution* detection, in which the task is to discern whether samples—in our case, words—are likely to have been generated by a given distribution—in our case, the author's body of work—or instead are out of distribution—i.e., the result of an error in transmission. Ren et al. (2019) propose the use of likelihood ratios to determine out-of-distribution samples for images and genomic sequences, a metric which we slightly modify. Sometimes error detection is validated by philological experts; at other times it is confirmed directly by manuscripts that were either sidelined or misread by previous scholars in the course of preparing the first or subsequent printed editions. To our knowledge, this paper is the first to identify and correct scribal errors via machine learning.

3 Methodology

Logion is a three-stage framework for the discovery and emendation of textual errors in a given corpus. The initial stage involves training a BERT model, which undergoes broad pre-training on premodern Greek text followed by subsequent fine-tuning on specific works of interest, as outlined in subsection 3.1.

The second and third stages harness the learned distributions of the fine-tuned BERT to detect and emend errors, respectively. Before describing the later stages in full, we briefly recount the conditional distributions which BERT learns. Given a sequence of tokens $w_1, \ldots, w_n$, consider a single token w_i and denote the surrounding (bidirectional) context by $w_{-i} = (w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)$. From the masked-language model (MLM) training task, the model learns the distribution

$$p(w|w_{-i}) \tag{1}$$

over tokens w which occur in the i^{th} position of a sentence when surrounded by context w_{-i} (Devlin et al., 2019). For inference on words comprised of multiple tokens, we extend p to a distribution over sequences of tokens via beam search. Therefore, in what follows, when $(w_1, \ldots, w_k)$ is a sequence of words, rather than tokens, we will let $p(w|w_{-i})$ denote the corresponding distribution over words w which is derived from Expression 1 via beam search.

In the second and third stages, described in subsection 3.2 and subsection 3.3, respectively, existing statistical theory is applied to the learned distribution p to determine the tokens which are most likely to contain errors, and subsequently to propose emendations. The stages are illustrated together in Figure 1.

3.1 BERT Training

We initially trained a BERT model on a dataset of 6.4 million words of premodern Greek, which we gratefully received from Pranaydeep Singh. This is the base model used in Anonymous (2023). Singh et al. (2021) assembled this data from open-source databases, such as the Perseus Digital Library and the First1KGreek corpus, in the course of training a BERT model for ancient and medieval Greek. We subsequently assembled a much larger dataset of roughly 70 million words.[2] We divided this data into a 90-10 train-test split and trained the BERT model using two NVIDIA A100 GPUs for 200 epochs until validation loss stabilized. To prepare the tokenized input, we maximized the amount of punctuation-separated text in each input, up to a limit of 512 input tokens. We used a batch size of 16 and a mask ratio of 0.15.[3]

[2] Some texts were kindly provided to us by the Director of the Thesaurus Linguae Graecae to fine-tune the model but not to disseminate further; other texts were provided to us by colleagues who would be happy for us to share them with any interested parties. We can, therefore, share some but not all of our training data upon request: we are constrained by the license currently restricting access to the Thesaurus Linguae Graecae.

[3] At the task of 1-token prediction on our test set, the model achieves 84.4% top-1 accuracy and 95.2% top-5 accuracy, and obtains a low pseudo-perplexity of 2.162 (Wang and Cho, 2019). We note that these metrics are dependent on specific tokenizations and should not be compared to models with different tokenizations.

To evaluate the impact of Singh et al.'s pretraining on modern Greek, we trained two models, one with random initialization and one initialized from Singh et al. (2021)'s Ancient Greek BERT. Both times, we used Singh et al. (2021)'s tokenizer which had been created for Modern Greek subwords, since they themselves fine-tuned a Modern Greek BERT. We find that both trainings converge to the same validation loss after a small number of epochs, indicating no discernible benefit from pretraining on Modern Greek. A future model may be more effective with a tokenization optimized for Ancient Greek: see section 6. The resulting premodern Greek BERT model is available for use at [X].

To better learn the distribution of particular works in which we would like to identify errors, we then perform a fine-tuning of the broadly trained premodern Greek BERT. We partition selected works into a 90-10 train-test split and continue training using the MLM objective until validation loss stabilizes.

3.2 Error Detection

In this section, we show how certain metrics derived from the distribution p learned by BERT function as indicators of the likelihood that a given word contains an error.

Given a corpus, the goal is to flag words which are most likely to be erroneous, in order to provide domain experts with a shortlist of potential errors and emendations. A word is flagged if it satisfies certain conditions based on the metrics we define below.

3.2.1 Metrics

We propose three metrics for flagging potential errors. Additional metrics may achieve higher accuracy at error detection in the future.[4] That said, the metrics presented here have the benefit of certain theoretical guarantees, as shown by Proposition 1 in subsubsection 3.2.3.

1. Given a word w_i with (bidirectional) context w_{-i}, the **chance** of word i is defined as
$$p(w_i|w_{-i})$$
that is, the probability that the word exists in its given context, as determined by the model.

2. The model's **confidence** at word i is defined as
$$\max_{\text{word } w} p(w|w_{-i})$$
that is, the probability of the top suggested replacement in the given context around position i, as determined by the model.

3. The **scribal distance** at word i is defined as
$$d\left(w_i, \quad \operatorname*{argmax}_{\text{word } w} p(w|w_{-i})\right)$$
where $d(x, y)$ denotes the Levenshtein distance between strings x and y.

3.2.2 Rare Words

While low chance may seem to be the most intuitive indicator of errors, we find that the other two metrics are helpful for avoiding false positives. If chance were the only metric considered, genuine but rare words would be incorrectly flagged as errors.[5] Moreover, scribal errors are sometimes graphically or phonetically similar to the correct text. Thus, we choose to flag low-chance portions of text which are close in sound or shape to high-confidence model suggestions. As experimentally demonstrated in subsection 4.3, accuracy at detecting artificially generated errors is greatly improved by considering both chance and confidence, in comparison to using either metric alone.

3.2.3 Combining Metrics

Depending on the application of interest, one can combine metrics in various ways to generate error flags. In what follows, we present two ranking schemes that appear to be effective at finding either real scribal errors or artificial errors introduced in order to test the effectiveness of our approach.

Chance-confidence ratio rankings

Suppose that we are given a sequence of words $s = (w_1, \ldots, w_n)$. As a measure of likelihood for the i^{th} word to be an error, we propose the quantity

$$\rho_i(s) := \frac{p(w_i|w_{-i})}{\max_{w \in \mathcal{W}_k(w_i)} p(w|w_{-i})} \qquad (2)$$

[4] These metrics are certainly not the only ones that would lead a human philologist to consider a word suspicious, but they serve for now as a useful tool, as evidenced by Anonymous (2023). In the future, we expect that more end-to-end methods—such as training for detecting errors directly—and regressions accounting for more metrics will outperform what is shown here.

[5] This is because chance considers only the absolute probability of a word w_i in context w_{-i}, instead of the relative probability when compared to plausible alternatives. Such relative probabilities are achieved by the chance-confidence ratio, which we present in the next section.

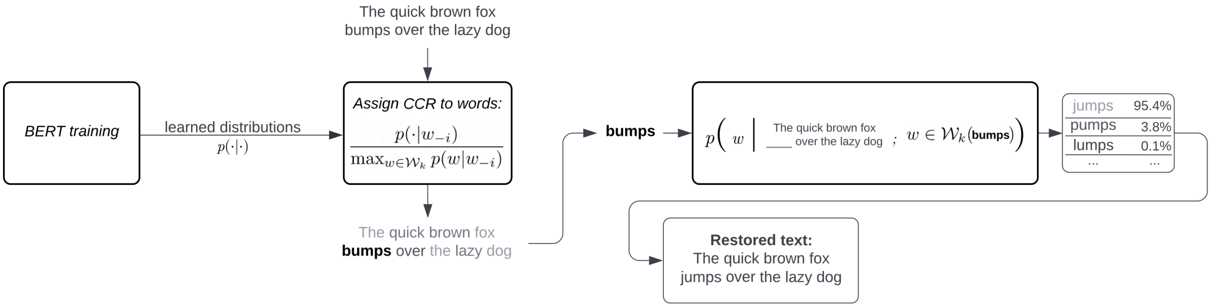


Figure 1: Logion pipeline. Here, the given text has been corrupted with a change from "jumps" to "bumps." In the first stage (left), a BERT model is fine-tuned to learn $p(\cdot|w_{-i})$ for a given corpus. In the second stage (middle), to identify the error, Logion computes the CCR of each word in the sentence (this is depicted as the brightness of each word) and identifies "bumps" as having the lowest CCR. In the third stage (right), to correct the error, a change from "bumps" to "jumps" is proposed based on the learned distribution, when restricted to words which are one-character modifications of "bumps" (here, $k = 1$). Error and emendation proposals are then vetted by domain experts.

where

$$\mathcal{W}_k(x) = \{y : d(x, y) \leq k\}$$

and k is a fixed positive integer. To motivate $\rho_i(s)$ as a measure of the likelihood that the i^{th} word is erroneous, we note that, intuitively, $\rho_i(s)$ is small when its numerator is small and its denominator is large, i.e. when word i has low chance and is close in Levenshtein to a high-confidence model suggestion. By the discussion in subsubsection 3.2.2, then, we expect erroneous words to correlate with words for which $\rho_i(s)$ is small.

We will refer to $\rho_i(s)$ as the **chance-confidence ratio** (CCR) of the i^{th} word of s. This name derives from the fact that if the distributions $p(\cdot|w_{-i})$ used to compute chance and confidence are further conditioned on the event that $d(\cdot, w_i) \leq k$, then the ratio of the (conditioned) chance and confidence is equal to $\rho_i(s)$.

One natural motivation for the CCR is the following: suppose we are allowed to change only one character of a sentence and want to do so in such a way that it most resembles what a given author has written. Then, the character which we should change is exactly the one which would result in the smallest CCR of the affected word. This is formalized in the following proposition, which we prove in the Appendix.[6]

Proposition 1. (Correspondence between CCR and relative probabilities of sentences) *Let $p(s)$ be a joint distribution on sentences s. Given a sentence s, suppose that*

$$s^* = \operatorname*{argmax}_{s' \in \mathcal{W}_1(s)} p(s')$$

Then $s^* = s$ if and only if $\rho_i(s) > 1$ for all i. Moreover, if $s^* \neq s$ and i^* is the word index at which s^* differs from s, then

$$i^* = \operatorname*{argmin}_i \rho_i(s)$$

Furthermore, s^* is obtained by replacing w_{i^*} with the model top suggestion at i^* restricted to $\mathcal{W}_1(w_{i^*})$.

In other words, the proposition states that, assuming a joint probability distribution exists,[7] the CCR indicates the one-character alteration of s which the model determines most likely to have been written by the author.[8] This motivates ranking words by their CCR (i.e., by the values $\rho_i(s)$) in order to detect plausible errors. In section 4, we artificially generate errors and find that the word with index

$$\operatorname*{argmin}_i \rho_i(s)$$

indeed contains an error 90% of the time, showing that such rankings are effective at detecting artificially generated errors (see Table 1 and Figure 3). Moreover, in 98% of such instances, the top model suggestion at the erroneous word w_i,

[6] For alterations of $k > 1$ characters, the proposition generalizes to the corresponding statement with the assumption instead that s' lies in the set of all sentences which differ from s in a single word by at most k characters.

[7] There do not appear to be theoretical guarantees for this in the literature. Wang and Cho (2019) attempt to construct a joint distribution $p(s)$ directly by showing that BERT is a Markov random field, but the paper has since been retracted.

[8] That said, care must be taken in concluding that s^* was the original formulation of the author. Scribal errors may skew toward easier readings of the text and may thus increase p. This is an effect we consider further in section 6.

$\text{argmax}_{w \in \mathcal{W}_1(w_i)} p(w|w_{-i})$, recovers the correct ground-truth word.

Another interpretation of ρ_i is that it is the likelihood-ratio statistic, assuming the prior on w which is uniform on $\mathcal{W}_k$ and vanishes elsewhere. In this sense, the CCR builds on Ren et al. (2019) and Gangal et al. (2020), which achieved success at detecting out-of-distribution samples with the likelihood-ratio statistic. This interpretation amounts to treating the ground truth word at position i as an unknown parameter w, the value of which determines the conditional distribution $p(w_{-i}|w)$ of the surrounding words. In this case—again assuming that scribes only make errors which do not exceed a Levenshtein distance of k from the original text—we can formulate error detection as the hypothesis testing problem

H_0 : The word w_i is correct as written.
H_1 : The original word has been altered and lies in $\mathcal{W}_k \setminus \{w_i\}$.

The corresponding likelihood-ratio statistic for this hypothesis test is given by

$$\frac{p(w_{-i}|w_i)}{\max_{w \in \mathcal{W}_k} p(w_{-i}|w)}$$

In a Bayesian framework with uniform prior on $\mathcal{W}_k$, one can see that this is equivalent to the CCR. In Figure 2 (i), we plot the distribution of the likelihood-ratio statistic under the hypotheses H_0 and H_1. The distributions under each hypothesis are distinct, allowing for formal hypothesis testing via the likelihood-ratio test.

Thresholding

In some applications, thresholding for each metric individually can provide more flexibility for generating a shortlist of errors. In Anonymous (2023), the results were generated by thresholding for confidence of at least 50%, scribal distance at most 3, and ranking the remaining words in order of increasing chance. A selection of flags resulting from this scheme is shown in section 5. The choice of a 50% threshold for confidence is convenient because it respects the property that, among words which pass the threshold, the model's top suggestion is the same before and after thresholding for scribal distance.

Thresholds determine the precision and recall of the model when it is used to identify erroneous words. For applications where one wishes to find a list of strong candidates for erroneous words (i.e. high precision is desirable), one can set the confidence threshold to be high (e.g. 90%) and the chance and scribal distance thresholds to be low (e.g. 10^{-6} and 2, respectively). For applications in which one wishes to find more corrupted words and can tolerate sifting through weaker candidates (i.e. high recall is desirable), one can set the confidence threshold to be low (e.g. 50%) and the chance and scribal distance thresholds to be high (e.g. 10^{-4} and 4, respectively).

3.3 Emendation

Once a subset of the corpus has been flagged as potentially erroneous, we can easily propose emendations via Proposition 1. In the case $k = 1$, for example, Proposition 1 suggests that the highest probability one-character alteration of the input text is found by replacing the flagged word (say, w_i) with the model top suggestion at position i when restricted to only one-character alterations:

$$\underset{w \in \mathcal{W}_1(w_i)}{\text{argmax}}\ p(w|w_{-i})$$

This is the scheme which is employed for the experiments in the following section. Since producing more than one suggested emendation can be helpful for domain experts, in practice, we report any number of the most likely words $w \in \mathcal{W}_k(w_i)$ according to the distribution

$$p(w|w_{-i})$$

for any $k \geq 1$.

4 Experiments

In this section, we study the effectiveness of the proposed approach at finding artificially generated errors, while noting that the proposed approach has already resulted in the discovery of real errors, as outlined in Anonymous (2023). A sample of that work is shown in section 5.

4.1 Artificially Generated Errors

Artificially generating scribal errors is made difficult by the fact that the data-generating mechanism is inherently complex and difficult to reproduce. Such errors are often dependent on individual scribes, the context in which they were working, and their interest in what they were copying: scribal errors can be quite varied and complex.

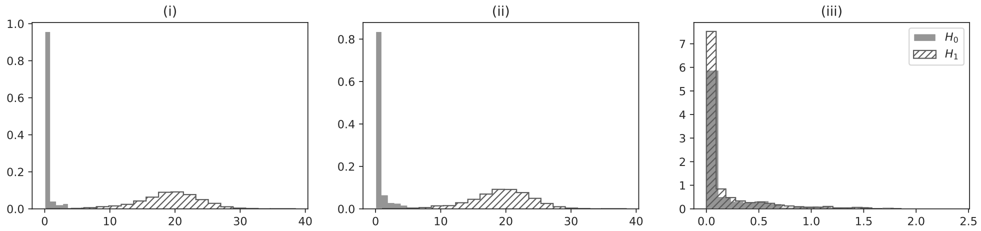


Figure 2: Distribution of metrics under hypotheses H_0 and H_1. The metrics shown are (i) CCR as in Expression 2, (ii) chance, (iii) confidence. Each horizontal scale is $-\log T$, where T is the metric associated with the plot. Here, H_1 is modeled by the scheme to generate artificial errors described in section 4, where we restrict to only single token replacements in order to produce samples efficiently. Each plot contains roughly 500,000 samples from H_0 and 1,000 samples from H_1.

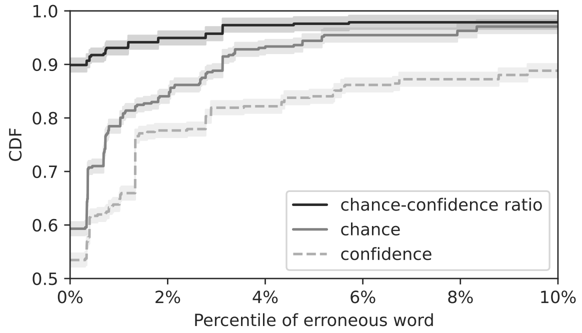


Figure 3: Artificial errors are inserted into one word per paragraph (average 230 words). The metrics of every word in the paragraph are computed (CCR, chance alone, confidence restricted to scribal distance 1), and the percentile of the erroneous word is measured when ranked by metric within the paragraph. This plot shows the cumulative distribution function of these percentiles. 90.5% of words rank first (i.e. 0^{th} percentile) in their paragraph by CCR, and 59.7% of words rank first by chance, in agreement with Table 1. Error bands are computed via the DKW inequality with 99% coverage probability.

That said, some errors are fairly banal, such as changes in pronunciation that can result in spelling errors due to phenomena such as itacism.[9] For the purpose of this simulation, we generate scribal errors in the following manner: within every paragraph, we replace a randomly chosen character with another random character such that the modified word is in the dictionary of words used by the author at least ten times. If the modified word does not meet this criterion, we continue substituting characters until it does. This process ensures that a simple dictionary check could not catch the errors we generate.

4.2 Results

Within every paragraph, we rank words by CCR (Equation 2), as described in subsubsection 3.2.3 with $k = 1$. Out of 615 randomly generated instances, the erroneous word ranked first 556 times, yielding a 90.5% top-1 accuracy. Among instances in which the erroneous word ranked first, the ground-truth word was the top suggested replacement for the erroneous word 98.1% of the time. The results are summarized in Table 1.

4.3 Ablation Study

To demonstrate that consideration of all three metrics introduced in Section 2.3 improves accuracy at detecting artificial errors, here we compare ranking by CCR to two alternative ranking schemes which do not involve all three metrics: (i) ranking by confidence when restricted to scribal distance 1, and (ii) ranking by chance alone.

Figure 3 shows the distribution of artificially corrupted words when ranked by the ranking schemes proposed. The distribution of each ranking scheme is heavily left-skewed: more than 85% of erroneous words lie in the bottom 10% of words when ranked by any metric. This suggests that each of the rankings proposed correlates with artificial errors.

However, when ranking by either scheme (i) or scheme (ii), the corrupted word is ranked first in less than 60% of cases; in comparison, the CCR metric ranks the corrupted word first in 90.5% of cases (see Table 1). Therefore, we conclude that ranking words by either scheme (i) or scheme (ii) is less effective in identifying artificial errors than ranking by the CCR.

[9] The term itacism describes a confusion between different vowels and diphthongs, all of which came to be pronounced /i/.

Accuracy	CCR	Chance alone	Confidence alone
Top-1	90.5%	59.7%	54.2%
Top-5	95.9%	88.2%	81.1%
Top-10	97.6%	93.1%	83.5%

Table 1: Accuracy at detecting a single artificial error out of 230 words according to different schemes of combining metrics. Best performance is achieved by using CCR, although chance is also a viable metric. Since the task is to generate shortlists of potential errors for review (and domain experts can often verify quickly whether a flagged word is a true error) top-10 accuracy is a significant metric here.

5 Philologically Significant Results

The metrics presented here have successfully identified errors that were previously undetected, ranging from scribal errors in the manuscripts, typographical errors in printed editions, and errors caused by digitization. These findings underwent philological peer review and have been accepted for publication in [X], the research journal [X].

In Anonymous (2023), we show at proof-of-concept stage how the approaches introduced here improve on previous knowledge of premodern Greek texts by identifying and sometimes solving several different philological problems. For detailed examples and further discussion, please see Anonymous (2023). Below, we offer a single example to illustrate one type of error which may be detected (in this case a misreading of the manuscript on the part of modern scholars rather than an actual error in the manuscript itself).

In Psellus's *Hist. brev.* at lines 81.89–90, Aerts' edition reads:

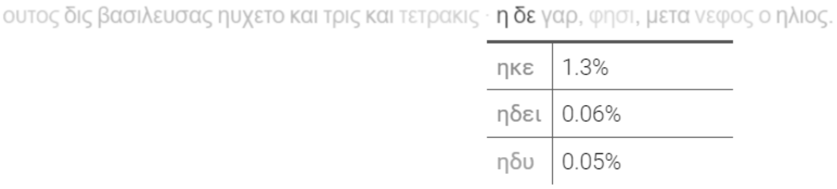
> οὗτος δὶς βασιλεύσας ηὔχετο καὶ τρὶς καὶ τετράκις· ἧ δὲ γάρ, φησι, μετὰ νέφος ὁ ἥλιος.
> Houtos dis basileusas ēucheto kai tris kai tetrakis: ē de gar, fēsi, meta nephos, ho hēlios.
> 'This man, having been king twice, prayed for a third and fourth term. For indeed, he said, there is sun after clouds.'

When thresholding for confidence and scribal distance, the token δε was one of the lowest chance tokens in the test set. The algorithm output depicted in Figure 4 and the subsequent examination of the manuscript on which this edition is based led to the realization that the manuscript actually reports "ἡδὺ", not "ἧ δὲ". The sentence can now be translated as follows: "This man, having been king twice, prayed for a third and fourth term. For, he said, 'sun after clouds is sweet'." In this case, then, the error turned out to be not in the early manuscript but in subsequent readings of it.

6 Future Work

One major line of future work concerns developing an application which is adopted by domain experts and used to assist their work. Given any text, such an application would be capable of generating shortlists of suspected errors and proposed emendations for review. Future research directions in this area include developing efficient and linguistically motivated sub-word tokenization schemes and the capability to include or exclude sections of the dataset from consideration at inference time: this is relevant when one is interested in performing error detection on a section of text which was included in the training set without retraining the model entirely. In working towards the latter goal, one promising architecture is DEMix, which enables dynamic expert mixtures at inference time (Gururangan et al., 2021). Another idea for future work, and one which sets scribal error detection apart from traditional error detection, concerns treating scribal modifications as diffusion processes. As scribal errors are often contextually driven, text altered by scribes may paradoxically evaluate to having higher probability than the original text.[10] On this view, then, the text evolves over time as a diffusion process with a transition kernel derived from p (for example, one option is to model the trajectory of the text by Gibbs sampling according to the conditionals $p(w_i|w_{-i})$). Diffusion models are designed to recover original data from diffused data, so it may be fruitful to apply such models for recovery of original text from scribally-modified text (Sohl-Dickstein et al., 2015). While not itself a diffusion model, ELECTRA is a promis-

[10] In philology, this is the principle known as *lectio difficilior potior*. Because "the normal tendency is to simplify, to trivialize, to eliminate the unfamiliar word or construction," the more difficult reading (i.e., *lectio difficilior*) should normally be taken to be the authentic one (West, 1973).



ουτος δις βασιλευσας ηυχετο και τρις και τετρακις · η δε γαρ, φησι, μετα νεφος ο ηλιος.

ηκε	1.3%
ηδει	0.06%
ηδυ	0.05%

Figure 4: Algorithm output that led to the discovery of a scribal error in the words η δε. *Top line:* words are given a grayscale color according to their CCR, as in Equation 2; the word δε was flagged by the algorithm because it obtained the smallest CCR of all words in its given context (the surrounding 512 tokens, not all of which are pictured here). *Below top line:* algorithm-generated suggestions, given a grayscale color according to their likelihood. In each case, the algorithm suggests merging two words by deleting the space before δε. The third suggestion, ηδυ, is, in fact, transmitted in the relevant manuscript and must be what was originally written by the author (Anonymous, 2023). The small probability awarded here reflects the complexity of scribal errors. Some are trivial, including the ones we generate artificially; others, including this one, are harder to emend.

ing architecture for such future work (Clark et al., 2020). ELECTRA learns representations by simultaneously training a generator, designed to corrupt input text with plausible replacement tokens, and a discriminator, which classifies tokens in the input sequence as either replaced or original. Future work may, therefore, involve extending ELECTRA to situations in which input tokens are not randomly replaced once by the generator, but many times in a row, simulating the diffusion view of scribal error generation. By enforcing further that replaced tokens are chosen not only contextually but also according to visual and phonetic similarity, it may be possible to use such an architecture to model and detect scribal errors simultaneously.

7 Conclusion

In this study, we have trained a BERT model to support philological work on premodern Greek texts: in particular, we use statistical and machine-learning-based approaches to identify scribal errors that accrue in the process of textual transmission and to propose emendations. In a broader sense, this research aims to contribute to the future of philology, understood as a discipline concerned with preserving, elucidating, and making publicly accessible the global archive of premodern texts. Some of what we have presented here is of relevance also for authors and languages we have not considered, as well as for modern text editing in general.

References

Anonymous. 2023. The title of the article. *Journal Name*.

Y. Assael, T. Sommerschield, B. Shillingford, et al. 2022. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603:280–283.

K. Clark et al. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the 8th International Conference on Learning Representations*, Addis Ababa, Ethiopia. International Conference on Learning Representations.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

P. Etoori, C. Manoj, and M. Radhika. 2018. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Student Research Workshop*, pages 146–152. Association for Computational Linguistics.

V. Gangal, A. Arora, A. Einolghozati, and S. Gupta. 2020. Likelihood ratios and generative classifiers for unsupervised out-of-domain detection in task oriented dialog. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*, pages 7764–7771. Association for the Advancement of Artificial Intelligence.

M. Can Ganiz et al. 2020. Grammar and spell checking for turkish language. Cse4197 – analysis and design document, T.C. Marmara University, Faculty of Engineering, Computer Engineering Department.

S. Gururangan et al. 2021. Demix layers: Disentangling domains for modular language modeling.

D. Naber. 2003. A rule-based style and grammar checker. Diplomarbeit Technische Fakultät, Universität Bielefeld.

J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. A. DePristo, J. V. Dillon, and B. Lakshminarayanan. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, volume 19, pages 14640–14651, Red Hook, NY. Curran Associates Inc.

P. Singh, G. Rutten, and E. Lefever. 2021. A pilot study for bert language modelling and morphological analysis for ancient and medieval greek. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 128–137. Association for Computational Linguistics.

J. Sohl-Dickstein et al. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265. PMLR.

A. Wang and K. Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov Random Field Language Model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis. Association for Computational Linguistics.

M. L. West. 1973. *Textual Criticism and Editorial Technique Applicable to Greek and Latin Texts*. Teubner, Stuttgart.

Appendix

Proof of Proposition 1.

$$\begin{aligned}\max_{s'\in\mathcal{W}_1(s)} p(s') &= \max_{1\leq i\leq n}\max_{w\in\mathcal{W}_1(w_i)} p(w_1,\ldots,w,\ldots w_n)\\ &= \max_{1\leq i\leq n}\max_{w\in\mathcal{W}_1(w_i)} p(w|w_{-i})p(w_{-i})\\ &= \max_{1\leq i\leq n}\max_{w\in\mathcal{W}_1(w_i)} \frac{p(w|w_{-i})p(w_{-i})}{p(w_i|w_{-i})p(w_{-i})}p(s)\\ &= p(s)\max_{1\leq i\leq n}\max_{w\in\mathcal{W}_1(w_i)} \frac{p(w|w_{-i})}{p(w_i|w_{-i})}\\ &= p(s)\max_{1\leq i\leq n} \quad \frac{1}{\rho_i(s)}\end{aligned}$$

which establishes that, for $s^* \in \mathcal{W}_1(s)$,

$$p(s^*) = \max_{s'\in\mathcal{W}_1(s)} p(s')$$

if and only if s^* differs from s in word i^* and

$$\rho_{i^*}(s) = \min_i \rho_i(s).$$

On the other hand, we have

$$\begin{aligned}\rho_i(s) &= \frac{p(w_i|w_{-i})}{\max_{w\in\mathcal{W}_1(w_i)} p(w|w_{-i})}\\ &= \min_{s'\in\mathcal{W}_1(s):s'\text{ differs from }s\text{ at word }i} \frac{p(s)}{p(s')} > 1\end{aligned}$$

if and only if $\forall s'$ such that s' differs from s only in word i, and only by one character, we have $p(s) > p(s')$. If this holds for all i, then $s = s^*$ by definition. If not, then for some i, it holds that $p(s) \leq p(s')$. In this case, by uniqueness of the maximum, for some i for which this holds, we must have $p(s) < p(s')$. Thus $s \neq s^*$. □