An Annotated Dataset of Errors in Premodern Greek and Baselines for Detecting Them

Creston Brooks[*†] **Johannes Haubold**[*]
Charlie Cowen-Breen[†] **Jay White** **Desmond DeVaul**
Frederick Riemenschneider **Karthik Narasimhan** **Barbara Graziosi**
Princeton University, Heidelberg University, MIT
cabrooks@princeton.edu, ccbreen@mit.edu

Abstract

As premodern texts are passed down over centuries, errors inevitably accrue. These errors can be challenging to identify, as some have survived undetected for so long precisely because they are so elusive. While prior work has evaluated error detection methods on *artificially-generated* errors, we introduce the first dataset of *real* errors in premodern Greek, enabling the evaluation of error detection methods on errors that genuinely accumulated at some stage in the centuries-long copying process. To create this dataset, we use metrics derived from BERT conditionals to sample 1,000 words more likely to contain errors, which are then annotated and labeled by a domain expert as errors or not. We then propose and evaluate new error detection methods and find that our discriminator-based detector outperforms all other methods, improving the true positive rate for classifying real errors by 5%. We additionally observe that scribal errors are more difficult to detect than print or digitization errors. Our dataset enables the evaluation of error detection methods on real errors in premodern texts for the first time, providing a benchmark for developing more effective error detection algorithms to assist scholars in restoring premodern works.

1 Introduction

Ancient texts have been passed down over hundreds of years. The oldest surviving manuscripts of Sophocles, Plato, and Aristotle date to the ninth and tenth centuries CE, long after the original works were composed in the fifth and fourth centuries BCE. Thus, what is left to us today are copies of copies of copies. Throughout this process of copying, errors have accumulated in three main ways:

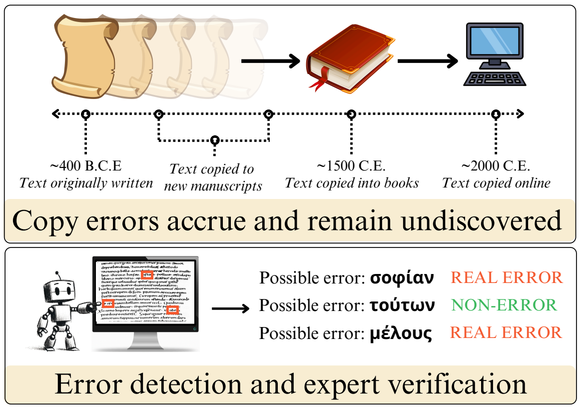

Figure 1: Errors in premodern texts accumulate over centuries of copying. Using machine-learning methods and expert labeling, we create the first dataset of real errors in premodern Greek texts.

Scribal errors: Scribes copying manuscripts over centuries introduce changes—such as adding, omitting, repeating, or simplifying text—that go unnoticed by subsequent scribes and are then copied forward as though they were the original text.

Print errors: Modern scholars occasionally misread manuscripts or introduce typos when creating editions, leading to mistakes in published versions.

Digitization errors: The conversion of printed texts to online versions, whether through manual typing or automated processes, introduces additional errors.

Errors made at all stages, from the earliest copies of an ancient text to what we read online today, threaten the faithful preservation of that text, change its original wording, and impede our understanding of it. The most insidious errors are not simple typos, but alterations that make logical sense, allowing them to persist undetected.

Only one unsupervised method has been proposed for detecting errors in premodern texts using machine-learning techniques: Cowen-Breen et al.

[*]CB and JH contributed equally as first authors.
[†]Correspondence addressed to.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 7203–7217
April 29 - May 4, 2025

(2023) directly leverage distributions learned by a BERT model (Devlin et al., 2019) without task-specific fine-tuning. This method, while successful in identifying a limited number of errors (Graziosi et al., 2023), has only been broadly evaluated on detecting artificial errors generated by random character replacement.

Until now, there has been no available dataset of errors that resulted from the natural process of copying illustrated in Figure 1. In this paper, we introduce the first expert-labeled dataset of real errors (scribal, print, and digitization), enabling the evaluation of error detection methods on real errors rather than artificial ones. We use a form of automated over-sampling to select potential errors, which a domain expert then spends over 100 hours labeling (see Section 5).

Using this dataset, we evaluate Cowen-Breen et al.'s (2023) existing error detection method and propose new unsupervised methods, including one inspired by protein engineering and another using an ELECTRA discriminator (Clark et al., 2020). We also establish a large language model (LLM) baseline with few-shot prompting using GPT-3.5 and GPT-4 (OpenAI et al., 2023). The ELECTRA discriminator improves the true positive rate over the next best method by 5%, while GPT-3.5 and GPT-4 perform only marginally better than random chance, with AUROCs of 0.51 and 0.57, respectively. We additionally observe across methods that scribal errors are more difficult to detect than print or digitization errors.

2 Related Work

Recent years have seen significant progress in training language models (LMs) on premodern languages including Greek (Singh et al., 2021; Yamshchikov et al., 2022; Riemenschneider and Frank, 2023). These works make use of various masked language models (MLMs) for tasks such as dependency parsing, lemmatization, and gap infilling. Assael et al. (2022) focus on filling gaps in inscriptions, and Jones et al. (2022) use support vector machines and decision trees to adjudicate between New Testament manuscript variants. Cullhed (2024) explores the fine-tuning of modern foundation models for filling gaps in ancient papyri, and Duan et al. (2024) take a multimodal approach towards restoring ancient Chinese texts. Notwithstanding these efforts, the field of machine-learning assisted textual restoration remains nascent.

Other work has focused on the supervised detection and correction of errors introduced by Optical Character Recognition (OCR) and Handwritten Text Recognition (HTR), as opposed to scribal errors and print errors (Chiron et al., 2017; Amrhein and Clematide, 2018; Schaefer and Neudecker, 2020; Nguyen et al., 2020; Pavlopoulos et al., 2023). Although errors introduced by OCR and HTR can result in garbled text that is challenging to *correct*, they are generally easy to *detect*, since a simple dictionary check can flag nonsensically distorted words. Additionally, these studies largely rely on extensive datasets of OCR/HTR text with aligned ground truth. Many errors we consider (scribal and print) have survived because they often make logical sense and are thus more difficult to detect.

3 Contributions

Computational textual restoration has previously involved either (i) domain experts using error-detection algorithms to discover a limited number of real errors (Graziosi et al., 2023), or (ii) broadly evaluating error detection algorithms using datasets of artificially generated errors (Spencer et al., 2004; Roos and Heikkilä, 2009; Hoenen, 2015). In contrast, we introduce the first error detection dataset composed of real errors. We then use this dataset to evaluate the existing error detection method as well as additional methods which we propose. We summarize our contributions as follows:

1. We create a dataset of textual errors flagged by machine-learning methods and annotated by a domain expert.[1]

2. We propose two new error detection methods: one inspired by protein engineering and another using an ELECTRA discriminator.

3. We pre-train a suite of models with varying architectures to evaluate the existing and proposed error detection methods using our expert-labeled dataset.

With real textual problems, labeled and annotated by a domain expert, error detection methods can be effectively evaluated at scale for the first time. In turn, improved error detection capabilities lead to better identification of errors for future domain

[1]We make this dataset available, along with the error detectors we evaluate: https://github.com/brooksca3/logion_error_dataset.

expert review, propelling the discovery cycle. Here, we enable the cycle of accelerated error discovery seen in Figure 2.

4 Error Detection

Given a word w_i and its surrounding context $\mathbf{w} = (w_1, \ldots, w_k)$, the task of error detection is to determine whether the given word is an error. More precisely, an *error detector* is a function T such that $T(\mathbf{w}, i)$ produces an error score for the word w_i in the given context $\mathbf{w}$.

Error detectors are useful because the scores they produce can yield a list of words deemed most likely to be errors. For example, a word w_i may be shortlisted as a potential error if $T(w, i) > 0.99$ for a given detector T. Assuming a tolerably successful error detector, words with scores above a certain threshold can be passed on to domain experts for review.

5 Dataset Creation

5.1 Identifying Real Errors

We create a dataset of real errors that accumulated as texts were copied first from handwritten manuscripts, then to printed editions, and eventually to digital versions. To do so, we choose the corpus of the 11th-century Byzantine author Michael Psellos, due to its considerable size (1M words) and availability in digitized form. Our domain expert is a philologist who has worked closely with the texts in question (Haubold, 2023).

The rarity of real errors within the corpus means that drawing random words for expert review would be statistically unlikely to yield any positive labels. Additionally, the labeling process is time-consuming, as the domain expert must consult various printed editions, manuscript versions, and, in the case of suspected scribal errors, a range of philological resources.

Therefore, we follow the methodology proposed by Cowen-Breen et al. (2023) to over-sample real errors, which we subsequently label:

- Using a premodern Greek BERT model, we assign a Chance-Confidence Ratio (CCR) score (see subsection 6.1) to every word in a subset of the corpus.[2]

[2]In practice, we randomly divided the text into five parts and presented the top 500 CCR-scoring words from each to the domain expert, who labeled 1,000 words from two parts.

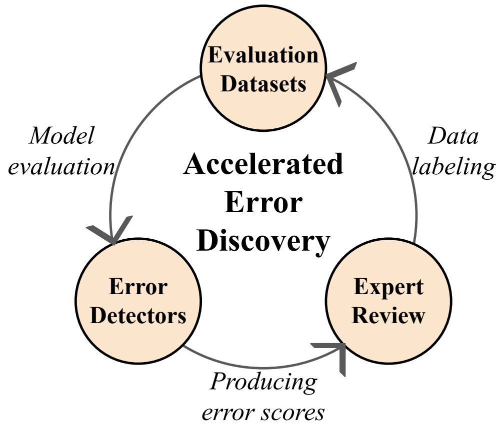

Figure 2: **Proposed pipeline for accelerated error discovery**. Expert labeling creates evaluation datasets (Section 5), leading to better error detectors (Section 6), providing higher-quality samples for the next round of expert review.

- We present a list of the 1,000 words with the highest CCR scores to the domain expert who determines whether each word is an error or not. The expert additionally annotates each example with brief philological comments to justify the given label.

5.2 Labeling Process

The domain expert decides that a word is an error and gives the label $y = 1$ for any of three reasons:

1. *Digitization Error (42 instances):* The expert confirms that the word in question is an error by comparing it with the corresponding text in the printed edition.

2. *Print Edition Error (114 instances):* The expert confirms that the word in question is an error by comparing it with the corresponding text in the available manuscripts.

3. *Scribal Error (61 instances):* The expert assesses the word in question to be a scribal error by philological reasoning.

Figure 3 presents an abridged example from the dataset that contains a scribal error. For the manuscript referenced by the expert in identifying this scribal error, see Appendix A. We note that digitization and print errors can be identified with far greater confidence than scribal errors: for the latter, the assessments in the dataset must be considered preliminary only.

τὸ γὰρ ἐπίρρημα τοῦ ‘ ἐκεῖ ’ τοῦτό μοι ἐμφαίνειν δοκεῖ, ὅτι καὶ τὴν κατὰ μῆκος κινούμενος κίνησιν, ἣν ἀνωτέρω ὁ λόγος ἐδήλωσεν, οὐδὲ **τὸ** πρὸς νότον κατιέναι καὶ αὖθις ἐκεῖθενπρὸς βορρᾶν ἀνιέναι ἐστέρηται, ἀλλὰ κἀκεῖσε πορεύεται κἀνταῦθα κεκίνηται.

Psellos construes στερέω with the genitive (active and passive).
Cf. Ep. 336.6 Papaioannou ὁ μὲν ἤδη καὶ τοῦ βοηθεῖσθαι ἐστέρηται.

τὸ → **τοῦ**

Figure 3: **Abridged dataset example**. The word τὸ is labeled as an error (in this case scribal). The expert notes that Psellos uses the genitive with στερέω, suggesting the text should read τοῦ, and cites a parallel example from Papaioannou's edition of Letter 336.6 where Psellos uses τοῦ with the same verb form. Appendix D provides the complete version of this example, and Appendix A includes an image of the manuscript showing how this scribal error may have been introduced.

Not all words presented to the domain expert could be definitively labeled as real errors or not. In cases of potential scribal errors, where there is no explicit ground truth to verify an error and only reasoning based on textual evidence, the expert identified some words as possible errors, but not with sufficient confidence to label as $y = 1$; a total of 237 such instances were labeled as either "plausible" or "uncertain." We include these examples in the dataset but do not use them for evaluation purposes. Of the 763 words that were definitively labeled by the domain expert, 28% were assigned the positive label $y = 1$ (i.e., an error is present), while 72% were assigned the negative label $y = 0$ (i.e., no error is present).

5.3 Impact of Over-Sampling

The result of our sampling method is that all words presented to the domain expert, regardless of the label they receive, have a high CCR score (see subsection 6.1). To mitigate the distribution shift for non-errors ($y = 0$) caused by over-sampling, we include a set of 237 randomly selected words from the corpus, assume they are non-errors due to the rarity of real errors, and assign them the label $y = 0$.

We note that this approach of over-sampling true positives is similar to that employed in computational methods for drug discovery, in which datasets are usually skewed toward drugs already likely to be effective, due to the similarly high cost of evaluation (Wishart, 2006; Sliwoski et al., 2014; Zagidullin et al., 2019). The case of computational drug discovery is similar in the sense that its goal is discovery—rather than scientific classification—and its bottleneck is in real-world evaluation, rather than computation.

5.4 Summary of the Dataset

In summary, we used Cowen-Breen et al.'s (2023) CCR metric to score a subset of words from the corpus of Michael Psellos, selecting the top 1,000 for expert review. The labeling process took over 100 hours and resulted in 763 words being definitively labeled. The remaining 237 words were labeled "plausible" or "uncertain."

The resulting dataset poses a challenging classification task, as many labels were determined through careful adjudication, consultation of source documents, and analysis of textual parallels. The classification task is made more challenging by the fact that the error detectors we consider have access to none of these materials.

6 Deriving Error Detectors from LMs

In this section, we describe the CCR metric and introduce two new error detection scoring metrics derived from LMs: (1) the Pseudo Log-Likelihood Ratio (PLLR), originally developed for classification tasks in protein engineering, and (2) discriminator scoring, using an ELECTRA discriminator without any additional fine-tuning. We also describe our methodology for prompting instruction-tuned LLMs to judge whether words are errors.[3]

6.1 Chance-Confidence Ratio

CCR is an error detector proposed by Cowen-Breen et al. (2023) for the purpose of error detection and emendation. Given any MLM with learned conditionals $p(\cdot|\cdot)$, CCR scoring is defined

[3]Future work should explore fine-tuning open-source LLMs on the task of error detection or posing it as a reward modeling task.

Model Type	Training Objective	Tokenization	Model Instance(s)
Encoder	Masked Language Modeling	Character	BERT
		Sub-word	BERT (15% & 40% mask ratio)
		Both	BERT
	Replaced Token Detection	Sub-word	ELECTRA
Encoder-decoder	Span Corruption Denoising	Character	T5
		Sub-word	T5

Table 1: Suite of pre-trained models evaluated on error detection.

as follows:

$$T_{CCR}(\mathbf{w}, i) = \frac{\max_{w \in \mathcal{W}^k_{w_i}} p(w|w_{-i})}{p(w_i|w_{-i})}$$

where $\mathcal{W}^k_{w_i}$ denotes the set of words within Levenshtein distance k of w_i, and w_{-i} denotes the contextual sequence $\mathbf{w}$ with the entry at index i masked. Intuitively, CCR is large when the **chance** of a word occurring in its given context, $p(w_i|w_{-i})$, is small relative to the **confidence** of the top model suggestion when restricted to Levenshtein distance k, $\max_{w \in \mathcal{W}^k_{w_i}} p(w|w_{-i})$. For dataset creation and all error detection experiments, we use $k = 1$.

6.2 Pseudo Log-Likelihood Ratio

PLLR is a heuristic used by Brandes et al. (2023) to predict whether a mutated protein sequence is malignant or benign. They find it to be an excellent zero-shot indicator of malignancy. PLLR takes a sequence and a mutated variant of that sequence and computes the ratio of the pseudo log-likelihoods of the sequence and its variant.

We propose applying PLLR to error detection by considering the hypothesis that each sequence of words in our text is itself a mutated variant of some original reference sequence, computing the score as follows:

$$T_{PLLR}(\mathbf{w}, i) = \frac{\max_{w \in \mathcal{W}^k_{w_i}} \hat{p}(w_1, \ldots, w, \ldots, w_n)}{\hat{p}(w_1, \ldots, w_i, \ldots, w_n)}$$

Following Brandes et al. (2023), we compute pseudo-likelihood $\hat{p}(\cdot)$ with a single forward pass of a MLM by multiplying the probabilities of the ground-truth token at each output position, taking advantage of the fact that MLMs output a probability distribution at all positions. While this approach is highly heuristic, computing $\hat{p}(\cdot)$ is efficient insofar as it requires only a single forward pass.

6.3 Discriminator Scoring

We additionally propose using a discriminator model for binary classification on each token to predict whether it is the original or a replacement sampled from a generator. This aligns closely with the phenomenon that we are attempting to model, where a scribe, acting as a generator, occasionally alters words in a text.

6.4 Few-Shot Prompting

Although today's instruction-tuned LLMs are not specifically designed for tasks involving premodern Greek, their training on extensive internet crawls suggests that they could encounter some relevant data (OpenAI et al., 2023; Touvron et al., 2023). We provide sequences of premodern Greek and ask the instruction-tuned LLM to assess whether a specified word is an error, giving examples with expert annotations. We prompt the LLM to return a score from 1 to m indicating how likely a given word is to be an error.[4] More prompting details are made available in our source code.

7 Overview of LM Pre-Trainings

Each error detector we evaluate is unsupervised, using distributions from language model pre-training objectives rather than being trained on a labeled error dataset. Crucially, we pre-train all models from scratch, avoiding existing premodern Greek models to prevent contamination between their training data and our dataset.[5] Our goal is to compare error detection methods, not specific models, which vary in data, compute, and parameters. To ensure a fair comparison, we keep these factors as consistent as possible across the seven models we pre-train.

7.1 Pre-Training Data

We assemble pre-training data from sources made available by prior work, including Singh et al. (2021), Cowen-Breen et al. (2023), and Riemenschneider and Frank (2023). We divide the train-

[4] We try $m = 2, 3, 5, 10$ and find $m = 5$ to be best.

[5] Note, however, that we have no such assurances about the training data used for GPT-3.5 and GPT-4.

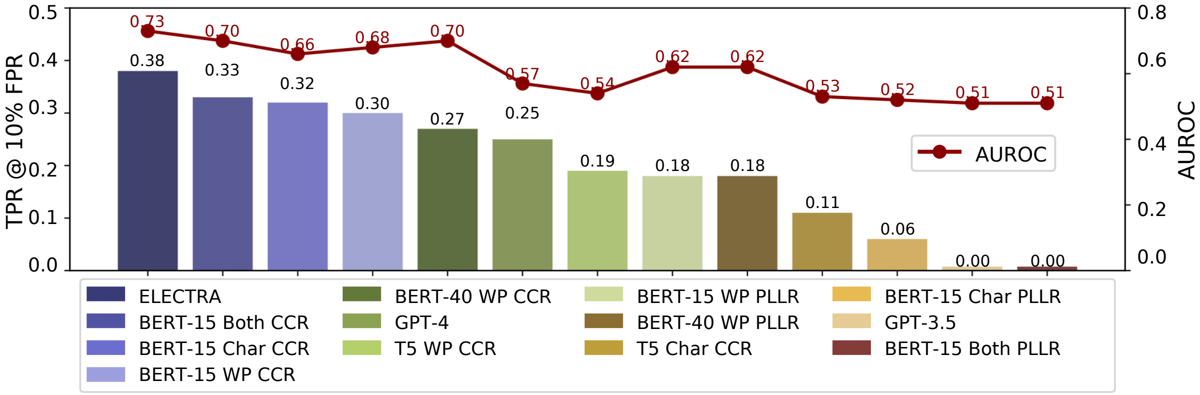

Figure 4: **AUROC and TPR at 10% FPR for each error detector**. "15" and "40" refer to mask ratios, "Char" and "WP" refer to character and sub-word tokenization, and "Both" refers to the combined tokenization method.

Model	Type of error		
	Digitization	Print	Scribal
ELECTRA	0.75	0.71	**0.59**
BERT (Best)	0.65	0.67	**0.57**
T5 (Best)	0.61	0.53	**0.52**
GPT-4 (Best)	0.53	**0.52**	**0.52**

Table 2: AUROC of select detectors when $y = 1$ examples are limited to specific error categories. Scribal errors are universally the most challenging (in bold). "Best" refers to the highest-AUROC detector of each model type.

ing, validation, and testing splits so that no exact 50-character overlap in training occurs in validation or testing. In total, our training set contains about 120M words of premodern Greek. We do not remove redundancies within the training split. We do, however, exclude all texts in the corpus of Michael Psellos, ensuring that the dataset remains fully held-out from all model trainings.

7.2 Tokenization

Since error detection requires sensitivity to character-level changes in text, it is possible that prevalent sub-word tokenization methods such as Byte-Pair Encoding (Sennrich et al., 2015) and WordPiece (Schuster and Nakajima, 2012) are sub-optimal for the task. To investigate this, we pre-train models with both a WordPiece tokenizer with a vocabulary size of 50K and a character-level tokenizer. Following Assael et al. (2022), we additionally train a character-level BERT model with an auxiliary sub-word embedding table, with the aim of incorporating different token granularities for prediction. Although different models utilize different tokenizers, we standardize training examples to contain identical text for each. Specifically, we maximally stack consecutive sentences until the number of character-level tokens exceeds 1,024.

7.3 Pre-Training Configurations

We train several variations of bidirectional encoder or encoder-decoder models as listed in Table 1. These include four BERT models: three models with 15% and 40% mask ratios using a sub-word tokenizer, and a 15% mask ratio using a character-level tokenizer.[6] The fourth is a custom character-level BERT integrated with an auxiliary sub-word embedding table. Additionally, we pre-train two T5 models (Raffel et al., 2020), one each with sub-word and character-level tokenizers. Finally, we pre-train an ELECTRA discriminator in tandem with a generator which we later discard. We train each model on four A100 GPUs for six days or until validation loss converges. For full model training parameters, see Appendix B.

8 Evaluation

An error detector T is evaluated by the quality of its predictions $T(\mathbf{w}, i) = \hat{y}$ on labeled data. For evaluation purposes, we treat T as a binary classifier which declares w_i to be an error when $T(\mathbf{w}, i) \geq t$ for a fixed threshold $t \in \mathbb{R}$. We compare error detectors based on their true positive rate (TPR) at a fixed false positive rate (FPR), as seen in Figure 4. We also consider AUROC, defined to be the area under the graph consisting of pairs of FPRs and TPRs over all $t \in \mathbb{R}$.

[6] Wettig et al. (2023) suggest that a 40% mask ratio is superior to 15% for uniform masking.

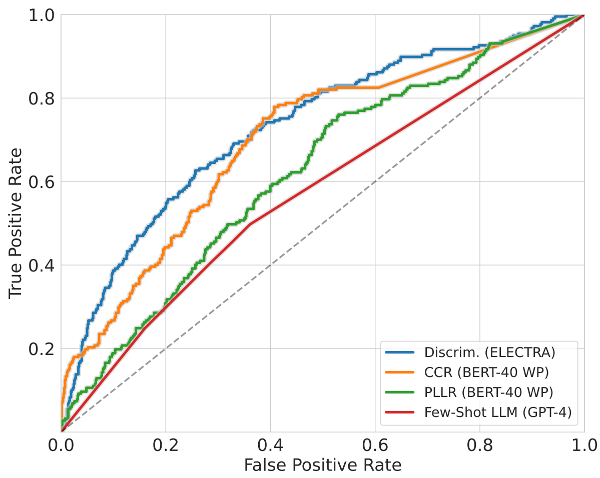

Figure 5: ROC curves of the best performing error detectors of each type. BERT-40 WP denotes the sub-word BERT model trained with 40% mask ratio.

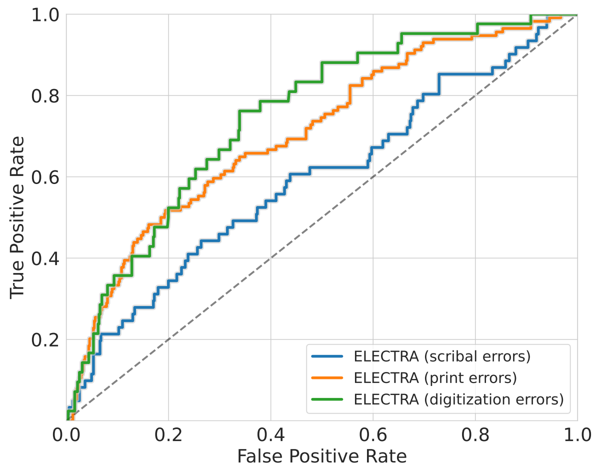

Figure 6: ROC curves of ELECTRA across types of errors.

8.1 Computing Error Scores

We use BERT and T5 models for computing CCR scores, BERT models for PLLR scores, ELECTRA for discriminator scores, and GPT-3.5 and GPT-4 for few-shot prompting scores.[7] We evaluate these error detectors on 763 labeled examples from our dataset and 237 randomly sampled words from the corpus that are presumed to be non-errors.

8.2 Results

The ELECTRA-based error detector achieves the highest scores in both TPR at 10% FPR and AUROC, marking a new state-of-the-art on the classification task introduced with our new dataset. The four BERT-based CCR error detectors are the next best performing in both metrics. In comparison, PLLR-based detectors, T5-based CCR detectors, and few-shot prompted LLMs are noticeably less effective.

Considering the best-performing detector from each category, we observe a clear ranking, as illustrated by the ROC curves in Figure 5: Discriminator Scoring is best, followed by CCR, then PLLR, then Few-Shot LLM Prompting. The results do not provide a strong signal for which tokenization method is best. Extended comparisons across models and methods can be found in Appendix E.

Moreover, we observe across methods that scribal errors are more challenging to detect than print and digitization errors. Table 2 shows that the best-performing detectors of each model type have the lowest AUROC scores for classifying scribal errors. For ELECTRA and BERT-based CCR, which are the most effective error detectors, the drop is especially pronounced. Figure 6 shows this phenomenon for ELECTRA, with ROC curves corresponding to different error types clearly separated. AUROC scores on scribal errors for all models hover relatively close to the random baseline of 0.5. The ease of detecting errors correlates with the recency of the stage in which they were introduced.

[7] We use `gpt-3.5-turbo` and `gpt-4-1106-preview` with a temperature of 1.0.

9 Discussion

The superior performance of ELECTRA as an error detector on our newly created dataset has important implications for machine learning-assisted error discovery. Until now, unsupervised error detection in premodern texts has only employed BERT-based CCR. However, our results indicate that discriminator-based models, like ELECTRA, outperform CCR when evaluated on real copying errors. That said, there are still advantages to using BERT-based models: for a given index, $\arg\max_{w \in \mathcal{W}^k_{w_i}} p(w|w_{-i})$ produces a suggested token within a specified Levenshtein distance, enabling error correction in addition to detection. Future work in error correction could leverage a generator alongside the discriminator to a similar effect.

ELECTRA's success is, in some ways, surprising: the method of over-sampling words with high CCR scores to create this dataset creates a bias for words with a *low* chance metric (see subsection 6.1); on the other hand, the ELECTRA discriminator is primarily trained to detect erroneous tokens with *high* chance values, as they are sampled directly from a generator.[8] Among other considerations, future work could restrict the generator to sample only from $\mathcal{W}_{w_i}^k$ to better simulate the distribution of real errors.

Despite a marked improvement in TPR from GPT-3.5 to GPT-4, both models struggle to classify words effectively, with AUROC scores of 0.51 and 0.57, respectively. Both models produce seemingly well-reasoned yet ultimately misinformed explanations for their classifications. In one telling reply, GPT-3.5 rationalizes a $5/5$ error score as follows:

> "The word 'σαφες' is indeed an error. The correct form should be 'σαφης,' as it should agree with the neuter noun 'το πραγμα' in the nominative singular form. The ending -ες is masculine, while -ης is the proper form for a neuter adjective in this context. This is a clear grammatical error that needs correction."

The word in question is, in fact, correct and GPT-3.5's explanation disregards basic rules of Greek grammar. We cannot blame this particular lapse on the contamination of modern data, as σαφές remains a neuter form in Modern Greek.

We also note the relative under-performance of the proposed PLLR metric. During experiments, we observe that the words maximizing a sequence's pseudo-likelihood are often nonsensical. It appears that adding noise in one position of a sequence can counterintuitively bolster the ground-truth logits occurring in other positions in this pseudo-likelihood setting.

10 Conclusion

We present the first annotated dataset of real errors in premodern Greek texts with a view to improving the evaluation of error detection. We propose new error detection methods and evaluate them on the new dataset using an array of pre-trained models, including different configurations of BERT and T5, ELECTRA, and instruction-tuned LLMs like GPT-4. We find that our proposed discriminator-based detector outperforms other methods and establishes a state-of-the-art for the error detection task introduced by our new dataset. Additionally, we observe across methods that scribal errors are more challenging to detect than print and digitization errors.

Our dataset serves as an important new resource for evaluating the efficacy of machine learning methods in detecting real errors in premodern texts and offers a benchmark for the development of more effective error detection algorithms. Evaluating error detection methods on real errors paves the way for accelerated error discovery and machine-learning assisted restoration of premodern texts. We hope that by creating this dataset and presenting new error detection methods, we can introduce an iterative cycle of improvement, where better datasets lead to better detectors, which in turn lead to even better datasets, and so forth.

Limitations

Models like BERT, ELECTRA, and T5 are traditionally pre-trained and then fine-tuned for specific tasks. In our case, we employ these models directly from pre-training for error detection, which leads to misalignment with their original training objectives. For instance, while the standard MLM task masks about 15% of tokens (roughly 75 tokens in a 500-token example), error detection methods like CCR and PLLR can involve masking just one token at a time, thus resulting in an input that is out of distribution.[9] In this study, we aim to better understand the use of pre-trained language models in the zero-shot setting of error detection scoring.

The circularity of dataset creation and error-detector evaluations is a legitimate concern. Due to the very slow pace (up to many hours per datapoint) of annotation, there is no other known option than to oversample likely errors in some way. Moreover, we note that although the labeled words are oversampled using the BERT CCR metric, the ELECTRA-based detector outperforms the BERT CCR detectors. It is our hope that this dataset will spark the development of better error detectors than those we present here, and that those will yield datasets of their own, which may be cross-referenced against ours to measure the legitimacy of this concern.

Furthermore, our dataset is limited to 1,000 words from a single author. It is restricted in both size and scope due to the significant demands that

[8] ELECTRA learns to sometimes propose the *lectio difficilior*, whereas error detectors guided by chance propose the *lectio facilior*, to employ the terminology of philological scholarship.

[9] A training adjustment to alleviate this effect could be a decaying mask-ratio scheduler.

generating it places on domain experts. We focus on the end task of error detection and deliberately omit examining the relationship between different manuscript copies.

Ethics Statement

Pre-training language models is computationally intensive. As we focus on an underrepresented language, we hope that the models and methods we produce will serve as valuable resources for the scholarly community, with utility extending beyond the scope of this paper.

Acknowledgements

We thank Anirudh Ajith, Julia Balla, Jack Geld, Mirjam Kotwick, Anika Maskara, and Howard Yen for their valuable feedback and advice. We gratefully acknowledge funding from a Magic Grant awarded by the Princeton Humanities Council and computational resources provided by Princeton Language and Intelligence (PLI).

References

Chantal Amrhein and Simon Clematide. 2018. Supervised ocr error detection and correction using statistical and neural machine translation methods. *Journal for Language Technology and Computational Linguistics (JLCL)*, 33(1):49–76.

Yannis Assael, Thea Sommerschield, Brendan Shillingford, Mahyar Bordbar, John Pavlopoulos, Marita Chatzipanagiotou, Ion Androutsopoulos, Jonathan Prag, and Nando de Freitas. 2022. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603(7900):280–283.

Nadav Brandes, Grant Goldman, Charlotte H Wang, Chun Jimmie Ye, and Vasilis Ntranos. 2023. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 55(9):1512–1522.

Guillaume Chiron, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. 2017. Icdar2017 competition on post-ocr text correction. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1423–1428. IEEE.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Charlie Cowen-Breen, Creston Brooks, Barbara Graziosi, and Johannes Haubold. 2023. Logion: Machine-learning based detection and correction of textual errors in Greek philology. In *Proceedings of the Ancient Language Processing Workshop*, pages 170–178, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Eric Cullhed. 2024. Instruct-tuning pretrained causal language models for ancient greek papyrology and epigraphy. *arXiv preprint arXiv:2409.13870*.

Desmond DeVaul. 2023. Desformers. https://huggingface.co/ddevaul/desformers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Chengyu Dong, Liyuan Liu, Hao Cheng, Jingbo Shang, Jianfeng Gao, and Xiaodong Liu. 2023. Fastelectra for efficient pre-training. *arXiv preprint arXiv:2310.07347*.

Siyu Duan, Jun Wang, and Qi Su. 2024. Restoring ancient ideograph: A multimodal multitask neural network approach. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14005–14015, Torino, Italia. ELRA and ICCL.

Barbara Graziosi, Johannes Haubold, Charlie Cowen-Breen, and Creston Brooks. 2023. Machine learning and the future of philology: A case study. *TAPA*, 153(1):253–284.

Johannes Haubold. 2023. Konjekturen zu michael psellos, de philosophia. *Byzantion*, 93:241–261.

Armin Hoenen. 2015. Das artifizielle Manuskriptkorpus TASCFE. In *DHd 2015 - Von Daten zu Erkenntnissen - Book of abstracts*. DHd.

Mason Jones, Francesco Romano, and Abidalrahman Mohd. 2022. Machine learning in textual criticism: An examination of the performance of supervised machine learning algorithms in reconstructing the text of the greek new testament. In *Proceedings of the 2022 7th International Conference on Machine Learning Technologies*, pages 1–5.

Thi Tuyet Hai Nguyen, Adam Jatowt, Nhu-Van Nguyen, Mickael Coustaty, and Antoine Doucet. 2020. Neural machine translation with bert for post-ocr error detection and correction. In *Proceedings of the ACM/IEEE joint conference on digital libraries in 2020*, pages 333–336.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

John Pavlopoulos, Vasiliki Kougia, Paraskevi Platanou, and Holger Essler. 2023. Detecting erroneously recognized handwritten byzantine text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7818–7828, Singapore. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Frederick Riemenschneider and Anette Frank. 2023. Exploring large language models for classical philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

Teemu Roos and Tuomas Heikkilä. 2009. Evaluating methods for computer-assisted stemmatology using artificial benchmark data sets. *Literary and Linguistic Computing*, 24:417–433.

Robin Schaefer and Clemens Neudecker. 2020. A two-step approach for automatic OCR post-correction. In *Proceedings of the 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 52–57, Online. International Committee on Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Pranaydeep Singh, Gorik Rutten, and Els 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, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Gregory Sliwoski, Sandeepkumar Kothiwale, Jens Meiler, and Edward W. Lowe. 2014. Computational methods in drug discovery. *Pharmacological Reviews*, 66(1):334–395.

Matthew Spencer, Elizabeth A Davidson, Adrian C Barbrook, and Christopher J Howe. 2004. Phylogenetics of artificial manuscripts. *Journal of Theoretical Biology*, 227(4):503–511.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

D. S. Wishart. 2006. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, 34(90001):D668–D672.

Ivan Yamshchikov, Alexey Tikhonov, Yorgos Pantis, Charlotte Schubert, and Jürgen Jost. 2022. BERT in plutarch's shadows. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6071–6080, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bulat Zagidullin, Jehad Aldahdooh, Shuyu Zheng, Wenyu Wang, Yinyin Wang, Joseph Saad, Alina Malyutina, Mohieddin Jafari, Ziaurrehman Tanoli, Alberto Pessia, et al. 2019. Drugcomb: an integrative cancer drug combination data portal. *Nucleic acids research*, 47(W1):W43–W51.

Appendix

A Manuscript Section Containing Scribal Error

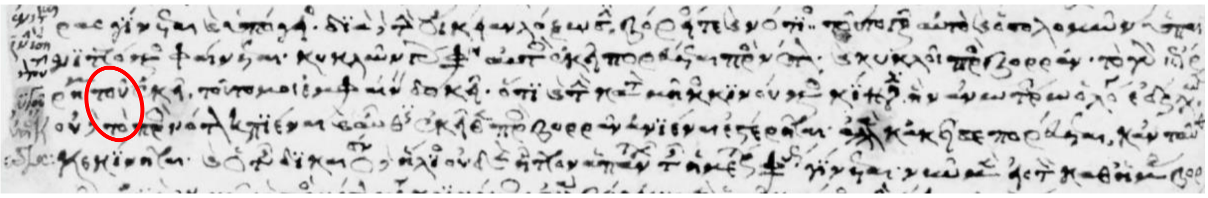

Figure 7: Manuscript section (Cod. Paris. gr. 1182, f. 26v) containing text discussed in Appendix D.

In the figure above, within the red oval, we see τοῦ (on top) and τὸ (below), corresponding to τοῦ in τοῦ ἐκεῖ and τὸ in τὸ πρὸς νότον from the snippet of text in Figure 3 and Appendix D. The BERT-based CCR detector flagged τὸ as an error, which the domain expert determined to be a scribal error based on textual parallels and Psellos's usage of the verb στερέω. Upon further review of the manuscript, the expert noted that this error is connected to another mistake in the line just above: τοῦ in the line above (also within the red oval) should read τὸ. The proximity and similarity of these two words likely caused the confusion.

B Model Training Hyper-Parameters

While the remaining weights are initialized randomly, we initialize the embedding table of the ELECTRA discriminator using a pre-trained BERT model. We train the ELECTRA generator from scratch in tandem with the discriminator. Preliminary testing showed that using a pre-trained generator, even with a temperature schedule (cf. Dong et al. (2023)), hindered the discriminator's learning. For the character-level BERT model with an auxiliary sub-word embedding table, we use DeVaul's (2023) implementation, which is a fork of HuggingFace's `BertForMaskedLM` module.

Hyperparameter	**BERT**	**ELECTRA**	**T5**
Attention Heads	12	12	12
Per Device Batch Size	16	16	16
Hidden Dropout	0.1	0.1	0.1
Hidden Size	768	768	768
Learning Rate (LR)	$5 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$1 \cdot 10^{-4}$
LR Scheduler	linear	linear	cosine
Nb. of Layers	12	12	$2 \cdot 12$
Warmup Steps	0	0	10000

Table 3: Hyper-parameter settings for model training. Our experiments involve two types of models: those utilizing a 50,000-token subword vocabulary and those with character-level input. The remaining hyper-parameters are unchanged from the corresponding HuggingFace model configurations.

C CCR Implementation Details

As detailed in Section 6, our evaluation requires that each experiment produces a list of scores $t \in \mathbb{R}$, corresponding directly to the list of ground truth labels $y \in \{0, 1\}$.
As words can consist of multiple sub-word tokens, in practice we calculate CCR (subsection 6.1) for w_i with tokens $t_1, ..., t_n$ with the following heuristics for chance and confidence:

$$\text{chance} \leftarrow \min_{j=1}^{n} p(t_j | t_{-j})$$

For confidence, we replace masking w_i with 1 to n mask tokens and beam search across each masked sequence to find the top suggestions within Levenshtein distance k of w_i. The confidence is determined as:

$$\text{confidence} \leftarrow \max_{m=1}^{n} \left(\max_{w' \in \mathcal{W}_{w_i}^{k}} p(w' | w_{-m}) \right) ,$$

where w_{-m} indicates the sequence with w_i replaced by m mask tokens. We use a beam size of 10, and if beam search cannot find any w' within distance k of w_i, we return a score of 0. With BERT and T5 models, we compute $p(\cdot|t_{-i})$ by inserting a masked token at position i and then applying softmax to the logits at position i.

Computing $p(\cdot|w_{-i})$ with BERT is straightforward: simply replace the token at position i with a mask token and perform a forward pass to obtain the desired distribution. With T5, this computation is more heuristic: instead of directly replacing a single token, a span corruption approach is used where a token at position i is replaced with the placeholder <extra_id_0>. We then make use of the distribution of potential spans produced by a forward pass.

D Dataset Example

Transmitted Word in Question: τὸ

Expert Label: GOOD FLAG.

Model-Suggested Alternative: τoυ

Further Expert Notes:

GOOD FLAG. GOOD SUGGESTION. Scribal. Codex unicus. Corrupt.

MS P. Psellos construes στερέω with the genitive (active and passive). The error appears to be related to a further corruption earlier in the same sentence, which the error detector did not identify: for transmitted τοῦ ' ἐκεῖ ' read τὸ ' ἐκεῖ ' and note the position of τοῦ < τὸ immediately above τὸ < τοῦ in the relevant manuscript (Cod. Paris. gr. 1182, f. 26v).

1. Michael PSELLUS Epist., Hagiogr., Phil., Polyhist. et Theol. Theologica 2702.012 Opusculum 107 line 56
ρημα τοῦ 'ἐκεῖ' τοῦτό μοι ἐμφαίνειν δοκεῖ, ὅτι καὶ τὴν κατὰ μῆκος κινούμενος κίνησιν, ἣν ἀνωτέρω ὁ λόγος ἐδήλωσεν, οὐδὲ τὸ πρὸς νότον κατιέναι καὶ αὖθις ἐκεῖθεν πρὸς βορρᾶν ἀνιέναι ἐστέρηται, ἀλλὰ κἀκεῖσε πορεύεται

Word Index in Text: 27

Text:

τὸ γὰρ ἐπίρρημα τοῦ ' ἐκεῖ ' τοῦτό μοι ἐμφαίνειν δοκεῖ , ὅτι καὶ τὴν κατὰ μῆκος κινούμενος κίνησιν , ἣν ἀνωτέρω ὁ λόγος ἐδήλωσεν , οὐδὲ τὸ πρὸς νότον κατιέναι καὶ αὖθις ἐκεῖθεν πρὸς βορρᾶν ἀνιέναι ἐστέρηται , ἀλλὰ κἀκεῖσε πορεύεται κἀνταῦθα κεκίνηται . Καὶ ' ὁ τῆς δικαιοσύνης ' δὲ ' ἥλιος ' οὐδὲν ἧττον ἁπανταχοῦ τῆς ἡμετέρας φύσεως γίνεται , νῦν μὲν εἰς τὸν καθ' ἡμᾶς βορρᾶν ἀνιών , νῦν δὲ πρὸς νότον μετακλινόμενος . ἀλλὰ βόρειον μὲν ἡμῖν μέρος πρὸς ὕψος ἠρμένον καὶ πολλαῖς μοίραις τῆς γῆς μετεωριζόμενον ὁ κοσμῶν νοῦς τὴν ψυχήν · νότιον δὲ ἡ μετέχουσα τοῦ νοῦ ψυχή , ὑποβεβηκυῖα μὲν ἐκεῖνον καὶ κάτω ποι τεταγμένη , οὐδ' αὐτὴ δὲ ἀμοιροῦσα τοῦ θείου φωτός . ἢ βορρᾶς μὲν ἡμῖν τὸ σύμπαν νοητόν , ὅσον τε ἐν νῷ καὶ ὅσον ἐν τῇ ψυχῇ , νότος δὲ τὸ συμπεριειλημμένον τῇ ὕλῃ σῶμα , μᾶλλον δὲ τὸ ταύτην συμπεριλαβόν . ἔμελλε γὰρ ἡ καθ' ἡμᾶς ὕλη ὅσον ἐπὶ τῇ οἰκείᾳ φύσει ἀμέτοχος εἶναι καλοῦ , ἀλλ' ὁ πορευόμενος πρὸς νότον καὶ κυκλῶν πρὸς βορρᾶν οὐδὲ ταύτην ἀποστερεῖ τῶν οἰκείων μαρμαρυγῶν , οὐ μόνον οἷς ἐπιτηδείαν ἐργάζεται πρὸς εἴδους καταδοχήν , οὐδ' ὅτι ὁμοῦ τε ὑπέστησε καὶ πρὸς τὴν κοσμοποιίαν ἐχρήσατο , ἀλλ' ὅτι καὶ τὰ πολλὰ τῶν πρακτικῶν ἀρετῶν διὰ ταύτης κατορθοῦσθαι εἴωθεν , εἴπερ αἱ μὲν δέονται σώματος , τὸ δὲ ὕλης οὐκ ἄτερ . Εἶτα πῶς οὐκ ἐσκότωνται οἱ μὴ τὸν τοῦ πατρὸς λόγον κυρίως θεὸν ὀνομάζοντες , δι' οὗ καὶ τὸ θεοῦσθαι τοῖς θεουμένοις ἐστίν , ἀλλὰ τὴν μὲν γέννησιν ἀπαρνούμενοι , ἵνα μὴ πάθος εἰσαγάγωσι , τὴν δὲ κτίσιν αὐτοὶ ἀναπλάττοντες , ἵν' ὁμόδουλον ἡμῖν τὸν δημιουργὸν ποιήσωσιν ; εἰσὶ δὲ οἳ προσίενται μὲν τὴν γέννησιν , ὥσπερ δὴ καὶ τὴν ἀγεννησίαν , οὐσίας δὲ ταύτας ἀντιδιῃρημένας φασίν , ὥσπερ τὸ σῶμα καὶ τὸ ἀσώματον , καὶ θεὸν μὲν ἑκατέραν τῶν οὐσιῶν λέγουσιν , ἀκυρίαν δὲ καὶ ὁμωνυμίαν προσάπτουσι τοῖς μόνοις κυρίοις καὶ ὑπὲρ πᾶσαν λογικὴν μέθοδον . Πρὸς οὓς ὁ μέγας πατὴρ ἁπαντῶν ' ὁ μὲν οὖν ἡμέτερος ' φησί ' λόγος ὥσπερ ἵππου καὶ βοὸς καὶ ἀνθρώπου καὶ ἑκάστου τῶν ὑπὸ τὸ αὐτὸ εἶδος εἷς λόγος ἐστί · καὶ ὃ μὲν ἂν μετέχῃ τοῦ λόγου , τοῦτο καὶ κυρίως λέγεσθαι , ὃ δ' ἂν μὴ μετέχῃ , τοῦτο μὴ λέγεσθαι ἢ μὴ κυρίως λέγεσθαι .

E Additional ROC Curves

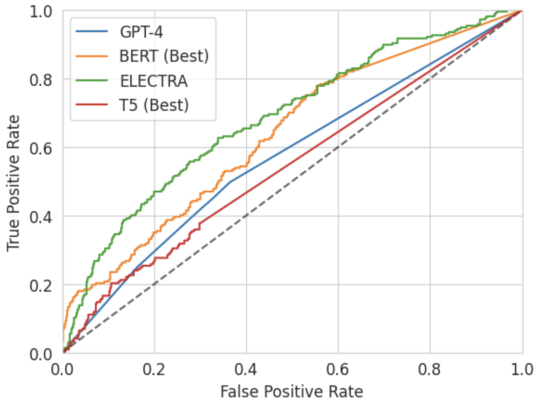

Figure 8: ROC curves of the best performing error detectors of each model type excluding the 237 presumed non-errors sampled from the corpus.

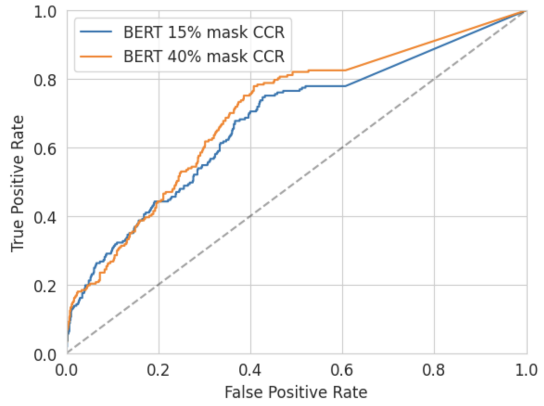

Figure 9: Comparison of ROC curves for BERT models trained with different mask ratios

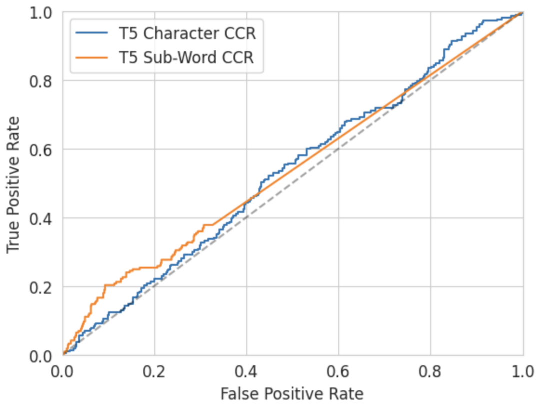

Figure 10: Comparison of ROC curves for T5 models trained with different tokenizers.

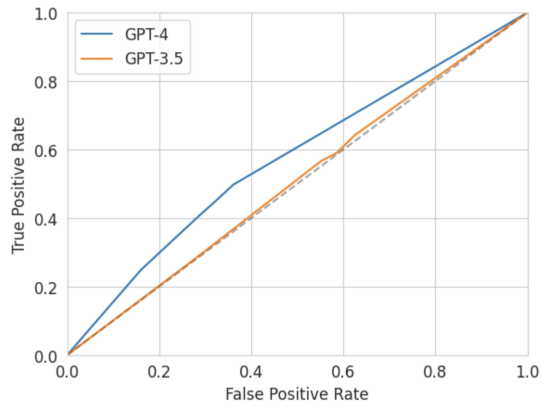

Figure 11: Comparison of ROC curves for GPT-3.5 and GPT-4.