Language Modeling for Epigraphs: a BERT model for EDR's Latin Epigraphs text completion

Olmo Ceriotti
Archeo&Arte3D Lab
DigiLab Sapienza University of Rome
Rome, Italy
ceriotti.2193258@studenti.uniroma1.it

Federico Gerardi
Archeo&Arte3D Lab
DigiLab Sapienza University of Rome
Rome, Italy
gerardi.1982783@studenti.uniroma1.it

Saverio Giulio Malatesta
Archeo&Arte3D Lab
DigiLab Sapienza University of Rome
Rome, Italy
saveriogiulio.malatesta@uniroma1.it

Silvia Orlandi
Department of Science of Antiquities
Sapienza University of Rome
Rome, Italy
silvia.orlandi@uniroma1.it

***Abstract*—The Epigraphic Database Roma (EDR) stands as the most comprehensive and precise collection of Ancient Roman inscriptions, boasting over one hundred thousand entries curated by the International Federation of Epigraphic Databases. Given that the dating of these inscriptions span across centuries, many have suffered from erosion, resulting in missing text. Our objective is to reconstruct these lost segments. To achieve this, we plan to fine-tune LatinBERT, the leading language model for Latin, using the EDR database. This process will yield a specialized language model adept at filling in the gaps within these ancient texts. This advanced model represents a stepping stone in language models trained on inscriptions.**

***Index Terms*—Epigraphy, Text Reconstruction, Deep Learning, Masked Language Modeling, Multi-task Learning, Chronological Classification**

I. Introduction

In Epigraphy, the study of inscriptions written on different materials, such as stone and metal, text reconstruction is a fundamental task. It's not, however, an easy task, since the toll of time on century old inscriptions can lead to a near total loss of information, and current epigraphers' methods are time consuming, slow, and prone to mistakes. The challenge of automatically restoring these lacunae has garnered increasing attention, with recent work such as Locaputo et al. [1] outlining research agendas that propose various deep learning strategies, including the fine-tuning of pretrained language models like LatinBERT [2]. In this paper, we build upon such promising directions by presenting a concrete implementation and empirical evaluation of a method for text reconstruction using the Epigraphic Database Rome (EDR) [3]. We specifically fine-tune LatinBERT on the EDR dataset and expand upon the proposed methodologies by incorporating a multi-task learning objective for chronological classification. Through this approach, we achieved a significant performance improvement on the task of Masked Language Modeling when compared to similar models, actively demonstrating the feasibility of AI-driven reconstruction for Latin epigraphs. This paper is structured as follows: It begins with an introduction to the problem of reconstructing damaged Latin inscriptions from the Epigraphic Database Roma (EDR) [3] and outlines the proposed AI-driven solution. A review of relevant background and related work in computational epigraphy and Latin language modeling follows. The paper then describes the EDR dataset, details the corpus manipulation techniques applied, and defines the primary task of Masked Language Modeling (MLM) alongside an auxiliary chronological classification task. The methodology section elaborates on the model architecture, the custom tokenization process, the training procedure, and the metrics used for evaluation. Subsequently, the results section presents the quantitative performance on both MLM (including an ablation study) and classification tasks, supported by qualitative examples of text completion. The paper concludes by summarizing the key findings and suggesting directions for future work.

II. Background and related work

Prior work in computational epigraphy has explored various avenues for text restoration. For Greek inscriptions, models like PYTHIA and notably Ithaca [4] have demonstrated success using neural networks, with Ithaca also incorporating geographical and chronological attribution. These advancements have set a strong precedent for applying similar deep learning techniques to Latin epigraphy.

The development of **LatinBERT** [2], a transformer-based language model [5] pretrained on a broad corpus of Latin texts, marked a significant step for Latin NLP. While the original work showcased its utility on literary texts, its direct application to the distinct domain of epigraphy requires further investigation. Building on the availability of such models, Locaputo et al. [1] proposed a research agenda for filling lacunae in Latin inscriptions, outlining several deep learning approaches, including the fine-tuning of pretrained models like LatinBERT and exploring architectures similar to Ithaca.

More recently, Brunello et al. [6] presented a case study specifically investigating the fine-tuning of LatinBERT on the

Epigraphik-Datenbank Clauss/Slaby (EDCS) for lacunae infilling in Latin inscriptions. Their experiments, which focused on standard Masked Language Modeling (MLM) objectives, reported suboptimal performance (e.g., a Top-1 token accuracy of 4.02% after fine-tuning on their processed dataset of 211,601 inscriptions), highlighting challenges in adapting existing models to the specific characteristics of epigraphic Latin and underscoring the need for more effective methodologies or refined fine-tuning strategies for this domain. Our work aims to address this by introducing a multi-task learning approach on the EDR dataset to improve MLM performance for Latin inscriptions.

The **Epigraphic Database Roma (EDR)** [3], which forms the basis of our dataset (82,534 cleaned records), is a crucial resource, providing a richly annotated collection of Latin inscriptions essential for training and evaluating computational models for epigraphic tasks.

III. Dataset and Task Description

A. Epigraphic Database Roma

The Epigraphic Database Roma (EDR) [3] represents the most comprehensive and sophisticated epigraphic dataset currently available. EDR is compiled according to a rigorous rulebook that establishes core annotation principles to richly describe each epigraphic entry. Given that the objective of our project is to instill epigraphic Latin knowledge into the LatinBERT model, these annotations had to be parsed into representations of the epigraphs that closely mirror the original texts. Comprising a total of 114,365 inscriptions, after cleaning and parsing it reached a total of 82,534 records.

B. Corpus Manipulation

Many annotations simply indicated uncertainties in textual reconstruction or denoted wholly missing words or letters that could not be restored. These were removed and replaced with an [UNK] token, ensuring that the model recognized the absence of information in the affected sections. Another class of annotation involved marking reconstructed text to signal that certain words were not present in the original inscription. In such cases, the tags were removed and the reconstructed words retained, with the aim of training the model to perform accurate text reconstruction.

Further annotations addressed abbreviations—shorthand conventions employed by the original stonecutters to conserve what was once a valuable material resource. To preserve the full semantic content of these inscriptions, the abbreviated characters were expanded into their complete word forms. This was extremely significant because avoiding to expand the abbreviations would've added an unnecessary layer of complexity to the model tasks. Lastly, some annotations corrected misspellings made by the original sculptors. For the purpose of providing clean input to the model and avoid unnecessary noise, these incorrect forms were replaced with their standardized counterparts. It did not, however, reduce the presence of linguistic variations in the inscriptions since said annotations only address grossly mispelled words.

For example, this is how a standard entry in EDR appears:

```
P(ublius) Mulvius P(ubli) f(ilius)
Cla(udia)<BR>Atiliae Q(uinti)
f(iliae) Secundae,<BR>L(ucio)
Mulvio L(uci) f(ilio)
Primo,<BR>St(atio) Mulvio P(ubli)
f(ilio) Stabilio,<BR>C(aio) Mulvio
P(ubli) f(ilio) fratri,<BR>sibi et
suieis (:suis).
```

After our cleaning process, the text is transformed into:

```
publius mulvius publi filius
claudia atiliae quinti filiae
secundae, lucio mulvio luci filio
primo, statio mulvio publi filio
stabilio, caio mulvio publi filio
fratri, sibi et suis.
```

This version expands abbreviations, corrects grammar, and removes unnecessary tokens such as line break markers.

C. Task Description

The tasks addressed in this study were twofold: **Masked Language Modeling (MLM)** and **epigraph dating (chronological classification)**.

The primary task, Masked Language Modeling, involved fine-tuning the LatinBERT model to enable it to accurately predict masked words within a sentence. This task is particularly relevant for historical and epigraphic texts, where incomplete or partially preserved inscriptions are common. By training the model to infer missing words based on surrounding context, we aim to improve its capacity for contextual understanding and reconstruction of fragmentary Latin texts. This approach is essential not only for enhancing the linguistic capabilities of LatinBERT but also for supporting downstream applications such as automatic restoration of damaged inscriptions.

The secondary task, chronological classification, was implemented as a multi-task learning objective wherein the model was trained to simultaneously predict the approximate date of each epigraph alongside performing MLM. Although considered auxiliary, this task contributes valuable temporal contextualization, which is crucial for epigraphic interpretation. Moreover, incorporating age classification enriches the training signal and provides a form of inductive bias, encouraging the model to learn features relevant to both linguistic content and historical periodization. This multi-task setup was designed to enhance the model's generalization ability and to capture diachronic variations in the Latin language across centuries.

IV. Methodology

A. Model Architecture

The model implemented for the dual tasks of Masked Language Modeling (MLM) and chronological classification, termed `LatinBERTForMLMAndClassification`, leverages the foundational LatinBERT encoder [2]. This pretrained BERT-base model [7] is instantiated using the method

provided by the researchers, ensuring that the final pooling layer is present. This configuration ensures the availability of both token-level hidden states and a sequence-level pooled representation from the underlying BERT model [7].

Atop this shared LatinBERT encoder, two distinct task-specific heads are integrated to perform the operations outlined in Section III.C:

1) **Masked Language Modeling (MLM) Head:** To facilitate the MLM task, a dedicated head is appended. This head comprises a single linear layer that takes the final hidden-state sequence output from the LatinBERT encoder. This output, characterized by dimensions of `[batch_size, sequence_length, hidden_size]`, is projected by the linear layer to the model's vocabulary size. This projection generates the logits necessary for predicting masked tokens at each position within the input sequence.
2) **Sequence Classification Head:** For the chronological classification task, a separate head is employed. This head processes the pooler output from the LatinBERT encoder, which represents an aggregated embedding of the entire input sequence (typically derived from the `[CLS]` token's final hidden state and passed through a dedicated pooling layer). This pooler output, with dimensions `[batch_size, hidden_size]`, is first passed through a dropout layer for regularization. Subsequently, another linear layer maps this regularized representation to the predefined classifications label, 12, 10 corresponding to the centuries from 5 BC to 5 AC, and 2 for epigraphs dated before or after said period, thereby producing the logits for the classification task.

During a forward pass, the `LatinBERTForMLMAndClassification` model processes input sequences through the common LatinBERT encoder. The resulting output sequence is then fed to the MLM head, while the pooler output is directed to the classification head. This architecture enables simultaneous training on both tasks, allowing the model to learn representations beneficial for both token-level linguistic understanding and sequence-level temporal categorization from the epigraphic data.

B. Tokenization

Input epigraphic texts are processed using a custom tokenizer, `LatinTokenizer`, specifically designed for this study. At its core, this tokenizer utilizes a `SubwordTextEncoder` [8] from the tensor2tensor library [9], pre-trained on a Latin corpus. This subword approach is beneficial for handling the rich morphology of Latin and out-of-vocabulary words that might arise from fragmentary inscriptions or orthographic variations.

The `LatinTokenizer` establishes a vocabulary that incorporates both special tokens required by BERT-like architectures and the subword units from the loaded encoder. Specifically, the following special tokens are assigned fixed integer IDs:

- `[PAD]` (ID: 0) for padding sequences to a uniform length.
- `[UNK]` (ID: 1) for representing tokens unknown to the subword vocabulary or deliberately marked as unknown.
- `[CLS]` (ID: 2) prepended to every sequence for classification tasks.
- `[SEP]` (ID: 3) appended to every sequence (and used to separate segments if applicable, though not primarily used here).
- `[MASK]` (ID: 4) for replacing tokens during the Masked Language Modeling task.

The subword units from the `SubwordTextEncoder` are then integrated into the vocabulary with their original IDs offset by +5 to prevent collision with these predefined special token IDs.

The tokenization pipeline proceeds as follows:

1) Input text is first split into space-separated raw words.
2) A crucial pre-processing step, consistent with the corpus manipulation described in Section III.B, involves identifying explicit markers of missing or unrestorable text within the input (e.g., the string `<unk>`. Such markers are directly converted to the model's standard `[UNK]` token.
3) Words that are not corpus-specific unknown markers or predefined special tokens are then passed to the underlying encoder, which segments them into subword units based on the learned subword vocabulary.
4) The resulting subword IDs are adjusted by the +5 offset and mapped to their corresponding subword strings to form the final token sequence.

Finally, to prepare the input for the `LatinBERTForMLMAndClassification` model, the tokenized sequence is formatted according to standard BERT conventions. This includes prepending a `[CLS]` token and appending a `[SEP]` token. Sequences are then either truncated or padded with `[PAD]` tokens to a maximum sequence length, set to 128 in our experiments, and an attention mask is generated accordingly. The resulting vocabulary size for our experiments, including special tokens and subword units, is 32900.

C. Training Procedure

The `LatinBERTForMLMAndClassification` model was fine-tuned using the AdamW optimizer [10]. Key hyperparameters, including a learning rate of `7e-5`, a weight decay of `0.05`, an Adam epsilon of `1e-8`, the choice of a linear learning rate scheduler with a warm-up phase over the first `12%` training steps, a total of `9.0` epochs, and a batch size of `16` per device, were determined through empirical methods to optimize performance on our specific dataset and tasks. The maximum sequence length for input texts was set to `128` tokens; this value was chosen because the vast majority of the epigraphic texts in our corpus are shorter than this limit, allowing for efficient processing while capturing most of the relevant information. For the Masked Language

Modeling (MLM) task, tokens were masked with a probability of `15%`, following common BERT pre-training practices [7]. Gradient clipping was applied with a maximum gradient norm of `1.0` to stabilize training, a value also refined empirically. The overall loss function for our multi-task learning setup [11] is a weighted sum of the individual losses from the two tasks:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \lambda \cdot \mathcal{L}_{\text{classification}} \tag{1}$$

where $\mathcal{L}_{\text{MLM}}$ is the CrossEntropyLoss for the MLM task, and $\mathcal{L}_{\text{classification}}$ is the CrossEntropyLoss for the chronological classification task. The hyperparameter λ, which balances the contribution of the classification task, was set to `0.5` in our experiments, a value also found through empirical tuning. All training and evaluation procedures were executed on a machine with 64 GBs of RAM and an NVIDIA RTX 4070 with 16 GBs of VRAM with a total run time of 1 hour and 33 minutes, utilizing the PyTorch framework [12] and the Hugging Face Transformers library [13].

D. Evaluation Metrics

To assess the performance of the `LatinBERTForMLMAndClassification` model on the two designated tasks, a comprehensive set of evaluation metrics was employed. These metrics were calculated on a held-out testing dataset, comprising `10%` of the full dataset, separated before training.

For the Masked Language Modeling (MLM) task, performance was primarily evaluated using the following metrics:

- **Perplexity (PPL):** Calculated as the exponential of the average MLM cross-entropy loss on the evaluation set. Lower perplexity indicates better predictive performance.
- **Top-1 Accuracy (Acc@1):** The percentage of masked tokens for which the model's highest probability prediction was the correct token.
- **Top-5 Accuracy (Acc@5):** The percentage of masked tokens for which the correct token was among the model's top 5 highest probability predictions.
- **Top-10 Accuracy (Acc@10):** The percentage of masked tokens for which the correct token was among the model's top 10 highest probability predictions.

These MLM metrics provide insight into the model's ability to understand contextual information and predict plausible linguistic units in place of masked tokens.

For the Chronological Classification task, which involves assigning each epigraph to one of the 12 predefined temporal periods, the following standard classification metrics were utilized:

- **Overall Accuracy:** The proportion of epigraphs in the evaluation set that were correctly assigned to their true temporal period.
- **Weighted Precision:** The precision calculated for each class and then averaged, weighted by the number of true instances for each class. This metric accounts for class imbalance.
- **Weighted Recall:** The recall calculated for each class and then averaged, weighted by the number of true instances for each class, also accounting for class imbalance.
- **Weighted F1-score:** The F1-score (harmonic mean of precision and recall) calculated for each class and then averaged, weighted by the number of true instances for each class. This provides a single balanced measure of performance, especially useful in the presence of class imbalance.

The use of weighted averages for precision, recall, and F1-score was chosen to provide a more robust evaluation given the potential for imbalanced distribution of epigraphs across the different temporal periods in the dataset. Model checkpoints were saved based on the best combined evaluation loss, and the final reported metrics correspond to the performance of this best-performing checkpoint on the evaluation set.

V. Results

This section presents the performance of our `LatinBERTForMLMAndClassification` model on the tasks of Masked Language Modeling (MLM) and chronological classification, evaluated on the held-out test set.

A. Masked Language Modeling Performance

The efficacy of our model in predicting masked tokens within Latin epigraphic texts was assessed using perplexity and top-k accuracy metrics. Table I summarizes the performance of this model and provides an ablation study comparing it with two other configurations to contextualize its effectiveness. Our primary model achieved a perplexity of 8.3 on the evaluation set, indicating its proficiency in modeling the linguistic structure of the epigraphic Latin. Furthermore, the top-1 accuracy reached 63.7%, demonstrating that the model's most confident prediction was frequently correct. The accuracy increased to 77.5% and 81.7% when considering the top-5 and top-10 predictions, respectively, highlighting the model's ability to often include the correct token within a small set of plausible candidates. Furthermore, the performance of our full model (Acc@1 63.7% on our EDR-derived dataset of 82,534 inscriptions) represents a significant improvement over previously reported results for fine-tuning LatinBERT on Latin epigraphic data for MLM, such as the 4.02% Top-1 token accuracy achieved by Brunello et al. [6] on their larger EDCS-derived dataset (211,601 inscriptions). This substantial difference highlights the effectiveness of our specific corpus manipulation techniques, multi-task learning strategy, and fine-tuning procedure in markedly boosting the model's ability to predict lacunae in Latin inscriptions, even when working with a comparatively smaller training set.

B. Qualitative Examples

To provide a qualitative insight into the model's predictions, we present two examples from our evaluation set. These examples illustrate different scenarios of the model's performance on the MLM task.

TABLE I
MLM PERFORMANCE ABLATION STUDY.

Model	PPL	Acc@1 (%)	Acc@5 (%)	Acc@10 (%)
Full model	**8.3**	**63.7**	**77.5**	**81.7**
No classification	15.7	55.3	70.0	74.9
Base model	4619.05	9.8	19.9	24.4

a) Example 1: All Masks Correctly Predicted as Top Choice: The following example demonstrates a case where the model successfully predicted the correct token as its first choice for all masked positions detailed in the log.

Original Text:

```
[CLS] dis_ manibus_ ._ tiberius_ iulius_
amati us_ fecit_ sibi_ et_ coniugi _ et_
suis_ liberti s_ libert abus que_
utriusque_ sexus_ poster isq ue_ eorum_ ._
[SEP]
```

Masked Input Text:

```
[CLS] dis_ manibus_ [MASK] tiberius_
iulius_ amati us_ [MASK] sibi_ et_ coniugi _
[MASK] suis_ liberti s_ libert abus que_
utriusque_ sexus_ poster isq ue_ eorum_ ._
[SEP]
```

Note: The prediction logs below detail the model's output for four specific masked positions that were part of this test instance.

MLM Predictions:

- **Position 3** (Input: `[MASK]`, True Original Target: `'.'`):
  1. `.` (ID: 8, Logit: 15.7743)
  2. `sacrum_` (ID: 6272, Logit: 9.4583)
  3. `,` (ID: 7, Logit: 9.4232)
  4. `fecit` (ID: 433, Logit: 8.6104)
  5. `fecerunt_` (ID: 2061, Logit: 7.0163)
  ... *(and 5 more predictions)*
- **Position 8** (Input: `[MASK]`, True Original Target: `'fecit_'`):
  1. `fecit_` (ID: 433, Logit: 15.4278)
  2. `fecerunt_` (ID: 2061, Logit: 10.1594)
  3. `posuit_` (ID: 1291, Logit: 8.7752)
  4. `,` (ID: 7, Logit: 7.6805)
  5. `vivus` (ID: 15929, Logit: 7.2654)
  ... *(and 5 more predictions)*
- **Position 12** (Input: `''`, True Original Target: `''`):
  1. `''` (ID: 9, Logit: 17.4233)
  2. `,` (ID: 7, Logit: 7.2450)
  3. `i_` (ID: 19, Logit: 7.1876)
  4. `s_` (ID: 14, Logit: 7.1839)
  5. `ni_` (ID: 205, Logit: 7.1335)
  ... *(and 5 more predictions)*
- **Position 13** (Input: `[MASK]`, True Original Target: `'et_'`):
  1. `et_` (ID: 10, Logit: 12.6462)
  2. `suae_` (ID: 459, Logit: 8.9087)
  3. `,` (ID: 7, Logit: 8.6413)
  4. `suis` (ID: 291, Logit: 8.2428)
  5. `filiis_` (ID: 1535, Logit: 7.3847)
  ... *(and 5 more predictions)*

b) Example 2: Correct Token Within Top Predictions: This example illustrates a scenario where for one of the masked tokens, the correct prediction was not the top choice but was present within the top-10 predictions, specifically at rank 8.

Original Text:

```
[CLS] bra etia e_ mani_ filiae_ quarta e_
[UNK] [SEP]
```

Masked Input Text:

```
[CLS] bra etia e_ [MASK] [MASK] quarta e_
[UNK] [SEP]
```

MLM Predictions:

- **Position 4** (Input: `[MASK]`, True Original Target: `'mani_'`):
  1. `cai_` (ID: 11149, Logit: 11.4374)
  2. `quinti_` (ID: 20987, Logit: 10.3114)
  3. `titi_` (ID: 30796, Logit: 10.0980)
  4. `marci_` (ID: 13145, Logit: 10.0979)
  5. `libert` (ID: 29484, Logit: 9.7493)
  6. `sexti_` (ID: 27842, Logit: 9.2751)
  7. `numeri_` (ID: 5488, Logit: 7.1043)
  8. **`mani_`** (ID: 18896, Logit: 7.0033)
  9. `luci` (ID: 5218, Logit: 6.9648)
  10. `auli` (ID: 18994, Logit: 6.0596)
- **Position 5** (Input: `[MASK]`, True Original Target: `'filiae_'`):
  1. `filiae_` (ID: 4748, Logit: 12.8525)
  2. `ae_` (ID: 89, Logit: 10.3673)
  3. `''` (ID: 9, Logit: 8.5043)
  4. `filia` (ID: 1750, Logit: 8.0327)
  5. `e_` (ID: 23, Logit: 7.2115)
  ... *(and 5 more predictions)*

These examples showcase the model's ability to generate contextually relevant completions, providing valuable insights into its potential for assisting epigraphic text reconstruction.

c) Example 3: Uncertainty in Numerical Prediction: Another interesting scenario emerges when the model predicts numerical values, such as the number of days someone lived, often recorded in funerary inscriptions. The following example concerns the masked tokens in the phrase '... dies_ [MASK] [MASK] ._' from a longer inscription. The true original target for the first of these '[MASK]'s (labeled as Position 54 in the provided data) is 'numero_', and for the second '[MASK]' (Position 55) is 'xxii_'. We focus on the predictions for Position 55 to illustrate a specific challenge.

Original Text (with relevant masks filled for context):

```
[CLS] quinto_ egn atio_ galli eno_ [UNK] [UNK] tar roni
o_ pison [MASK] [MASK] clarissimo _ [MASK] ,_ nobili_
genere_ nato_ ,_ [MASK] [MASK] latinis_ litteris_ erudito _ ,_
quintus_ egn atiu [MASK] [MASK] enus_ perpetu us_ ,_ vir_
[MASK] [MASK] [MASK] pater_ fecit_ annos_ vixit_ numero_
xviii_ ,_ menses_ numero_ x_ ,_ dies_ numero_ xxii_ ._ [SEP]
```

Masked Input Text (sequence fed to model):

```
[CLS] quinto_ egn atio_ galli eno_ [UNK] [UNK] tar roni
o_ pison [MASK] [MASK] clarissimo _ [MASK] ,_ nobili_
genere_ nato_ ,_ [MASK] [MASK] latinis_ litteris_ erudito _ ,_
quintus_ egn atiu [MASK] [MASK] enus_ perpetu us_ ,_ vir_
[MASK] [MASK] [MASK] pater_ fecit_ annos_ vixit_ numero_
xviii_ ,_ menses_ numero_ x_ ,_ dies_ [MASK] [MASK] ._ [SEP]
```

MLM Predictions (for Position 55, the second `[MASK]` after `dies_`):

- **Position 55** (Input: `[MASK]`, True Original Target: `'xxii_'`):
  1. `'iii_'` (ID: 201, Logit: 12.1328)
  2. `'x_'` (ID: 134, Logit: 10.4840)
  3. `'iiii_'` (ID: 2181, Logit: 9.8927)
  4. `'v_'` (ID: 84, Logit: 9.5858)
  5. `'ii_'` (ID: 101, Logit: 8.9988)

TABLE II
CLASSIFICATION PERFORMANCE.

Metric	Value
Accuracy (%)	68.8
W. Precision	67.7
W. Recall	68.7
W. F1-score	67.9

6. `'i_'` (ID: 19, Logit: 8.5722)
7. `'vi_'` (ID: 206, Logit: 8.5265)
8. `'ix_'` (ID: 501, Logit: 8.4852)
9. `'xv_'` (ID: 752, Logit: 8.4433)
10. `'_'` (ID: 9, Logit: 8.4370)

As observed from the predictions for Position 55, the model suggests various plausible numerical tokens (e.g., 'iii_', 'x_', 'v_') or parts of them. This indicates it understands a number is expected in this context (following 'dies_' and, in the complete original, 'numero_'). However, it does not confidently pinpoint the correct target 'xxii_' as its top choice. This behavior underscores a key challenge: while the model learns that a numerical value is appropriate, predicting the exact number is a difficult task, reflecting the ambiguities and complexities that even human epigraphers face when restoring fragmented or variably abbreviated numerals in ancient inscriptions.

C. Chronological Classification Performance

The model's capability to determine the temporal period of epigraphs was evaluated using standard classification metrics. Table II presents the overall accuracy and weighted precision, recall, and F1-score for our `LatinBERTForMLMAndClassification` model. The model achieved an overall accuracy of 68.8% in classifying epigraphs into one of the 12 defined chronological periods. The weighted F1-score, which accounts for potential class imbalances, was 67.9%, with a weighted precision of 67.7% and a weighted recall of 68.7%. It is important to note that Table II only presents the results for our primary `LatinBERTForMLMAndClassification` model. As the chronological classification task was considered an auxiliary objective, primarily intended to potentially enhance the main MLM task rather than being a central focus of this study, a comparative analysis with other models or ablations for this specific task was not performed.

VI. CONCLUSION

This paper introduced a specialized BERT-based model, `LatinBERTForMLMAndClassification`, fine-tuned on the extensive Epigraphic Database Roma (EDR) for the task of Latin epigraphic text completion. Our results demonstrate the efficacy of this approach, with the model achieving a perplexity of 8.3 and a top-1 accuracy of 63.7% on the Masked Language Modeling (MLM) task, as detailed in Table I. Notably, the inclusion of an auxiliary chronological classification task, which itself achieved an accuracy of 68.8% (Table II), appeared to enhance the primary MLM performance, suggesting a beneficial synergistic effect from the multi-task learning setup. This significantly surpasses the performance of a non-fine-tuned LatinBERT and a model fine-tuned solely on MLM, underscoring the value of domain-specific adaptation and multi-task learning for understanding and reconstructing fragmentary ancient texts. Future work could explore several promising directions. Firstly, incorporating more granular epigraphic metadata, such as precise provenance or inscription type, could further refine the model's contextual understanding and dating capabilities. Secondly, extending the model to handle the generation of longer missing sequences, rather than just single masked tokens, would be a significant step towards practical, full-scale text reconstruction. Additionally, training the model on an abbreviation disambiguation task would further help the epigraphers in their reconstruction tasks, automating a key step in the process. Finally, creating an interactive tool that allows scholars to leverage these predictions and provide feedback could bridge the gap between AI-driven reconstruction and expert epigraphic practice, ultimately advancing the study and preservation of these invaluable historical documents.

REFERENCES

[1] A. Locaputo, B. Portelli, E. Colombi, G. Serra, and others, "Filling the Lacunae in ancient Latin inscriptions.," in IRCDL, 2023, pp. 68–76.

[2] D. Bamman and P. J. Burns, "Latin BERT: a contextual language model for classical philology," 21st September 2020, arXiv: arXiv:2009.10053. doi: 10.48550/arXiv.2009.10053.

[3] [EDR-EDR Official Website]. http://www.edr-edr.it/default/index.php (accessed May 16, 2025)

[4] Y. Assael et al., "Restoring and attributing ancient texts using deep neural networks," Nature, vol. 603, no. 7900, pp. 280–283, Mar. 2022, doi: 10.1038/s41586-022-04448-z.

[5] A. Vaswani et al., "Attention is all you need," Aug. 02, 2023, arXiv: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[6] A. Brunello et al., "Usage of language model for the filling of lacunae in ancient Latin inscriptions: A case study," in Proceedings of the 2nd Workshop on Artificial Intelligence for Cultural Heritage (IAI4CH 2023) co-located with the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023), Roma, Italy, 2023, pp. 113–125.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 24, 2019, arXiv: arXiv:1810.04805. doi: 10.48550/arXiv.1810.04805.

[8] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," Jun. 10, 2016, arXiv: arXiv:1508.07909. doi: 10.48550/arXiv.1508.07909.

[9] A. Vaswani et al., "Tensor2Tensor for neural machine translation," Mar. 16, 2018, arXiv: arXiv:1803.07416. doi: 10.48550/arXiv.1803.07416.

[10] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," Jan. 04, 2019, arXiv: arXiv:1711.05101. doi: 10.48550/arXiv.1711.05101.

[11] R. Caruana, "Multitask learning," Machine Learning, vol. 28, no. 1, pp. 41–75, Jul. 1997, doi: 10.1023/A:1007379606734.

[12] A. Paszke et al., "PyTorch: an imperative style, high-performance deep learning library." Accessed: May 13, 2025. [Online]. Available: https://arxiv.org/abs/1912.01703

[13] T. Wolf et al., "HuggingFace's Transformers: state-of-the-art natural language processing," Jul. 14, 2020, arXiv: arXiv:1910.03771. doi: 10.48550/arXiv.1910.03771.