Lamba: Mamba State Space Model Adapted to Latin Semantic Retrieval*

Elia Scapini[1,*,†], Federico Iezzi[2,*,†]

[1]Dipartimento di Educazione e Scienze Umane (DESU) - Viale Timavo 93, 42121 Reggio nell'Emilia RE, Italy.
[2]Dipartimento di Educazione e Scienze Umane (DESU) - Viale Timavo 93, 42121 Reggio nell'Emilia RE, Italy.

Abstract

This paper presents *Lamba*, a novel adaptation of the Mamba State Space Model for semantic retrieval in Latin texts. While Transformer-based models dominate natural language processing, their quadratic complexity limits their applicability to long and morphologically rich texts typical of ancient languages. We investigate whether Mamba's linear-time architecture can serve as an efficient alternative by implementing a two-phase adaptation pipeline. Phase 1 focused on tokenizer extension and selective retraining of embeddings and output layers to align the model with Latin morphology. Phase 2 applied parameter-efficient fine-tuning (LoRA) with an unsupervised SimCSE objective to optimize sentence embeddings for retrieval tasks. Evaluation on the Gretino benchmark demonstrates significant improvements over the baseline, confirming the viability of this approach for low-resource domains. The results highlight Mamba's potential for advancing semantic tools in Digital Humanities and open perspectives for extending the methodology to other historical languages. We release Lamba with the training corpus together with the development code.

Keywords

Mamba, State Space Models (SSM), Semantic retrieval, Latin NLP, Ancient languages, SimCSE, Gretino benchmark

1. Introduction and Related Works

The attention mechanism has profoundly transformed the field of Natural Language Processing (NLP)[1], serving as the foundational principle of the main Transformer architectures[2, 3]. Following the release of landmark models such as BERT and RoBERTa, the first adaptation specifically targeting Latin was proposed approximately two years later by Bamman and Burns in 2020. This contribution initiated a gradual proliferation of Transformer-based adaptations across various NLP tasks for Latin, systematically documented and evaluated within the EvaLatin evaluation campaigns[5, 6, 7]. However, to date, no models have been explicitly optimized for the task of semantic retrieval. The first and currently only significant contribution in this area, albeit primarily focused on Greek, is the work by Riemenschneider and Frank, who introduce SPhilBERTa, a multilingual SBERT model (English, Latin, and Greek) obtained through distillation from their earlier PhilBERTa model[9]. Our evaluation experiments on the semantic retrieval task, conducted using the Gretino benchmark developed by the authors[1], confirm the superior performance of this model compared to other existing approaches. Despite the dominance of Transformer models in NLP due to their strong empirical performance, the quadratic computational cost of the self-attention mechanism poses substantial limitations when processing long sequences. In this context, State Space Models (SSMs)[11], architectures inspired by control theory, have emerged as a promising alternative, aiming to overcome these constraints by enabling linear-time sequence processing. This property allows for the efficient handling of extremely long contexts with significantly reduced computational and memory requirements. Mamba, the most recent high-performance SSM, introduces a selective state mechanism that enables the model to dynamically

IRCDL 2026: 22nd Conference on Information and Research Science Connecting to Digital and Library Science, February 19-20, 2026, Modena, Italy

*Corresponding author.
†These authors contributed equally.
elia.scapini@unimore.it (E. Scapini); federico.iezzi@unimore.it (F. Iezzi)
0009-0001-6698-0457 (E. Scapini); 0009-0005-5488-1426 (F. Iezzi)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[1]The details of which are not discussed here, as it is currently under publication[10].

CEUR-WS.org/Vol-4240/short7.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-08-20

focus on relevant information. By maintaining a compressed state that is updated token by token, the model can selectively forget irrelevant inputs or preserve long-range dependencies, thereby combining the efficiency of recurrent models with the expressive power typically associated with Transformer architectures. In light of these considerations, the following research question arises: can the Mamba architecture, a recent and highly efficient State Space Model (SSM), serve as a viable alternative to traditional Transformer models for the complex task of semantic retrieval in Latin?

2. Gretino Silver Benchmark Dataset

Model performance was evaluated using the synthetic Latin portion of the Gretino Dataset[10]. We designed Gretino to quantitatively evaluate sentence-level semantic retrieval in Latin, Ancient Greek and cross-linguistic. This study evaluates model performances on a benchmark consisting of 100 query sentences and 5 correct target phrases associated with each query, resulting in a total of 500 target sentences. The evaluation procedure, for each of the 100 queries, embeds all 500 target sentences and rank them by semantic similarity. The primary success metric is the model's ability to rank the 5 correct targets for a given query as highly as possible. Performance is measured by the average number of correct targets successfully retrieved in the top 10 results (`Avg_hits@10`).

3. Preliminary Configuration

3.1. PEFT

Fine-tuning large models is computationally expensive. Parameter-Efficient Fine-Tuning[12] (PEFT) methods adapt models by training only a small fraction of their parameters. Low-Rank Adaptation[13] (LoRA) is a prominent PEFT technique that freezes the pre-trained model's weights and injects small, trainable low-rank matrices into its layers. This allows for significant adaptation with minimal computational overhead, making it an ideal strategy for efficiently specializing large models for new tasks.

3.2. Sentence Embedding

A language model like Mamba operates at the token level, producing a contextualized vector—or hidden state—for every token in an input sentence. For semantic retrieval, however, a single vector representing the entire sentence is required. To achieve this, we employed mean pooling as follows: a sentence is passed through the model, which outputs the final layer's hidden states, one vector per token. To handle batches of sentences with varying lengths, shorter sentences are padded with special tokens; these padding tokens are ignored using the model's attention mask. The hidden state vectors corresponding to the real tokens are summed and divided by the number of valid tokens, producing a fixed-size sentence embedding. This vector captures the overall semantic meaning of the sentence and serves as the representation for similarity calculations in the benchmark dataset.

3.3. Baseline

As a preliminary experiment, we tested whether the language barrier could be mitigated by translating the Latin benchmark into English and exploiting the model's stronger performance in that language. On the translated dataset, the system achieved an average score of 2.97/5 for top-10 retrieval, compared to 2.09/5 when applied directly to the Latin benchmark.

4. From Mamba to Lamba

The pipeline that led from Mamba to Lamba was performed in two distinct phases to systematically adapt the model to the target domain as shown in Figure 1. The Mamba model selected for the adaptation is `state-spaces/mamba-130m`

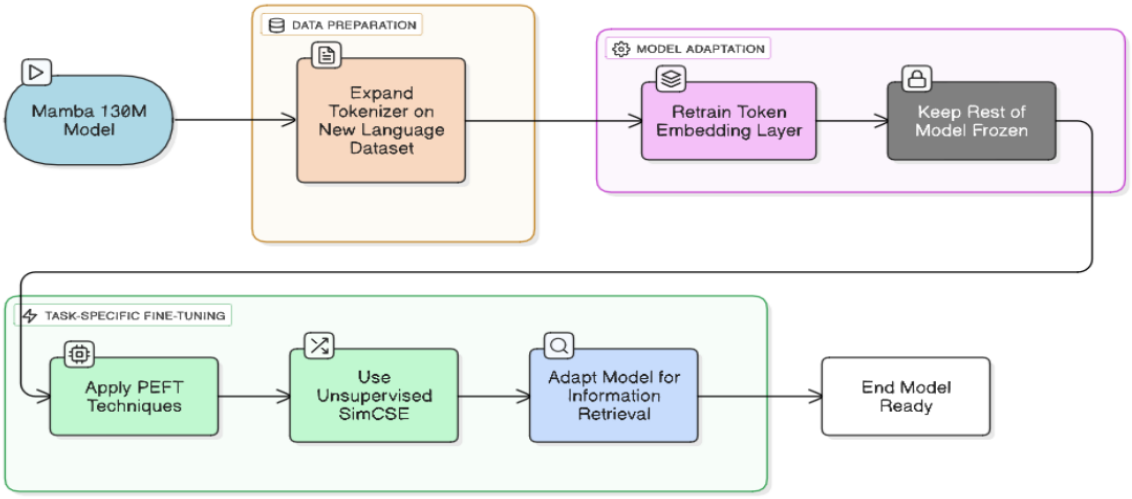

Figure 1: Phased Approach Schema.

4.1. Phase 1: Model Adaptation

The first adaptation phase focused on equipping the base Mamba model with the Latin vocabulary. This stage was not yet task-specific fine-tuning, but rather a preparatory step designed to align the model's internal representation space with the linguistic properties of Latin. It consisted of two main procedures: tokenizer adaptation and model adaptation.

4.1.1. Tokenizer Adaptation

The original Mamba tokenizer was trained on modern, English-centric corpora and lacked coverage for Latin morphology. To address this gap, we extended the tokenizer by training a temporary Byte-Pair Encoding (BPE) tokenizer on a large Latin corpus[2]. The new vocabulary was merged with the original tokenizer, adding previously unseen Latin lemmas while preserving compatibility with the pre-trained model. This ensured that the model could properly recognize and tokenize the morphological richness of the target language.

4.1.2. Embedding and Output Layer Retraining

Once the tokenizer was extended, we adapted the model's input and output interface to the new vocabulary. Specifically, we retrained only the token embedding matrix and the output layer (`lm_head`), while freezing all other parameters of the Mamba model. This choice was motivated by the consideration that freezing the core model preserved its strong sequence modeling capabilities. Furthermore, the embeddings and the output layer directly map token IDs to semantic vectors; without retraining them, the newly introduced Latin tokens would remain meaningless within the vector space of the model. We explored multiple learning rate (LR) configurations before converging on a setup where the embedding and output layers were optimized separately with slightly different learning rates. A cosine learning rate schedule with warmup was adopted to stabilize training, and evaluation against the Gretino dataset was integrated into the training loop to continuously track semantic retrieval performance.

The training was configured with a learning rate of 1.0×10^{-3} for the embedding layer and 7.5×10^{-4} for the output layer. We used a batch size of 32 and trained the model for 10 epochs, with a maximum sequence length of 1024 tokens. Optimization was performed using the AdamW optimizer with a cosine learning rate schedule, and a warmup ratio of 50% was applied to stabilize the initial training steps. The

[2]This dataset contains 170M Latin tokens. It is stored in Hugging Face Fece228/latin-literature-dataset-170M.

Table 1
Gretino Score Foundational Model Adaptation

Model Name	Baseline	Model 1	Model 2	Model 3	Best Checkpoint
`Score (n/5)`	2.09	1.09	1.15	1.95	1.99

total training process for this phase was executed on an NVIDIA H100 GPU and produced multiple checkpoints, of which the best-performing checkpoint was selected based on Gretino dataset evaluation. By the end of Phase 1, the model was capable of properly handling Latin input, providing a necessary basis for the contrastive learning objective applied in the subsequent stage.

4.2. Phase 2: Finetuning for Semantic Retrieval

We proceeded to task-specific adaptation using the LoRA fine-tuning technique. This strategy targets the linear projection matrices within the Mamba architecture (`in_proj, out_proj, etc.`), using low-rank decomposition for efficient updates while keeping the vast majority of the model's parameters frozen. We utilized the PEFT library to implement the LoRA adapter. The training pipeline was structured in a modular repository, with distinct modules for configuration, data handling, and a custom evaluation callback to monitor performance on our specific retrieval task.

The fine-tuning process was driven by an unsupervised SimCSE[14] (Simple Contrastive Learning of Sentence Embeddings) objective. This method is particularly powerful because it does not require labeled data (e.g., manually annotated pairs of similar sentences). Contrastive Learning, the fundamental goal of SimCSE is to refine the embedding space such that semantically similar sentences are pulled closer together, while dissimilar sentences are pushed further apart. This is achieved by training the model to distinguish between "positive pairs" (sentences that should have similar embeddings) and "negative pairs" (sentences that should have different embeddings). The key innovation of unsupervised SimCSE is how it generates these pairs. A positive pair is created by feeding the same sentence through the model twice within the same training batch. Because neural networks, particularly during training, utilize dropout that randomly deactivate neurons, the two resulting embeddings will be slightly different but semantically identical. This creates a positive pair without external data or labeling. All other sentences in batch are treated as negative pairs relative to the original sentence.

Our implementation followed this principle directly. For each training step, our custom data collator took a batch of sentences and duplicated each one. The model then generated embeddings for this expanded batch. A custom loss function was then used to calculate a similarity matrix (using cosine similarity) between all embeddings. The training objective was to maximize the similarity score for the positive pairs while minimizing the scores for all negative pairs. A temperature parameter was used to scale the similarity scores, controlling the difficulty of the task and helping the model learn a more discriminative embedding space.

The fine-tuning phase required careful hyperparameter selection to balance convergence stability with strict training time constraints. We experimented with different learning rates and temperature values for the SimCSE objective, ultimately identifying the configuration that yielded the best-performing checkpoint on the Gretino evaluation. The final training was conducted on an NVIDIA H100 GPU with gradient checkpointing and memory-optimized settings, running for 3 epochs over the Latin literature dataset. The key hyperparameters were a learning rate of 3×10^{-5}, a warmup ratio of 0.05, and a batch size of 128 with gradient accumulation steps set to 2. The maximum sequence length was 256 tokens, and a temperature of 0.15 was used for the SimCSE objective. Optimization was performed with AdamW using a cosine learning rate schedule and a weight decay of 0.01. Training precision was set to bfloat16 (H100-optimized), and the total training time was capped at approximately 5 days (around 120 hours).

This setup produced the most stable and discriminative embedding space, as confirmed by benchmark evaluation. The resulting checkpoint was retained as the final model for semantic retrieval tasks.

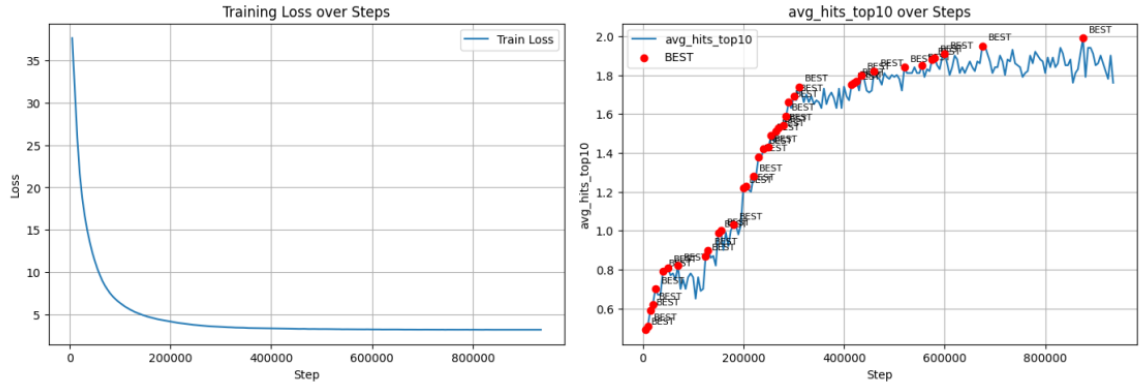

Figure 2: Training Loss and Average Hits @10 for best model in phase 1

Table 2
Gretino Score Advanced Fine-Tuning with LoRA

Model Name	Baseline	Phase 1 Best Model	Phase 2 Best Model
`Score (n/5)`	2.09	1.99	3.44

5. Evaluation

Phase 1, devoted to model adaptation, was periodically evaluated on the Gretino benchmark. We report training loss and `avg_hits@10` across steps. We ran three single-LR baselines where the embedding and output layer shared the same learning rate respectively of 1.5×10^{-5}, 2.5×10^{-5}, and 5×10^{-5}. As mentioned in section 4.2, the best checkpoint was obtained when decoupling the learning rates: 1.0×10^{-3} for the embedding layer and 7.5×10^{-4} for the output layer, with AdamW and cosine schedule, batch size 32, max sequence length 1024, and periodic Gretino evaluation. This setting is reported in Figure 2. We began with conservative single learning rates, but the experiments showed that increasing the learning rate and decoupling it for embeddings vs. the output layer was key to reaching the best checkpoint. The strongest single learning rate run achieved 1.95/5, while the per-layer learning rate configuration reached 1.99/5, and is used as the Phase-1 foundation for Phase-2 fine-tuning. Table 3 shows the benchmark results for retraining

In Phase 2 we initialized LoRA fine-tuning from the best Phase-1 checkpoint (`avg_hits@10` = 1.99/5), i.e. the model obtained after tokenizer extension plus embedding and output layer retraining. We ran unsupervised SimCSE with LoRA adapters, periodically evaluating `avg_hits@10` on Gretino. We explored several hyperparameters and focus here on the temperature, contrasting a run with $\tau = 0.20$ and the best run with $\tau = 0.15$. (Shared settings: LR 3e-5, batch 128 with gradient accumulation 2, max length 256, AdamW + cosine). Setting the temperature to $\tau = 0.15$ delivered steadier optimization and higher Gretino scores, indicating better local structure in the embedding space for top-10 retrieval. We initialized LoRA fine-tuning from the 1.99/5 Phase-1 checkpoint and explored temperature settings for SimCSE. A higher temperature ($\tau = 0.20$) pushed negatives too hard—improving over Phase-1 but limiting cluster cohesion—while $\tau = 0.15$ struck a better balance, yielding the best checkpoint at 3.44/5 `avg_hits@10`. In Table 2 we can observe the benchmark score retrieved for the different approaches of fine-tuning.

In terms of performances, the model prompted from the first phase of the pipeline was very similar to the original baseline. The final model, fine-tuned with the specialized LoRA technique, demonstrated a significant improvement in retrieval accuracy. The model successfully met our target performance goals, confirming that an architecture-aware PEFT method is critical for unlocking the full potential of the Mamba model.

To assess qualitative retrieval behavior, we report:

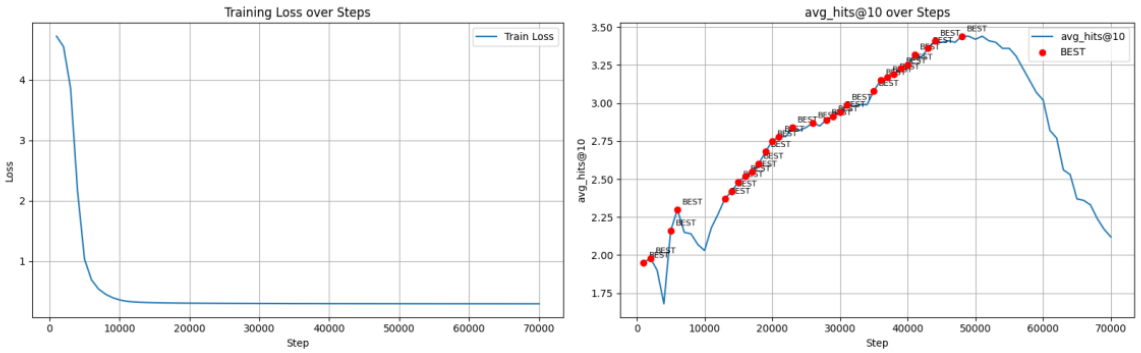

Figure 3: Training Loss and Average Hits @10 for best model in phase 2 ($\tau = 0.15$)

Table 3
Baseline vs Best SimCSE model Top10 and Top5 Coverage on Gretino

Measure	Baseline	Best SimCSE Model
`Avg_hits@10`	2.09/5	3.44/5
`Avg_hits@5`	0.75/5	2.95/5

Table 4
Baseline vs Best SimCSE model Top5 Matches over 5 hits

Matches over 5 hits	Baseline	Best SimCSE Model
5	0/100	18/100
4	1/100	18/100
3	7/100	21/100
< 3	92/100	43/100

- `avg_hits@10` and `avg_hits@5` in Table 3
- Per-query target coverage @5: for each of the 100 queries, we count how many of the 5 ground-truth targets appear in the top-5, and aggregate the distribution across queries: 5/5, 4/5, 3/5, and <3/5. This reveals how often the model returns complete or near-complete relevant sets at the very top of the ranking.

6. Conclusion

Although Transformer models continue to demonstrate superior performance (in our experiments, the finetuned Transformer models achieve an `Avg_hits@10` greater than 4 out of 5 on the semantic retrieval task[10]), the Mamba architecture represents a computationally efficient and high-performing alternative, also valuable for its ability to model long-context representations. Results on the Gretino dataset confirm the viability of this methodology for specialized, low-resource domains, marking the first pilot experiment employing a State Space Model for ancient languages.

It cannot be ignored that existing Mamba models are generative; in this work, however, we focused exclusively on the encoding component. The model we prompted can, in principle, generate Latin text, but the output quality is poor, and we do not currently see any scholarly utility in such material. By adopting a phased approach—initially retraining the tokenizer and embeddings, followed by fine-tuning with a SimCSE objective—we adapted a general-purpose English model (whose initial performance was very limited) into a tool for Latin text analysis.

Looking forward, this approach could be extended to other ancient languages, such as Ancient Greek, through dedicated corpus creation and tokenizer training. Further advances may arise from supervised

SimCSE, knowledge distillation using future Mamba models designed for English retrieval, or even training a State Space Model from scratch on large corpora of ancient languages, aiming to establish new standards for accurate, domain-specific linguistic representations.

Acknowledgments

This work was carried out within the Prototyping Lab, Use Case 9, Timebox 5 in the framework of the ITSERR (Italian Strengthening of the ESFRI RI RESILIENCE) project (cod. Progetto IR0000014 - CUP B53C22001770006) – Piano Nazionale di Ripresa e Resilienza (PNRR) – Finanziato dall'Unione europea - Next Generation EU, Missione 4, "Istruzione Ricerca", Componente 2, "Dalla ricerca all'impresa", Investimento 3.1, "Fondo per la realizzazione di un sistema integrato di infrastrutture di ricerca e innovazione" – rif. Avviso MUR 3264/2021. ITSERR is an interdisciplinary and distributed Research Infrastructure for Religious Studies that aims to strengthen the RESILIENCE RI project across Italy. The necessity of a semantic retrieval tool was raised by the case studies that Elia Scapini and Federico Iezzi approached during their work as Ph.D. students for the fourth work package (WP4) DaMSym (Data Mining: the Nicene-Constantinopolitan Symbolum) of the ITSERR project. Consequently, WP4 for Latin and Ancient Greek presented one Use Case dealing with semantic retrieval performed with LLMs. To pursue these objectives, the Ministero dell'Università e della Ricerca (MUR) has concluded a Grant Agreement (GA) with the ITSERR consortium for the provision of IT services to support and bring to a higher maturity level the existing national infrastructure. In this framework, the Fincons Group was appointed to support in software development the ITSERR infrastructure according to the presented Use Cases. Elia Scapini and Federico Iezzi are also members of FSCIRE, Fondazione per le Scienze Religiose in Bologna.

Declaration on Generative AI

During the preparation of this work, the authors used Generative AI (mostly GPT-4) and Grammarly for linguistic revisions. Authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[3] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. URL: http://arxiv.org/abs/1907.11692. doi:10.48550/arXiv.1907.11692. arXiv:1907.11692.

[4] D. Bamman, P. J. Burns, Latin BERT: A contextual language model for classical philology, 2020. URL: http://arxiv.org/abs/2009.10053. doi:10.48550/arXiv.2009.10053. arXiv:2009.10053.

[5] R. Sprugnoli, M. Passarotti, F. M. Cecchini, M. Pellegrini, Overview of the EvaLatin 2020 evaluation campaign, in: R. Sprugnoli, M. Passarotti (Eds.), Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages, European Language Resources

Association (ELRA), Marseille, France, 2020, pp. 105–110. URL: https://aclanthology.org/2020.lt4hala-1.16/.

[6] R. Sprugnoli, M. Passarotti, F. M. Cecchini, M. Fantoli, G. Moretti, Overview of the EvaLatin 2022 evaluation campaign, in: R. Sprugnoli, M. Passarotti (Eds.), Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages, European Language Resources Association, Marseille, France, 2022, pp. 183–188. URL: https://aclanthology.org/2022.lt4hala-1.29/.

[7] R. Sprugnoli, F. Iurescia, M. Passarotti, Overview of the EvaLatin 2024 evaluation campaign, in: R. Sprugnoli, M. Passarotti (Eds.), Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024, ELRA and ICCL, 2024, pp. 190–197. URL: https://aclanthology.org/2024.lt4hala-1.21.

[8] F. Riemenschneider, A. Frank, Graecia capta ferum victorem cepit. detecting Latin allusions to Ancient Greek literature, in: A. Anderson, S. Gordin, B. Li, Y. Liu, M. C. Passarotti (Eds.), Proceedings of the Ancient Language Processing Workshop, INCOMA Ltd., Shoumen, Bulgaria, 2023, pp. 30–38. URL: https://aclanthology.org/2023.alp-1.4.

[9] F. Riemenschneider, A. Frank, Exploring large language models for classical philology, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2023, pp. 15181–15199. URL: https://aclanthology.org/2023.acl-long.846. doi:10.18653/v1/2023.acl-long.846.

[10] H. O. Toyin, F. Iezzi, E. Scapini, G. Federico, G. Puccetti, Gretino: a Greek and Latin dataset to benchmark retrieval systems in classical languages, 2026.

[11] N. M. Cirone, A. Orvieto, B. Walker, C. Salvi, T. Lyons, Theoretical foundations of deep selective state-space models, 2025. URL: https://arxiv.org/abs/2402.19047. arXiv:2402.19047.

[12] C. Chandra, S. Balne, S. Bhaduri, T. Roy, V. Jain, A. Chadha, Parameter efficient fine tuning: A comprehensive analysis across applications, 2024. doi:10.13140/RG.2.2.10046.70729.

[13] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: https://arxiv.org/abs/2106.09685. arXiv:2106.09685.

[14] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, 2022. URL: https://arxiv.org/abs/2104.08821. arXiv:2104.08821.