Narrative Review Article

Comparative Evaluation of Latin Natural Language Processing Tools for Pedagogical Applications

Aidan Han

Ridgewood High School, 627 E Ridgewood Ave, Ridgewood, NJ 07450, United States

ABSTRACT

This narrative review compares contemporary Latin Natural Language Processing (NLP) tools and their use cases for pedagogical applications. While NLP has advanced significantly for high-resource modern languages, Latin remains underrepresented due to its complex morphology, flexible word order, orthographic variation, and limited annotated corpora. To address this gap, this review examines five prominent Latin NLP systems, LatinBERT, Stanza, LatinCy, LemLat 3.0, and Lamon/LamonPy, across architecture, training data, task performance, and educational relevance. Drawing on published evaluation metrics from peer-reviewed sources, including part-of-speech tagging, lemmatization, dependency parsing, and word sense disambiguation, this study analyzes the strengths and limitations of each approach. Transformer-based models demonstrate strong contextual understanding and high accuracy in disambiguation tasks, while rule-based systems offer transparency and reliability for vocabulary learning. Pipeline models provide comprehensive syntactic analysis but show performance variability across text genres. The reviewed evidence suggests that no single model performs optimally across all tasks or corpora, and that performance is strongly influenced by alignment between training and evaluation data. The review concludes that an integrated, multi-tool approach is most effective for supporting Latin pedagogy and outlines directions for future research, including standardized benchmarking and classroom-based evaluation.

Keywords: Latin; Natural Language Processing; Classical Languages; Language Pedagogy; Lemmatization; Part-of-Speech Tagging; Dependency Parsing; Universal Dependencies

INTRODUCTION

Natural Language Processing (NLP), a subfield of artificial intelligence focused on enabling computers to interpret, generate, and analyze language, has advanced rapidly over the past decade. NLP systems now lie beneath everyday technologies such as search engines, machine translation, grammar correction, and AI chatbots. While modern high-resource languages have greatly benefited from NLP systems, classical and historical languages remain comparatively underserved. These languages provide unique computational and resource challenges that make standard modeling difficult.

Among these, Latin, although no longer spoken in conversation, is undoubtedly a language that should be a focus. Its presence in academia across philosophy, history, theology, and linguistics, combined with the vast amount of unannotated textual material spanning hundreds of years, makes it both important and practical to develop NLP systems. The growth of digital classical linguistics, which applies computational methods and

Corresponding author: Aidan Han, E-mail: wjhan2020@gmail.com.
Copyright: © 2026 Aidan Han. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.
Accepted June 29, 2026
https://doi.org/10.70251/HYJR2348.441421

annotated corpora to classical languages like Latin, has made this more feasible and scholarly relevant (1). However, Latin poses a number of unique computational challenges. The language's highly inflected morphology, free word order, long-term spelling variation, and fewer annotated resources than those of other contemporary languages make it difficult to apply the same approaches to Latin as to modern languages.

A number of NLP resources now exist in response: transformer-based models like LatinBERT (2), Universal Dependencies pipelines like LatinCy's spaCy pipelines (3) and Stanza's Latin models (4), rule-based morphological analyzers like LemLat 3.0 (5), and the BiLSTM tagger Lamon/LamonPy (6). Broader frameworks such as the Classical Language Toolkit (CLTK) also use several of these components for pre-modern languages (7). Despite these many tools, these models are often evaluated in isolation, and few studies comparatively assess their potential performance for pedagogical use such as vocabulary learning, grammar, or reading comprehension, and support. This review addresses this gap by comparing leading Latin NLP tools through published performance data, as well as their architecture, data sources, task coverage, and educational potential, to identify potential pedagogical uses and the resources required.

This narrative review addresses that gap by synthesizing published performance data of five widely used Latin NLP tools, organized around four thematic dimensions: model architecture, training data, task performance, and pedagogical applications. The five tools were chosen for their prominence in contemporary Latin NLP work, their relevance to common annotation tasks and possible educational applications, and the availability of published performance data (2-6). Because no universal benchmark exists across these systems, metrics are drawn from each tool's primary source and compared qualitatively where alignment is possible. The metrics referenced throughout the review include UPOS (Universal Part-of-Speech tagging) accuracy, lemmatization accuracy, unlabeled and labeled attachment scores (UAS and LAS) for dependency parsing, morphological feature accuracy, and word sense disambiguation (WSD) F1 score. Many of these tools are evaluated against one or more Latin treebanks from the Universal Dependencies (UD) project (8, 9): ITTB (Index Thomisticus Treebank, regular scholastic prose), LLCT (Late Latin Charters Treebank, formulaic legal documents), PROIEL (Vulgate New Testament and classical prose), Perseus (classical literary works including poetry, building on the Ancient Greek and Latin Dependency Treebank (10)), and UDante (Dante's Latin works). These treebanks differ markedly by genre and time period, and that variation underlies much of the cross-model performance pattern discussed below. The review concludes by mapping each tool's strengths to specific pedagogical use cases and identifying directions for future research.

MODEL ARCHITECTURE

Latin NLP tools can be categorized into three broad architectural types, each with different trade-offs: rule-based systems, sequential neural networks, and transformer-based contextual models. Each architecture has different approaches that interact differently with Latin's unique features.

Rule-based systems such as LemLat 3.0 use predefined morphological rules and curated lexical resources to map word forms to candidate lemmas and morphological tags. The system uses an extensive lexical database from classical dictionaries, expanded to handle proper nouns and medieval lexical extensions (5). The architecture strips a token of its inflectional endings and matches the remaining stem against the lookup table, rather than learning from annotated treebanks or data. In educational settings, this approach is advantageous for its transparency and interpretability, but it lacks sentence-level contextual awareness. Due to its architecture, it struggles with disambiguating homographs without human intervention.

Neural network architectures process tokens sequentially, using probability and learned weights to infer syntactic roles from surrounding words. This overcomes the context issue of rule-based systems like LemLat. Models like Stanza (4) and Lamon/LamonPy (6) use variations of the Bidirectional Long Short-Term Memory (BiLSTM) network (11), often paired with a Conditional Random Field (CRF) decoding layer (12). Stanza provides a comprehensive, end-to-end neural pipeline that processes raw text through a series of models for tokenization, lemmatization, part-of-speech tagging, and dependency parsing, with separate Latin models trained on UD treebanks. It uses a BiLSTM-CRF architecture that processes sentences sequentially from both left to right and right to left, retaining a probabilistic memory of previous tokens to inform the tagging of the current token. This allows local and medium-range dependencies to be captured, but it cannot model long-distance relationships due to the sequential

propagation of information. Lamon/LamonPy is built on a BiLSTM tagger and lemmatizer. It uses supervised tagging trained on labeled data and is fast to run, but is constrained by corpus size and annotation quality. LatinCy is a CNN/tok2vec-based spaCy pipeline using static floret subword vectors. Burns presents LatinCy as a set of Latin-language core pipelines for spaCy, trained on available Latin data including all five Latin Universal Dependencies treebanks (3). The pipeline uses spaCy components and vector-based representations, including preprocessing decisions such as enclitic handling and orthographic normalization.

Transformer-based models such as LatinBERT (2) use self-attention and contextual embeddings (13), enabling them to capture word relationships across an entire sentence, regardless of distance, and to represent words in full context. LatinBERT adapts the BERT masked-language-modeling approach to Latin (14). This architecture is especially well-suited for Latin's flexible word order, since Latin's grammatical relationships do not rely on a word's position within a sentence and instead they rely on the words' endings. Self-attention enables each token in a sentence to be evaluated relative to every other token, allowing the model to assess which words are relevant when assigning a linguistic label. For example, a verb may reference either its subject or object across many words or in an unconventional word order, especially in poetry, but transformer models can connect those words. Unlike static embeddings that create a single vector for any given word form regardless of context, contextual embeddings create distinct representations for each occurrence of a particular form. Therefore, models can differentiate between identical surface forms that have different parts of speech or serve different semantic functions. This reduces errors caused by polysemy and homonymy, which are frequent in Latin, thereby increasing accuracy in POS tagging and lemmatization.

TRAINING DATA

Differences in training data play a major role in the accuracy of Latin NLP models across different domains. Performance generally increases when evaluation data closely resembles the syntactic regularity, genre, and annotation style of the training material, and decreases when texts introduce poetic structure, rare vocabulary, or historical variation. The five UD Latin treebanks introduced earlier provide a useful frame for understanding this pattern, since most pipeline models are trained and evaluated against subsets of them.

LatinBERT (2) is trained on a very diverse corpus spanning Classical, Medieval, and modern Latin, totaling approximately 642 million tokens drawn from the Perseus Digital Library, the Latin Library, the Patrologia Latina, the Corpus Thomisticum, Latin Wikipedia, and texts from the Internet Archive. This breadth allows the model to generalize well across genres, but performance still varies depending on the syntactic regularity of the evaluation corpus. POS tagging accuracy is highest on UD PROIEL (98.2%) and UD Index Thomisticus (98.8%), both of which contain relatively formal and predictable syntax. Accuracy drops on UD Perseus (94.3%), which contains a mixture of prose, poetry, and biblical texts that exhibit greater variation.

Stanza (4) is trained on annotated UD treebanks, and as a result, its performance is strongly tied to how closely the evaluation data matches its training distributions. Stanza achieves extremely high POS tagging accuracy on ITTB (98.76%) and LLCT (99.57%), both of which exhibit highly regular syntax and formulaic language similar to its training material. In contrast, accuracy drops on Perseus (91.40%) and UDante (90.08%), which are more irregular and feature poetry, unique word order, and rare inflected forms shaped by metrical constraints.

LatinCy (3) is trained on all five UD Latin treebanks and reports aggregate evaluation metrics for its sm/md/lg models (Table 1). The pipeline reports a high UPOS accuracy of 97.4% in the largest model. Because these scores are reported as aggregates rather than broken down by treebank, direct comparison with Stanza's per-treebank scores is limited; however, the same training-data dynamics that affect LatinBERT are likely to affect LatinCy as well, with degradation expected on more

Table 1. *LatinCy aggregate evaluation metrics for sm/md/lg models, reported across all five UD Latin treebanks (3).*

Type	sm score	md score	lg score
sentence segmentation f-score	.922	.931	.934
tagger accuracy (XPOS)	.932	.935	.941
tagger accuracy (UPOS)	.966	.969	.974
morphologizer accuracy	.915	.919	.928
trainable_lemmatizer accuracy	.939	.942	.947
parser accuracy (UAS)	.821	.818	.831
parser accuracy (LAS)	.764	.757	.776
ner f-score	.889	.892	.908

diverse literary texts. The pipeline also documents preprocessing decisions, including a custom tokenizer handling for the enclitic "-que" and an orthographic normalization step.

LemLat differs fundamentally from neural models in its training paradigm. Rather than learning from corpora, LemLat relies on large curated lexical databases and hand-encoded morphological rules derived from Classical Latin grammar. Its lexical basis is built from three dictionaries (Georges and Georges; Glare; Gradenwitz) totaling 40,014 lexical entries and 43,432 lemmas (5). The same underlying database now feeds the LiLa Knowledge Base, which links Latin lexical resources in the Semantic Web (15). Rather than treebank accuracy, the LemLat paper reports lexical coverage against a large diachronic vocabulary resource (TFTL): LemLat analyzed 400,886 of 554,826 distinct forms and could analyze 98.345% of occurrences in the TFTL texts. This produces high lemmatization accuracy for known word forms, but performance declines when encountering unknown forms or when context is necessary for disambiguation. Because LemLat does not learn from usage data, its performance does not improve with exposure to additional genres or periods — coverage can only be increased by expanding the underlying lexica.

Lamon and LamonPy (6) are trained on comparatively smaller corpora — 4,000 sentences from the Perseus UD treebank, supplemented by 440,000 sentences of self-training data — and prioritize semantic representation over formal grammatical annotation. Lamon reports evaluation on two test sets, Vivens and Perseus, with separate lemma accuracy, tag accuracy, and joint "both" accuracy figures (for instance, on Vivens: lemma 94.6, tag 83.0, both 81.1).

Across models, performance increases when evaluation data closely resembles the syntactic regularity, genre, and annotation style of the training material, and decreases when texts introduce poetic structure, rare vocabulary, or historical variation. Latin NLP performance is shaped not only by model architecture but by the interaction between training-data coverage and the linguistic diversity of the target corpus.

TASK PERFORMANCE

When examined across the major linguistic tasks of part-of-speech tagging, lemmatization, parsing, and morphological analysis, each model exhibits strengths and weaknesses that flow directly from its architectural and training-data choices. This section synthesizes published metrics by task rather than by tool, with the caveat that the metrics are drawn from heterogeneous studies and so support qualitative comparison rather than strict numerical equivalence.

Part-of-Speech Tagging

Transformer-based models such as LatinBERT (2) consistently achieve the highest POS-tagging accuracy across diverse corpora, with reported scores ranging from 94% to 98% across UD treebanks. LatinBERT's strength reflects its ability to integrate long-range contextual information (13), which is essential for resolving syntactic ambiguity in Latin. Performance remains high on structured corpora such as PROIEL and ITTB but declines on Perseus, where poetic syntax, enjambment, and stylistic variation increase ambiguity — a drop that reflects the intrinsic difficulty of POS tagging in heterogeneous literary Latin rather than a failure of the model itself.

Stanza (4) performs competitively on POS tagging in corpora with predictable syntactic patterns, achieving near-perfect accuracy on ITTB (98.76%) and LLCT (99.57%). However, its performance degrades more sharply than that of LatinBERT on Perseus (91.40%) and UDante (90.08%). This reflects the limitations of its BiLSTM-CRF architecture, which is less effective at modeling long-distance dependencies and non-canonical word order. In practice, Stanza excels in domains where Latin approximates the regularity of a controlled register but struggles in more expressive or poetic contexts.

LatinCy (3) reports a strong aggregate UPOS accuracy of 97.4% (lg model), but the absence of per-treebank breakdowns limits detailed comparison. Lamon (6) performs substantially worse on POS tagging, with accuracies between 80% and 83% across the Vivens and Perseus test sets — the lowest among the models surveyed. This lower performance reflects Lamon's task priorities rather than insufficient modeling: it is designed for semantic analysis rather than compliance with strict syntactic annotation schemes.

Lemmatization

Task-specific trends diverge more sharply for lemmatization. LemLat (5), a rule-based system, achieves 98.345% token coverage on the TFTL diachronic resource by deterministically mapping inflected forms to lemmas using extensive lexical resources. Its performance remains stable when lexical coverage is high, but declines when encountering unknown forms or

ambiguous tokens requiring contextual disambiguation. Stanza's (4) lemmatization scores range from 99.09% on ITTB and 96.86% on PROIEL down to 86.95% on UDante and 83.57% on Perseus, illustrating the same training-data sensitivity observed for POS tagging. LatinCy (3) reports an aggregate trainable-lemmatizer accuracy of 94.7% in its largest model, while Lamon (6) reports a lemma accuracy of 94.6% on Vivens. These figures show that no single model achieves uniformly high lemmatization accuracy across all corpora and conditions: rule-based systems dominate when lexical coverage is sufficient, while neural models are more flexible but degrade on rare vocabulary, poetic inflection, and proper names that are underrepresented in training data.

Dependency Parsing and Morphological Features

Dependency parsing and morphological-feature tagging are reported in fewer of the surveyed tools, but the available figures suggest similar patterns. LatinCy (3) reports labeled and unlabeled attachment scores (LAS .776 and UAS .831 in the lg model) and morphologizer accuracy of .928 — useful aggregate signals, though again without per-treebank breakdown. Stanza (4) provides treebank-specific dependency parsing scores in its official documentation and shows the same general drop on Perseus and UDante observed for tagging and lemmatization. LemLat (5) does not produce dependency trees; LatinBERT (2) likewise does not output explicit syntactic trees unless paired with an additional parser. For tasks requiring structural syntactic analysis, then, the field effectively narrows to neural pipeline systems.

Methodological Limitations

Direct numerical comparison across the metrics above is limited by methodological differences. Each tool evaluates using different treebanks, scripts, and annotation conventions, so a difference of points between two models reported in different studies does not mean that one model is always superior on a specific task. A cross-treebank evaluation of Latin morphological taggers shows that reported accuracy depends heavily on which treebank, time period, and feature set is used for testing, and that scores may not be directly comparable because of these conditions (16). Shared evaluation campaigns such as EvaLatin have begun to address this by providing common training and test data for Latin POS tagging and lemmatization (17). The published LatinBERT evaluation only reports POS accuracy for the PROIEL, ITTB, and Perseus UD treebanks, so LatinBERT cannot be compared to Stanza on the LLCT or UDante treebanks from the published data. Lamon's has reported data on the Vivens test set, which is not part of Universal Dependencies and therefore cannot be directly compared to the UD-based scores of other tools. Additionally, named entity recognition (NER) is reported only for LatinCy (NER F-score .889 to .908 across its sm/md/lg models) and is absent for the other tools. With these disjointed evaluations, it is cautioned against comparing the numerical scores cited throughout directly.

PEDAGOGICAL APPLICATIONS

Pedagogical Use and Evaluation Criteria

Pedagogical use refers to using NLP tools in a Latin learning activity that focuses on a specific instructional goal, for example, vocabulary acquisition, generating practice exercises, or grammar identification. Research on Intelligent Computer-Assisted Language Learning (ICALL) provides a useful framework for evaluation. Amaral and Meurers argue that classroom integration of NLP tools depends on how reliably NLP can match the design of the learning activity, and the tool's accuracy and computational cost (18). A tool is pedagogically useful when it is reliable, and its capabilities align with an activity's demands as well as the learner's level.

Different methodologies of teaching Latin may place different demands on NLP tools. For example, the grammar-translation and reading-based courses such as the Cambridge Latin Course and Ørberg's *Lingua Latina* focus on close reading and explicit morphology of set texts, while other approaches based on Krashen's Comprehensible Input and Teaching Proficiency through Reading and Storytelling (TPRS) emphasize more exposure to understandable Latin (19). Different NLP tools may suit different approaches better, for example, transparent rule-based tools may fit the focus of grammar and translation, while tools that generate large amounts of contextualized reading material fits the Comprehensible Input method. Using the ICALL criteria and these teaching methods, this section evaluates pedagogical relevance based on four dimensions: vocabulary acquisition, grammatical analysis, reading comprehension, and assessment or feedback.

Vocabulary Acquisition

Rule-based systems such as LemLat (5) are well-suited to the task of vocabulary acquisition, as it produces deterministic and transparent mappings from inflected forms to lemmas. This aligns with textbook and

vocabulary list-based curricula and supports activities such as vocabulary lookup and identification of headwords from inflected forms. Their inability to resolve ambiguity through context is less relevant at the stage where vocabulary acquisition is most important, as it should be the earliest stage of Latin learning and the basis on which learners continue, and the texts at this level are controlled to be simple. NLP can also help with vocabulary practice, as demonstrated by the Machina Callida e-tutor, which uses annotated Latin corpora to semi-automatically generate contextualized vocabulary exercises according to learning level, presenting words in authentic collocations to build recognition rather than in isolation and drawing attention to morphological agreement markers (20). Corpus-based language learning of this kind has been shown across studies to produce measurable gains in vocabulary and reading (21).

Grammatical Analysis

In the beginner and intermediate stages, learners should focus on grammatical analysis, identifying the case, number, tense, and grammatical relationships of words. POS tagging, morphological-feature tagging, and dependency parsing serve this competency, tasks that neural systems like LatinCy (3) and Stanza (4) are strongest on structurally regular Latin. They are reliable for automated annotation, syntactic feedback, and parsing exercises, as shown by their high accuracy on the ITTB and LLCT treebanks. However, both systems' lower accuracy on poetic and complex material like literary and metrical texts suggests caution, since automatically generated parses may be unreliable and should not be treated as unequivocal.

Reading Comprehension

Reading comprehension should be emphasized in all levels of Latin, but comprehension of authentic and syntactically complex Latin texts is the goal at intermediate and advanced levels, and here contextual disambiguation matters most. Transformer-based contextual models such as LatinBERT (2) are most useful at this stage, as their strength across genres and word sense disambiguation (75.4% accuracy) makes it usable for advanced texts where single forms have many different meanings in different contexts (22). This would help students move beyond literal translation and into a deeper understanding of text, with computational cost and interpretability as a tradeoff.

Assessment and Feedback

The ICALL framework is most demanding of accurate assessment and feedback, as tools should give accurate corrections to learners who rely on them (18). LatinCy's pipeline, which covers POS tagging, lemmatization, morphological features, dependency parsing, and named entity recognition (3), may be most fit for applying feedback, as a single annotation of text can support many different corrections of student error at once. However, reliability is a big constraint for these tools, as they lose accuracy on irregular or poetic Latin, so automated feedback is most trustworthy for controlled prose at beginner or intermediate levels, while they should always be supervised by a human teacher, especially for higher levels. Table 2 summarizes how these four pedagogical dimensions map to relevant NLP capabilities, best-fitting tools, example classroom uses, and the cautions each requires.

***Table 2**. Pedagogical mapping of NLP capabilities to Latin learning activities.*

Pedagogical goal	Relevant NLP capability	Best-fitting tools	Example classroom use	Required caution
Vocabulary acquisition	Lemmatization and morphological lookup	LemLat; LatinCy or Stanza as supplements	Students enter inflected forms from a passage and identify dictionary headwords before translation	Ambiguous forms should be checked against context and teacher guidance
Grammar identification	POS tagging, morphology, dependency parsing	Stanza; LatinCy	Students compare an automatic parse with their own case and function analysis	Use controlled prose before poetry; do not treat parses as infallible
Reading comprehension	Contextual representation and disambiguation	LatinBERT; LatinCy/ Stanza for support	Students examine how context changes the meaning of ambiguous forms or words	Transformer outputs need explanation because they are less transparent

***Continued Table 2**. Pedagogical mapping of NLP capabilities to Latin learning activities.*

Pedagogical goal	Relevant NLP capability	Best-fitting tools	Example classroom use	Required caution
Assessment and feedback	Integrated annotation pipeline	LatinCy; Stanza	A teacher reviews automatically generated feedback on agreement, lemma choice, or syntactic dependencies	Feedback should be teacher-supervised, especially for advanced or poetic texts
Corpus-based fluency practice	Exercise generation from annotated corpora	Machina Callida-style corpus tools; pipeline annotations	Students practice vocabulary in authentic collocations rather than isolated lists	Exercises must be matched to learner level and verified for accuracy

CONCLUSION

This research compared five of the top Latin NLP tools, including LatinBERT, Stanza, LatinCy, LemLat 3.0 and Lamon/LamonPy in terms of architecture, training data, task performance, and how applicable these models would be for teaching Latin. The performance data from published sources was then used to determine which of the available Latin NLP tools could be most useful for teaching Latin. A comparison of each tool's metrics was also done to show where similarities and differences existed among the tools. Overall, this study showed that there is no one single Latin NLP model with complete performance in every area or type of text. Neural models performed better than rule-based approaches for part-of-speech tagging and disambiguation: the transformer-based LatinBERT was strongest for contextual word sense disambiguation, while the neural pipelines LatinCy and Stanza achieved high tagging accuracy on syntactically regular prose. However, transformer-based models in particular require more computing resources and can be difficult to understand without additional tools, which may limit their use for some users. Rule-based models like LemLat 3.0 are particularly well-suited for learning new vocabulary because of their transparency; however, rule-based models do not make use of information about sentences. Stanza is an example of a model that includes a complete pipeline for performing syntactic analysis; however, for literary and poetic texts, the performance of Stanza drops significantly.

All models exhibit a single trend; they are less accurate when the genres of texts in the evaluation corpus diverge from those found in the model's training data. Therefore, as Latin students move from basic textbook compositions to increasingly difficult works of classical literature (such as poetry), the accuracy of tools that performed better with standard prose will decrease. The most effective way to implement this technology within a curriculum therefore appears to be through combining various tools. For example, while LemLat can be used for vocabulary lookup, other tools (like Stanza or LatinCy) that utilize neural networks can be employed for syntactic parsing and morphological tagging. Similarly, LatinBERT can be applied to higher-level functions such as word sense disambiguation.

There are also several limits to this research. Since the evaluation methodologies and reporting practices of the respective tools vary widely, all evaluations were conducted using the same metrics as presented in each tool's published documentation. A second limitation is that there was no external benchmarking procedure utilizing a common test corpus. Lastly, the results of the pedagogical assessment presented here were based on the degree of task relevance rather than actual student performance. Future research should include both a controlled test utilizing a shared UD benchmark across all the types of Latin texts that Latin students read and experience in their classrooms, as well as studies in actual teaching environments, to assess if the theoretical benefits discussed above result in real learning improvements.

CONFLICT OF INTEREST

The author declares that there are no conflicts of interest related to this work.

REFERENCES

1. Berti M, editor. Digital classical philology: ancient Greek and Latin in the digital revolution. Berlin: De Gruyter; 2019. doi:10.1515/9783110599572

2. Bamman D, Burns PJ. Latin BERT: A contextual language model for classical philology. arXiv: 2009.10053 [cs.CL]. 2020. Available from: https://arxiv.org/abs/2009.10053
3. Burns PJ. LatinCy: Synthetic trained pipelines for Latin NLP. arXiv:2305.04365 [cs.CL]. 2023. Available from: https://arxiv.org/abs/2305.04365
4. Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Stroudsburg (PA): Association for Computational Linguistics; 2020. p. 101–8. https://doi.org/10.18653/v1/2020.acl-demos.14
5. Passarotti M, Budassi M, Litta E, Ruffolo P. The Lemlat 3.0 package for morphological analysis of Latin. In: Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language. Linköping (Sweden): Linköping University Electronic Press; 2017. p. 24–31.
6. bab2min. Lamon/LamonPy: Latin POS tagger and lemmatizer [Internet]. Zenodo; 2020. Available from: https://doi.org/10.5281/zenodo.4091536 (accessed on April 25, 2026)
7. Johnson KP, Burns PJ, Stewart J, Cook T, Besnier C, Mattingly WJB. The Classical Language Toolkit: an NLP framework for pre-modern languages. *In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* 2021; p. 20–29. doi:10.18653/v1/2021.acl-demo.3
8. de Marneffe MC, Manning CD, Nivre J, Zeman D. Universal Dependencies. *Comput Linguist*. 2021; 47 (2): 255–308. doi:10.1162/coli_a_00402
9. Nivre J, de Marneffe MC, Ginter F, Goldberg Y, Hajič J, Manning CD, McDonald R, Petrov S, Pyysalo S, Silveira N, Tsarfaty R, Zeman D. Universal Dependencies v1: A multilingual treebank collection. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). Paris: European Language Resources Association; 2016. p. 1659–66. https://doi.org/10.63317/2hnj7od8afy8
10. Bamman D, Crane G. The design and use of a Latin dependency treebank. In: Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT 2006). Prague: ÚFAL MFF UK; 2006. p. 67–78.
11. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997; 9 (8): 1735–80. doi:10.1162/neco.1997.9.8.1735
12. Lafferty J, McCallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning (ICML 2001). San Francisco (CA): Morgan Kaufmann; 2001. p. 282–9.
13. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NeurIPS 2017). 2017. p. 5998–6008. arXiv:1706.03762
14. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT 2019. Minneapolis (MN): Association for Computational Linguistics; 2019. p. 4171–86. doi:10.18653/v1/N19-1423
15. Mambrini F, Cecchini FM, Franzini G, Litta E, Passarotti MC, Ruffolo P. LiLa: Linking Latin. Risorse linguistiche per il latino nel Semantic Web. *Umanist Digit.* 2020; 4 (8). doi:10.6092/issn.2532-8816/9975
16. Hudspeth M, O'Connor B, Thompson L. Latin treebanks in review: an evaluation of morphological tagging across time. In: Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024). Bangkok, Thailand and online: Association for Computational Linguistics; 2024. p. 203–18. doi:10.18653/v1/2024.ml4al-1.21
17. Sprugnoli R, Passarotti M, Cecchini FM, Pellegrini M. Overview of the EvaLatin 2020 evaluation campaign. In: Proceedings of LT4HALA 2020 — 1st Workshop on Language Technologies for Historical and Ancient Languages. Marseille: European Language Resources Association; 2020. p. 105–10.
18. Amaral LA, Meurers D. On using intelligent computer-assisted language learning in real-life foreign language teaching and learning. *ReCALL*. 2011; 23 (1): 4–24. doi:10.1017/S0958344010000261
19. Venditti E. Using Comprehensible Input in the Latin classroom to enhance language proficiency. *J Class Teach*. 2021; 22 (43): 22–8. doi:10.1017/S2058631021000039
20. Kuehnast M, Schulz K, Lüdeling A. Development of basic reading skills in Latin: a corpus-based tool for computer-assisted fluency training. *Cogent Educ.* 2024; 11 (1): 2416819. doi:10.1080/2331186X.2024.2416819
21. Boulton A, Cobb T. Corpus use in language learning: a meta-analysis. *Lang Learn*. 2017; 67 (2): 348–93. doi:10.1111/lang.12224
22. Navigli R. Word sense disambiguation: a survey. *ACM Comput Surv.* 2009; 41 (2): Article 10. doi:10.1145/1459352.1459355