Improving Latin Dependency Parsing by Combining Treebanks and Predictions

Hanna-Mari Kupari, Erik Henriksson, Veronika Laippala, Jenna Kanerva
TurkuNLP, University of Turku, Finland
{hanna-mari.kupari, erik.henriksson, mavela, jmnybl}@utu.fi

Abstract

This paper introduces new models designed to improve the morpho-syntactic parsing of the five largest Latin treebanks in the Universal Dependencies (UD) framework. First, using two state-of-the-art parsers, Trankit and Stanza, along with our custom UD tagger, we train new models on the five treebanks both individually and by combining them into novel merged datasets. We also test the models on the CIRCSE test set. In an additional experiment, we evaluate whether this set can be accurately tagged using the novel LASLA corpus (https://github.com/CIRCSE/LASLA). Second, we aim to improve the results by combining the predictions of different models through an atomic morphological feature voting system. The results of our two main experiments demonstrate significant improvements, particularly for the smaller treebanks, with LAS scores increasing by 16.10 and 11.85%-points for UDante and Perseus, respectively (Gamba and Zeman, 2023a). Additionally, the voting system for morphological features (FEATS) brings improvements, especially for the smaller Latin treebanks: Perseus 3.15% and CIRCSE 2.47%-points. Tagging the CIRCSE set with our custom model using the LASLA model improves POS 6.71 and FEATS 11.04%-points compared to our best-performing UD PROIEL model. Our results show that larger datasets and ensemble predictions can significantly improve performance.

1 Introduction

In recent years, significant progress has been made in morpho-syntactic dependency parsing for Latin, an advancement that greatly benefits a wide range of research in the humanities. Linguistically tagged corpora are crucial, as lemmatized corpora, for instance, are valuable also for historians searching for sources within databases. The Universal Dependencies (UD) framework plays a key role by organizing linguistic analysis into machine-readable databases with columns in tab-separated value tables. These CoNLL-U formatted treebanks provide essential information on lemmas, parts of speech, morphological features, syntactic roles, and dependency relations. In the realm of Latin treebanks notable recent developments include the morphological harmonization of the five largest Latin treebanks (ITTB, LLCT, Perseus, PROIEL, and UDante[1]), a significant milestone reached by Gamba and Zeman (2023a) as a continuation of earlier work on syntactic harmonization (Gamba and Zeman, 2023b).

Additionally, there have been many efforts to enhance the performance of Latin parsing tools. These include the EvaLatin campaigns Sprugnoli et al., 2022, 2024, as well as the application of GPT models for part-of-speech (POS) tagging (Stüssi and Ströbel, 2024). Despite these advancements, there remains potential for further improvement, particularly in syntactic parsing. For instance, the highest Labeled Attachment Score (LAS) reported by Gamba and Zeman (2023a) is 64.87% for the UDante and 59.43% for Perseus.

In the present study, we leverage the recently released harmonized treebanks (Gamba and Zeman, 2023a) to further enhance automatic parsing. Our focus is on the five largest established treebanks in the UD format, ensuring that our results are reliably comparable to previous studies. Our models can also easily be applied to parse new text corpora.

To achieve our goal, we employ two approaches: First, we train new parser models using these harmonized treebanks, along with two state-of-the-art parsers —Stanza (Qi et al., 2020) and Trankit (Nguyen et al., 2021)— as well as a custom UD tagger by fine-tuning a BERT-based Latin language model (Ströbel, 2022) following the architecture of Devlin et al. (2019). The parsing models are trained using both individual and diverse merged

[1] https://universaldependencies.org/la/

Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities, pages 216–228
November 16, 2024. ©2024 Association for Computational Linguistics

treebanks.

Second, we investigate whether combining predictions from our newly trained models in a voting system targeting part-of-speech (POS) and morphological features (FEATS) tags improves performance. Our hypothesis is that selecting the most common prediction from the different models enhances the results in a 'majority vote wins' scenario.

Third, we use the voting setup of the different models to analyze how unanimous the various parser models are in their POS predictions. This provides insight into which tasks are accurately tagged and offers potential for identifying prevailing issues in the annotation guidelines.

Upon the publication of this paper,[2] all data, code, and results, as well as the models, will be made openly and freely accessible for non-commercial use. These resources include clear instructions, designed to be easily used by scholars who may not be familiar with language technology but wish to experiment with their own texts.

2 Previous work

The first Latin BERT model by Bamman and Burns (2020) provided the state-of-the-art POS scores of its time (Perseus 94.3%, PROIEL 98.2%, ITTB 98.8%). Similarly, Nehrdich and Hellwig (2022) reported very competitive LAS scores for the previous releases of the treebanks using a biaffine parser on top of a Latin BERT (ITTB 92.99%, PROIEL 86.34% and PERSEUS 80.16%).

There have been some trials with merging existing treebanks into larger training datasets. Nehrdich and Hellwig (2022) combined the ITTB, Perseus, and PROIEL treebanks, while Smith et al. (2018) trained a single model for all ancient languages, including three Latin treebanks. Additionally, Kondratyuk and Straka (2019) combined all the UD treebanks into a single multilingual dataset and trained a model for all UD languages. While these studies demonstrated the potential for improving performance by merging training data from multiple treebanks, the first reports only a single experiment, and the latter two do not focus specifically on Latin, leaving room for further experiments. The challenge of selecting and combining treebanks is also brought to attention in the latest EvaLatin Campaign (Sprugnoli et al., 2024).

Merging treebanks for training models has not been widely explored, likely because the developers of the treebanks have varied interpretations of the UD guidelines since the treebanks have been composed at different points in time (with continuous updates regarding the annotation guidelines). These discrepancies in annotations has complicated combining them into larger merged training datasets. The work of Gamba and Zeman (2023a) focuses on the harmonisation of the datasets, and they train models using only the individual treebanks.

Combining the predictions of several models through voting has been tested in many studies. E.g. early pioneering work by Zeman and Žabokrtský (2005) applied majority voting for four parsers for Czech, reporting improvements of 2%-points in dependency relation prediction. Combining parser outputs has also been used by Passarotti and Dell'Orletta (2010) to improve the parsing of the ITTB treebank. More recent work by Stoeckel et al. (2020) developed an ensemble classifier by applying a voting model on top of several POS taggers. Their voting model was designed to learn which predictions to trust in different contexts.

3 Data

There are five Latin UD treebanks used for training: the Index Thomisticus Treebank (ITTB) (Passarotti, 2019), the Late Latin Charter Treebank (LLCT) (Cecchini et al., 2020b), Perseus (Bamman and Crane, 2011), PROIEL (Haug and Jøhndal, 2008), and UDante (Cecchini et al., 2020a). For a concise numerical comparison of these Latin UD treebanks and a detailed description of their contents, see [3]. For a general overview, see Gamba and Zeman (2023b).

The efforts of Gamba and Zeman (2023a) are crucial for merging the treebanks and serve as a foundation of our model training. These harmonized treebanks are accessible at a GitHub-repository [4]. For a concise numerical overview and a brief description of the treebanks used in this study, refer to Table 1.

[2] https://github.com/HannaKoo/Latin-Parsing

[3] https://universaldependencies.org/treebanks/la-comparison.html

[4] https://github.com/fjambe/Latin-variability/tree/main/morpho_harmonization/morpho-harmonized-treebanks

3.1 CIRCSE test set

The novel sixth UD Latin treebank, CIRCSE[5], consists solely of a test set because of its small size along the UD guidelines. This test set is valuable for evaluating our models because it differs from the established larger treebanks, which predominantly feature texts from the middle ages. For instance, the ITTB and LLCT together contain 692K tokens, whereas Perseus focuses on Classical texts with a total of only 29K tokens. CIRCSE is also distinct in genre, featuring a total of 13,294 tokens of tragedy: *Hercules Furens* (7,714 tokens, 555 sentences) and *Agamemnon* (5,580 tokens, 409 sentences) by Seneca (c. 4 BC – AD 65), along with the treatise *Germania* (5,674 tokens, 299 sentences) by Tacitus (c. AD 56 – c. 120).

3.2 Merged treebanks

Merging treebanks presents challenges not only due to potential differences in annotation guidelines but also because of the linguistic variation they reflect. The five treebanks span several millennia and cover a wide range of genres, factors that can influence the performance of models trained on them. One of the key research questions we explore is whether, for example, the inclusion of a large amount of medieval Latin training data affects the parsing results for Classical Latin.

In addition to merging all the training datasets, we combine the individual treebanks into five thematically organized merged treebanks, as shown in Table **??**, based on a holistic understanding of the nature of the different Latin UD datasets. We also experiment with merged sets focused on specific time periods, drawing on a heuristic understanding of historical linguistics and the evolution of the Latin language. The goal is to compile sets that support one another, rather than confuse the models with training data that is too varied or even contradictory. Beyond linguistic considerations, to address machine learning challenges and mitigate the risk of overfitting—particularly when working with datasets from unequally sized and heterogeneous treebanks—the merged training sets were constructed by iteratively concatenating one-fifth of each individual treebank, ordered from smallest to largest, into the new datasets.

3.3 The Corpus Corporum monolingual training set

While most of our experiments are based on the widely applied Stanza and Trankit parsers (see Section 4), neither of them support using a dedicated pre-trained Latin language model. Therefore, we also experiment with our custom tagger utilizing a language model trained on Latin data only (see Section 4.3). The language model (Ströbel, 2022) has been produced by using the Corpus Corporum dataset (Roelli, 2014). This dataset contains a considerably large portion of patristic texts from the Patrologia Latina (8.4 M words). For a concise overview of the texts currently included in this database see the listing on the project website [6]. The previous work of Bamman and Burns (2020) with a monolingual model for POS tagging is produced with a very large dataset of 642.7M tokens that includes for example Latin Wikipedia of 16M tokens. This provides obvious problems as to reliable quality of the training data, since contributions to Vicipaedia are not subject to expert language check and the RoBERTa Latin model by Ströbel (2022) is focused to solve this very issue.

3.4 The LASLA dataset

Since texts from the Classical period are underrepresented in the UD treebanks, we conduct a small experiment using the non-UD LASLA dataset, which lacks dependency parsing annotation. In terms of POS tagging, lemmatization and morphology, the 1.8M-token LASLA dataset is notably large, created through a joint effort by members of the LiLa and LASLA teams.[7] We use the LASLA corpus as a basis to make our own train, dev, and test sets for a small-scale experiment aimed at improving our custom model for the POS and morphological analysis of the CIRCSE test set. Our modification of the CoNLL-U Plus formatted files excludes the texts in the CIRCSE test set (see 3.1) and removes non-relevant fields. The larger files are split and concatenated in random order.

4 Methods

In our aim to improve morpho-syntactic parsing tools for Latin, we use two different methods: training new models and experimenting with a voting system. Our first task is the training of new parser

[5] https://universaldependencies.org/treebanks/la_circse/index.html

[6] https://mlat.uzh.ch/browser?path=/

[7] https://github.com/CIRCSE/LASLA?tab=readme-ov-file

Token counts or words in datasets					
Dataset	**Short Description**	**Train**	**Dev**	**Test**	**Total**
CIRCSE	Seneca's tragedies and Tacitus' treatise	-	-	19 483	19 483
ITTB	Texts of Thomas of Aquinas, 13th century	392 017	29 968	29 920	451 905
LLCT	8th century legal charters from Tuscany	194 193	24 195	24 079	242 467
Perseus	Classical auctors e.g. Caesar and Ovid	16 859	1 566	11 149	29 574
PROIEL	Classical auctors and New Testament	172 261	13 955	14 114	200 330
· Classical	E.g. Cicero and Palladius	76 647	-	-	76 647
· Vulgate	Jerome's Vulgate	95 614	7 123	-	102 737
UDante	Works of Dante Alighieri, 13th-14th century	30 567	11 689	13 502	55 758
CC	Massive Corpus Corporum text database				162 M
LASLA	Classical Latin database	1 856 296	32 756	35413	1 856 296

Table 1: Overview of the used datasets for train, dev and test. We have spilt PROIEL to include Classical secular texts and Vulgate. For Perseus, where the original release does not include a separate development set for parameter optimization, we created one by dividing the train set. The UD CIRCSE treebank only contains a test set due to its size. The Corpus Corporum dataset is the basis for the monolingual BERT (Ströbel, 2022) used for our custom model UD tagger. Our modification of the LASLA database (`https://github.com/CIRCSE/LASLA/tree/main`) is used in an experiment to improve the results of the CIRCSE test set.

Training data	**ITTB**	**LLCT**	**Perseus**	**PROIEL**	**UDante**	**Tokens in total**
Classical Latin			9%	91%		205 K
Late and Medieval Latin	62%	32%			6%	683 K
Later and Christian Latin	54%	28%		13%	5%	785 K
Merged	48%	25%	2%	21%	5%	887 K

Table 2: Overview of the merged treebanks used for training Stanza and Trankit and fine-tuning the custom model.

models based on the newest treebanks described in Tables 1 and 2. For full morpho-syntactic parsing, we apply the commonly used Trankit (Nguyen et al., 2021) and Stanza (Qi et al., 2020) toolkits. As neither Trankit nor Stanza support the usage of a custom pretrained language model, we also experiment with a custom part-of-speech and morphological tagger trained on top of a monolingual Latin language model (Ströbel, 2022) following the task-specific fine-tuning of Devlin et al. (2019).

4.1 Trankit

Trankit (Nguyen et al., 2021) is a light-weight transformer based toolkit, which provides a trainable pipeline for morpho-syntactic parsing. It reports outperforming prior multilingual NLP pipelines over sentence segmentation, POS and FEAT tagging as well as in dependency parsing while maintaining competitive performance for tokenization, multi-word token expansion, and lemmatization over 90 UD treebanks. It is based on training adapter modules (Houlsby et al., 2019; Pfeiffer et al., 2020) on top of the multilingual pretrained XLM-R language model (Conneau et al., 2020). The parser is designed to be efficient in multilingual usage (shared multilingual language model), while still giving state-of-the-art results for individual treebanks (treebank-specific adaptors).

4.2 Stanza

Stanza (Qi et al., 2020) is a trainable, language-agnostic neural pipeline for morpho-syntactic parsing. Stanza includes a Bi-LSTM encoder capable of utilizing pre-trained word embeddings, and uses the biaffine neural dependency parser by Dozat and Manning (2017). This is the same parser that Gamba and Zeman (2023a) employed. We use standard model training in order to have a model that matches the Trankit training to ensure a reliable comparison between the models.

4.3 Custom tagger with a Latin language model

Earlier studies, e.g. Pyysalo et al. (2021); Bamman and Burns (2020), have shown that for certain languages the usage of a dedicated monolingual language model may result in better performance compared to multilingual models or not using a pretrained language model at all. While neither Trankit nor Stanza support the usage of a custom pretrained language model, we implement a POS and morphological tagger by fine-tuning a monolingual Latin language model. As a pretrained language model we use the pstroe/roberta-base-latin-v3[8] pretrained on the Corpus Corporum Latin text collection (see Section 3.3). The tagger jointly predicts the POS and morphological features by adding a task-specific token classification layer on top of the pretrained language model, following the architecture of Devlin et al. (2019). The classification layer is trained on treebank data updating also the weights of the original language model.

4.4 Voting

In POS tag and FEATS predictions voting we run a simple majority vote of the three parsers (Trankit, Stanza, and Custom tagger), for each treebank selecting the generally best performing model of each parser. In a tie situation, the voting defaults to Trankit which generally receives the best individual scores. The voting script does not take into account the fact that the numerically highest scores for POS and UFEATS might come from different models, and our preference is for overall best results.

For POS tags, the possible voting scenarios when using three parsers are cases where all three agree, two outvote the third one and all parsers disagree. When analysing the model predictions for the Perseus treebank, in 86% of tokens the three parsers agree on UPOS, in 13% of tokens there is a majority agreement, and only in a bit more than 1% all three parsers disagree on UPOS.

However, in terms of morphological features the same agreement rates on Perseus are 59%, 31%, and 10% respectively, when voting on the level of full feature analyses — the entire FEATS field that consists of several categories such as number and tense. The large variation in predicted feature combinations therefore increases the percentage of tokens where there is no majority consensus available (10%).

To be able to at least partially account for these tokens as well, for morphological features we proceed the voting in two steps. First, the voting is done on the level of full feature analysis (e.g. for nouns this means that all the diverse elements in the category, such as case, number and gender), but in cases where we are not able to find a majority vote, we continue to the second option of voting on category level. In the second step, the feature analyses are split into individual (category, value) -pairs, and for each category we run the majority voting of values predicted for that particular category. To avoid the situation where the final analysis is a union of different categories predicted by three parsers, we obtain the categories from the default Trankit parser, therefore in practice only voting values for Trankit predicted categories. It should also be noted that the LASLA model for CIRCSE is not included in the vote, as it would require a close reading of potentially non-UD-style morphological annotations, which the script does not consider.

5 Results

The performance of the trained models is summarized in Table 3, which presents the results for the five largest established treebanks. Additionally, the outcomes specific to the CIRCSE treebank are detailed in Table 6 and Table 7. The findings underscore the importance of selecting optimal treebanks for training, as discussed by Sprugnoli et al. (2024). While the prevailing trend in training large language models has been to utilize increasingly larger datasets, our results indicate a different effect. Specifically, the Perseus treebank shows significant improvement when trained with the Classical dataset, indicating that quality of data is more critical than quantity, challenging the assumption that "more is better". The effects of this improvement are highlighted in Table 8.

The complete set of metrics is available on the project's GitHub page[9] and the all CoNLL-U formatted treebanks respectively[10]. In this paper, we report and discuss the scores for tokenization, POS, morphological features (FEATS), lemmatization, and syntax, including both the unlabeled attachment score (UAS) and labeled attachment score (LAS). For the custom tagger, only the UPOS and FEATS results are relevant. All metrics were generated using the UD evaluation tools, based on the CoNLL 2018 shared task script[11].

In the results presented below we discuss the

[8] https://huggingface.co/pstroe/roberta-base-latin-cased3

[9] https://anonymous.4open.science/r/Latin-Parsing-627B//Results/Evaluation_metrics/eval_table.tsv

[10] https://anonymous.4open.science/r/Latin-Parsing-43B5/Results/conllu_files

[11] https://github.com/UniversalDependencies/tools/blob/master/eval.py

Compilation of Results	**Tasks:**					
Treebank and model	**Tokens**	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**
ITTB						
Stanza	**100.00**	98.64	96.16	**99.05**	88.50	86.61
Trankit	99.99	98.99	97.52	97.63	**92.09**	**90.71**
Trankit Late and Christian	100.00	99.05	**97.61**	97.87	91.86	90.52
Trankit Five Merged	99.99	**99.07**	97.55	97.82	91.90	90.41
Custom tagger Late and Christian	-	98.72	96.61	-	-	-
LLCT						
Stanza	**100.00**	99.61	96.95	**98.07**	95.85	94.83
Trankit	99.99	**99.66**	**97.36**	96.50	96.15	95.37
Trankit Late and Medieval	99.99	99.66	97.18	96.69	**96.46**	**95.51**
Custom tagger	-	99.14	95.67	-	-	-
Perseus						
Stanza	**99.94**	89.44	80.17	80.97	69.75	61.93
Stanza Classical	99.92	90.09	81.33	**85.89**	75.28	68.29
Trankit Classical	99.74	90.50	**83.25**	74.60	**77.89**	**71.28**
Trankit Five Merged	99.79	**91.83**	80.94	76.55	77.72	70.59
Custom tagger Classical	-	89.58	82.58	-	-	-
Custom tagger Five Merged	-	89.66	78.43	-	-	-
PROIEL						
Stanza	**99.99**	97.22	92.14	**96.63**	78.12	74.56
Trankit	99.87	97.29	92.77	89.37	**84.09**	**80.97**
Trankit Five Merged	99.88	**97.30**	**92.96**	89.24	83.94	80.92
Custom tagger Five Merged	-	96.44	91.64	-	-	-
UDante						
Stanza	99.65	89.98	81.00	**86.94**	68.37	59.15
Trankit Five Merged	**99.66**	**91.46**	**84.42**	77.50	**79.63**	**73.42**
Custom tagger Five Merged	-	89.91	82.24	-	-	-

Table 3: A compilation of the most important F1-scores. The best score for each treebanks is in bold.

most relevant numbers and some case study examples. In Table 9 we also include the previous state-of-the-art outcomes from two recent studies. Our state-of-the-art results demonstrate improvements in POS-tagging of 8.41 %-points for Perseus, 7.78 for PROIEL, and 5.93 for UDante compared to the findings of Stüssi and Ströbel (2024). Additionally, our results show an improvement in LAS of 11.85%-points for Perseus and 16.10%-points for UDante compared to Gamba and Zeman (2023a).

All numerically highest F1 scores achieved by the models are in the Table 3. The effects of the merging of training data set for training are in Table 8. The results of the majority vote win for POS and FEATS are in Table 5.

5.1 Tokenization

Tokenization results have very little room for improvement, the best models already obtaining an F1 score of 100 % for ITTB and LLCT with individual training. From close reading we find that the only aspect of tokenization that requires improvement is the prediction of multi-word tokens (MWTs). This issue arises from the complete absence or inclusion of only a few trivial MWTs in these corpora. E.g. the ITTB train set contains only instances of *nonne* ’isn’t it?’, which is clearly insufficient for effectively training the models on something as complex as Latin enclitics). Upon close reading the output, we identified predictions that are significantly off. For instance, in the Perseus corpus parsed by Stanza, the word *pulsabantque*’and beat’ is incorrectly tokenized as "*pullaaa*" and "*que*" instead of the correct "*pulsabant*" and "*que*,"

The tokenization of the **CIRCSE** test set achieved a perfect accuracy of **100.00%** with the Stanza PROIEL model. However, this test set lacks punctuation, leading to poor performance in the task of **sentence segmentation** across all models. Several of our models were unable to segment sentences and attempted to dependency parse the entire dataset as a single 19K words long sentence. To address this, we experimented using a crude fix of adding a full stop at the end of each sentence using a script, and assigned a mock HEAD-tag pointing to the last word of each sentence, resembling the use of GS segmentation. For further details and results of this experiment, see Table 7.

5.2 Part-of-Speech (POS)

Overall, the results for POS tagging have for a long time highly accurate and for most treebanks can only be marginally improved.

All the results of the POS vote are written in a new ConLL-U-styled tsv-table that first includes the winner of the majority vote, the predicted forms in the following order: Trankit, Stanza and custom model.[12] After that a column indicates the results of the vote being either unanimous, two-to-one or even. The resulting file[13] includes also a column that indicates if the result of the vote is correct, this information is especially informative for close reading. Scholars are able to form a general idea of what kind of tasks the parsers are capable of predicting and can especially focus on the difficulties and understand if there is an underlying trend that could be fixed (i.e. relating to the annotation guidelines).

The most interesting cases are the ones with dispersed results and here we will highlight some case examples. From the ITTB treebank we find a case with the word *necesse-esse* '*necessarily existent*' with POS predictions: ADJ, VERB, AUX where our custom model gets it right according to the GS of the morphological harmonization, but the earlier realise tags this as NOUN. From LLCT we find instances like *decimas* (from phrase *per quadraginta annos abuerunt consuetudo offertas et decimas dare ad predicta ecclesia*) as ADJ, NOUN, NUM, where Stanza gets the POS tag of 'tithes' right. There are a lot of expressions of date, for example *in mense december* where one instance the vote is even for *december* resulting as NOUN, ADJ, NUM while all other instances in the test dataset get it unanimously right as ADJ. From the expression *adfinis terra* 'boundaries of the land' *adfinis* as ADJ, NOUN, ADP when Trankit gets it correct. The Perseus and UDante outputs have substantially more even votes than the other five established treebanks. These include *iuro* NOUN, VERB, ADJ from the phrase *per flumina iuro* (swear by the rivers), also we find *Aeoliis* as ADJ, PROPN, ADP where Trankit gets it correct. From PROIEL *promissa* as NOUN, ADJ, VERB from the expression *ceterorum que promissa* which is easy to understand, since the participe *promissum* 'promises' and we would also imagine this being difficult for Latin students, but Trankit is correct with NOUN. From UDante the phrase *praedictis finibus* 'of the aforementioned borders' where the participle is predicted as DET, VERB, ADJ and only Stanza is correct. An example of voting accuracy in PROIEL in Table 4.

PROIEL	Correct	%	Wrong	%
Unanimous	13 295	99%	132	0.98%
Two to one	463	75%	154	25%
Dispersed	15	44%	19	55%

Table 4: An example of the accuracies of the voting on POS tagging in PROIEL

For **CIRCSE**, the best UD framework based model part-of-speech tagging result comes from Stanza trained on PROIEL at 84.46%, but other models are close. However, our small experiment with the LASLA model does bring an improvement of 6.71%-points (UPOS **91.17%**) hinting that the results for many other out of genre texts from the Classical period might be considerably improved with larger training data.

5.3 UFeats

The morphological analysis results seem to vary greatly between different treebanks, from **ITTB** reaches already a very impressive result of **97.61%** but for **UDante** only at **84.42%**. This seems to follow the trend, that when there is enough of in-domain training data, the results have very little room for improvement. The best UD framework based **CIRCSE** morphological analysis is achieved with the Stanza PROIEL model (59.48%) as was for POS. Surprisingly using the LASLA model gives an improvement of 11.04%-points (UFeats **70.52%**).

5.4 Lemmas

Accurate automatic lemmatization is a very relevant task for a highly inflected language like Latin. The results have a high amount of variation across different treebanks but overall Stanza models seem to consistently outperform on this task. The re-

[12] Results/conllu_files/voted_extended

[13] Results/conllu_files/gold_extended

sults for **ITTB** comes from the Stanza individually trained model at an impressive **99.05%** as well as for **LLCT 98.07%**, **PROIEL 89.24%** and **UDante 86.94%**. For **Perseus** the best score **85.89%** is produced by using the Stanza Classical model.

The best lemmatization score for **CIRCSE** is Stanza Five Merged **78.00%**.

5.5 Unlabeled Attachment Score: UAS

Latin papers on automatic parsing usually report the unlabelled attachment scores (UAS) along with labeled attachment score (LAS). The UAS metric means the percentage of words that are assigned the correct head in the sentence. The results syntactic tagging vary greatly. The **Perseus** treebank benefits from only seeing training data from its own time period. On the contrary, the same does not apply for **UDante**, which benefits greatly from the merged training data and obtains a **79.63%** score with Trankit Five Merged (66.79% on UDante).

For **CIRCSE** the best score is only **51.29%** by the Trankit Five Merged model, this is understandable considering how far the training model data is as genre for parsing the tragedies. Adding the punctuation with a very coarse simple full stop addition at the end of each sentence makes this dataset much easier for models to syntactically parse, this alone leads to a 59.16% with above mentioned model.

5.6 Labeled Attachment Score: LAS

For the second metric on syntax, the Labeled Attachment Score LAS, the results are in line with UAS findings. The LAS score is the percentage of words that are assigned both to the correct head and the correct dependency label. The results in Table 3 show that the results tend to be dependent on the amount of similar training data.

The LAS score of the **CIRCSE** test set shows the true nature and difficulty of out of domain Latin syntax parsing. Our experiment reflects the more of a real life situation with parsing new data and our best score is **44.54%** from Trankit Five Merged. The altered punctuation yields a 50.91% score on same model. The EvaLatin2024 (Sprugnoli et al., 2024) results reach 77.41% for prose and 75.75% poetry. The task performance is not comparable since for the shared task included the use of train and dev datasets and had only the dependency parsing task. Straka et al. (2024) report leveraging the GS morphological annotation as an additional input for the parser.

5.7 Voting results

The results of the voting experiment are reported in Table 5, giving the baseline scores for the three parsers (Trankit, Stanza, custom tagger), and the majority voting results. In addition to this, we also report *Oracle* score to illustrate the theoretical upper bound for voting when it is based on these three parsing models, i.e. the accuracy of a hypothetical voting system that is always able to select the best option among the predictions. Based on the results by Zeman and Žabokrtský (2005) we expected a possible an increase of roughly two percentage points. The improvement of the voting results is reported in 5 and ranges from 0.00% to +0.89 for POS tagging and for FEATS from +0.09% to +3.15%.

6 Conclusion and future studies

The task of full morpho-syntactic parsing across the five largest established treebanks comprises 30 subtasks, of which 8 are best performed by the Trankit Five Merged model. This model demonstrates particular strength in part-of-speech labeling. Additionally, Stanza's lemmatization capabilities are noteworthy, consistently achieving the highest numerical values across all five treebanks.

Overall it can be stated that merging the available five Latin UD datasets is very beneficial especially when it comes to smaller treebanks and out of domain parsing. With our experiments, by using thematically compiled and everything merged datasets, we are able to set a new state of the art for many morpho-syntactic parsing tasks. The average improvement of our final results are reported in Table 9. Our initial results of morphological features are even further improved by using the FEATS atomic voting system especially on the smaller treebanks. The results reaching +3.15 %-points.

Future studies should first focus on addressing the issues related to the treatment of multi-word tokens. One approach could involve ensuring that the five established treebanks strictly adhere to current guidelines, such as avoiding the splitting of enclitics (e.g., *-que* 'and') into separate tokens. Additionally, the introduced voting system could be further refined and applied to a gold-standard pre-tokenized input, followed by a detailed numerical error analysis and close reading. This enables determining the specific morphological annotation tasks that our current models succeed upon. Such analysis could also determine whether observed errors

	ITTB		LLCT		Perseus		PROIEL		UDante		CIRCSE	
	UPOS	UFeat	UPOS	UFeat	UPOS	UFeat	UPOS	UFeat	UPOS	UFeat	UPOS	UFeat
Trankit	99.07	97.55	99.63	97.15	91.83	80.94	97.30	92.96	91.46	84.42	83.21	57.76
Stanza	98.64	96.15	99.61	96.96	90.81	82.03	97.14	92.18	89.85	80.92	84.47	56.85
Custom	98.72	96.61	99.14	95.67	89.58	82.58	96.44	91.64	89.91	82.24	79.72	55.29
Majority	99.07	97.64	99.64	97.32	92.72	85.73	97.78	93.98	91.73	85.25	85.25	60.23
Change	+0.00	+0.09	+0.01	+0.17	+0.89	+3.15	+0.48	+1.02	+0.27	+0.83	+0.78	+2.47
Oracle	99.60	99.01	99.82	98.46	96.11	92.64	98.83	96.98	94.19	90.69	90.22	65.31

Table 5: Results of the majority voting system compared to the three individual models used in voting. *Oracle* stands for a theoretical upper bound for voting of always selecting the best option among the predictions.

suggest the need for further harmonization of the treebanks themselves or are these cases difficult to grammatically analyze as such?

On one hand, many tasks are successfully accomplished using a single treebank for training, development, and testing, as demonstrated by the ITTB data, which does not require the inclusion of additional treebanks for improving performance. This highlights the importance of incorporating new genres across a broad time span into the UD Latin treebank family, ensuring that the training data is sufficiently diverse, comprehensive and large enough. While the development of novel gold-standard annotated datasets offers significant benefits, it is also highly demanding in terms of human resources. We hope that our high-performing models will facilitate the annotation of these datasets by providing accurate predictions that serve as a strong starting point for manual corrections, thereby easing the process.

On the other hand, one of the conclusions drawn from our diverse merged training sets is that the notion of "Latin is Latin" does not hold true. It is well established that medieval Latin is distinctly different from Classical Latin. In practical terms most scholars often identify themselves as experts in one or the other. However, a possible future study could investigate the specific attributes in a treebank's training data that make a parser model particularly adept at Classical or medieval Latin.

Another conclusion from our experiments is that the accuracy of parsing Latin from the Classical period (broadly defined) is diminished when the model is exposed to medieval training data. This warrants further exploration to define the characteristics that distinguish the two and will shed more light into computational historical linguistics. One study could be the evolution of medieval Latin and the extent to which medieval treebanks reflect preserving features of Classical Latin, analyzed by auctor and decade. It might reveal how well and what ways medieval writers were competent in Classical Latin. Another potential research direction is to investigate why parsing the UDante treebank appears less selective, with all five merged models performing well. This raises the question of whether users of Latin from this late medieval period were equally accustomed and influenced by reading both Classical and medieval authors. Alternatively, this phenomenon might be explained by the size of the training data, where additional examples contribute to improved results, as our LASLA experiment in the CIRCSE test set show.

7 Limitations

Firstly, the harmonization of UD Latin syntactic annotation (Gamba and Zeman, 2023b) and morphological annotation (Gamba and Zeman, 2023a) has been taken as a given and we have not subjected the annotations to any closer examination. As suggested by the case study sample finding of *necesse-esse* 'necessarily existent' (as discussed in the Section 5.2) the training datasets might include seldom errors from automatic processing. Secondly, the data in the LASLA corpus[14] has not been examined for any potential divergences from the UD framework. We don't inspect the results from the reserved test set we have set aside for possible further experiments on the LASLA corpus based model with our custom model. This would need more resources and we leave this for the future, since our focus only on one experiment of the CIRCSE test set.

Acknowledgements

The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources as well as The Emil Aaltonen Foundation for grant "Exploring linguistic variation in medieval Latin using computational methods" for Hanna-Mari Kupari 2022-2024.

[14] https://github.com/CIRCSE/LASLA/tree/main

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

David Bamman and Patrick J. Burns. 2020. Latin bert: A contextual language model for classical philology. *Preprint*, arXiv:2009.10053.

David Bamman and Gregory Crane. 2011. The ancient greek and latin dependency treebanks. In *Language Technology for Cultural Heritage*, pages 79–98, Berlin, Heidelberg. Springer Berlin Heidelberg.

Flavio M Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020a. Udante: First steps towards the universal dependencies treebank of dante's latin works. Accademia University Press.

Flavio Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020b. A new Latin treebank for Universal Dependencies: Charters between Ancient Latin and Romance languages. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 933–942, Marseille, France. European Language Resources Association.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations (ICLR)*.

Federica Gamba and Daniel Zeman. 2023a. Latin morphology through the centuries: Ensuring consistency for better language processing. In *Proceedings of the Ancient Language Processing Workshop*, pages 59–67, Varna, Bulgaria. INCOMA.

Federica Gamba and Daniel Zeman. 2023b. Universalising latin universal dependencies: a harmonisation of latin treebanks in ud. In *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 7–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dag Trygve Truslew Haug and Marius L. Jøhndal. 2008. Creating a parallel treebank of the old indo-european bibletranslations.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Sebastian Nehrdich and Oliver Hellwig. 2022. Accurate dependency parsing and tagging of Latin. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 20–25, Marseille, France. European Language Resources Association.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Marco Passarotti. 2019. *The Project of the Index Thomisticus Treebank*, pages 299–320. De Gruyter Saur, Berlin, Boston.

Marco Passarotti and Felice Dell'Orletta. 2010. Improvements in parsing the index Thomisticus treebank. revision, combination and a feature model for medieval Latin. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Sampo Pyysalo, Jenna Kanerva, Antti Virtanen, and Filip Ginter. 2021. WikiBERT models: Deep transfer learning for many languages. In *Proceedings* *of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 1–10, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv.org*.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Philipp Roelli. 2014. The corpus corporum, a new open latin text repository and tool. *Archivum Latinitatis Medii Aevi*, 72(1):289–304.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Rachele Sprugnoli, Federica Iurescia, and Marco Passarotti. 2024. Overview of the EvaLatin 2024 evaluation campaign. In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) @ LREC-COLING-2024*, pages 190–197, Torino, Italia. ELRA and ICCL.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, Margherita Fantoli, and Giovanni Moretti. 2022. Overview of the EvaLatin 2022 evaluation campaign. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 183–188, Marseille, France. European Language Resources Association.

Manuel Stoeckel, Alexander Henlein, Wahed Hemati, and Alexander Mehler. 2020. Voting for POS tagging of Latin texts: Using the flair of FLAIR to better ensemble classifiers by example of Latin. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 130–135, Marseille, France. European Language Resources Association (ELRA).

Milan Straka, Jana Straková, and Federica Gamba. 2024. úfal latinpipe at evalatin 2024: Morphosyntactic analysis of latin. *Preprint*, arXiv:2404.05839.

Phillip Benjamin Ströbel. 2022. Roberta base latin cased v2.

Elina Stüssi and Phillip Ströbel. 2024. Part-of-speech tagging of 16th-century Latin with GPT. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 196–206, St. Julians, Malta. Association for Computational Linguistics.

Daniel Zeman and Zdeněk Žabokrtský. 2005. Improving parsing accuracy by combining diverse dependency parsers. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 171–178, Vancouver, British Columbia. Association for Computational Linguistics.

A Appendix

CIRCSE test set results	Tasks:					
Model Name	Tokens	UPOS	UFeats	Lemmas	UAS	LAS
Stanza PROIEL	100.00	84.46	59.48	72.37	48.18	41.38
Trankit PROIEL	99.24	81.50	55.39	60.08	49.44	41.92
Custom Perseus	-	76.29	47.79	-	-	-
Custom PROIEL	-	79.72	55.29	-	-	-
Custom Five Merged	-	81.30	57.11	-	-	-
Custom Classical	-	80.84	56.53	-	-	-
LASLA	-	**91.17**	**70.52**	-	-	-
Stanza Classical	100.00	84.37	56.79	73.36	49.64	43.03
Stanza Five Merged	99.98	82.56	51.23	78.00	47.00	40.14
Trankit Classical	99.71	83.08	57.09	62.87	50.57	43.06
Trankit Five Merged	99.82	83.21	57.76	68.15	51.29	44.54

Table 6: The results of the CIRCSE test set. For models trained on individual treebank data only the results for PROIEL are given for all models, since both Stanza and Trankit Perseus models failed to run because of severe sentence segmentation issues.

CIRCSE altered test set	Tasks					
Automatically added punctuation	**Tokens**	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**
Stanza ITTB	*99.98*	*81.64*	*56.32*	*73.32*	*50.49*	*41.53*
Stanza LLCT	*99.99*	*75.41*	*40.54*	*56.13*	*37.24*	*25.18*
Stanza PROIEL	***100.00***	*79.98*	***62.06***	*74.20*	*46.17*	*38.59*
Stanza Perseus	*99.93*	*83.96*	*57.26*	*70.16*	*46.75*	*38.43*
Stanza Classical	*100.00*	*85.81*	*59.54*	*75.46*	*54.20*	*46.93*
Stanza Five Merged	*100.00*	*83.94*	*54.33*	***79.58***	*53.89*	*46.60*
Trankit Classical	*99.78*	*85.21*	*59.75*	*65.44*	*56.61*	*48.43*
Trankit Late and Christian	*99.80*	*84.85*	*58.20*	*66.69*	*54.53*	*45.41*
Trankit Late and Medieval	*99.74*	*82.68*	*55.35*	*63.18*	*51.99*	*42.52*
Trankit Five Merged	*99.79*	***87.05***	*61.39*	*71.73*	***59.16***	***50.91***

Table 7: The effects to the performance of the different models with the added punctuation to the CIRCSE gold standard test set. The results are not comparable to the UD released test set and given in italics.

Effects of merged treebanks in training	**Tasks:**					
Treebank and model	**Tokens**	**UPOS**	**UFeats**	**Lemmas**	**UAS**	**LAS**
ITTB						
Custom tagger	-	98.66	96.50	-	-	-
Improvement from Late and Christian	-	**0.06**	**0.11**	-	-	-
LLCT						
Trankit	99.99	99.66	97.36	96.50	96.15	95.37
Improvement from Late and Medieval	**0.00**	**0.00**	**-0.18**	**0.19**	**0.31**	**0.14**
Perseus						
Stanza	99.94	89.44	80.17	80.97	69.75	61.93
Improvement from Classical	**0.02**	**0.65**	**0.96**	**4.92**	**5.53**	**6.36**
Trankit	99.46	88.90	77.98	63.99	74.08	66.97
Improvement from Classical	**0.28**	**1.60**	**5.27**	**10.61**	**3.81**	**4.31**
Custom tagger	-	86.29	76.17	-	-	-
Improvement from Classical	**-**	**3.29**	**6.41**	**-**	**-**	**-**
PROIEL						
Custom tagger	-	96.42	91.26	-	-	-
Improvement from Five Merged	**-**	**0.02**	**0.38**	**-**	**-**	**-**
UDante						
Trankit	99.50	91.17	80.71	73.89	75.92	68.65
Improvement from Five Merged	**0.16**	**0.29**	**3.71**	**3.61**	**3.71**	**4.77**
Custom tagger	-	87.43	75.84	-	-	-
Improvement from Five Merged	-	**2.48**	**6.40**	-	-	-
Average improvement	**0.15**	**1.19**	**2.78**	**4.84**	**3.34**	**3.90**

Table 8: The most important results of the merging of diverse training data.

Tasks:	POS		UFEATS		UAS		LAS	
Treebank	Our highest	Change	Our highest	Change	Our highest	Change	Our highest	Change
ITTB	99.07	4.19	**97.64**	1.49	92.09	-0.19	90.71	2.42
LLCT	99.66	5.16	97.36	0.55	96.46	0.38	95.51	0.60
PERSEUS	**92.72**	8.41	**85.73**	7.87	77.89	8.92	71.28	11.85
PROIEL	**97.78**	7.78	**93.98**	1.26	84.09	-0.82	80.97	-0.28
UDante	**91.73**	5.93	**85.25**	5.95	79.63	12.84	80.97	16.10
Average change		6.29		3.42		4.23		6.14

Table 9: Summary of our best F1 scores. The ones produced by the voting system are given in a bold typeset. The change as percentage points to the most recent POS tagging study by Stüssi and Ströbel (2024). For **ITTB** the best score **99.07%** is predicted by Trankit Five Merged (in experimenting with a GPT model on POS tagging the best results reported by Stüssi and Ströbel (2024) is 94.88 produced on GPT-4 train1000). The same applies for **Perseus** as well **91.83%** (84.31 on GPT-4 train2000), **PROIEL** at **97.30%** (90.00 on GPT-4 train5000) and **UDante 91.46%** (85.8 on GPT-4 train200). For **LLCT** the best score **99.66%** (94.5 on GPT-4 train1000) is produced by the Trankit individually trained model. For UAS and LAS the results are compared to best numbers reported by Gamba and Zeman (2023a). They have accomplished this using jackknifing technique. In this the training data is divided into n parts, where n−1 parts are used to train a model to annotate the remaining nth part. When rotating this n times, we receive a version of the whole training data with predicted annotations, which can be used during final model training. Therefore, the final model is trained using predicted annotations, in this case the dependency parsing model is trained using predicted morphology and lemmas.