ÚFAL LatinPipe at EvaLatin 2024: Morphosyntactic Analysis of Latin

Milan Straka, Jana Straková, Federica Gamba
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Czech Republic
{straka,strakova,gamba}@ufal.mff.cuni.cz

Abstract

We present LatinPipe, the winning submission to the EvaLatin 2024 Dependency Parsing shared task. Our system consists of a fine-tuned concatenation of base and large pre-trained LMs, with a dot-product attention head for parsing and softmax classification heads for morphology to jointly learn both dependency parsing and morphological analysis. It is trained by sampling from seven publicly available Latin corpora, utilizing additional harmonization of annotations to achieve a more unified annotation style. Before fine-tuning, we train the system for a few initial epochs with frozen weights. We also add additional local relative contextualization by stacking the BiLSTM layers on top of the Transformer(s). Finally, we ensemble output probability distributions from seven randomly instantiated networks for the final submission. The code is available at https://github.com/ufal/evalatin2024-latinpipe.

Keywords: dependency parsing, part of speech tagging, EvaLatin, Latin, LatinPipe

1. Introduction

In this paper, we describe our entry to the EvaLatin 2024 Dependency Parsing shared task (Sprugnoli et al., 2024). Our system is called LatinPipe to resemble its predecessors, UDPipe (Straka and Straková, 2017) and UDPipe 2 (Straka, 2018). We submitted two variants, called *ÚFAL LatinPipe 1* and *ÚFAL LatinPipe 2*, placing 1st and 2nd in the shared task evaluation, respectively.

Our system is an evolution of UDPipe 2 (Straka, 2018). LatinPipe is a graph-based dependency parser which uses a deep neural network for scoring the graph edges. Unlike UDPipe 2, the neural network architecture of LatinPipe is a fine-tuned pre-trained language model, with a dot-product attention head for dependency parsing and softmax classification heads for morphological analysis to learn both these tasks jointly.

We provide an extensive evaluation of the approaches used in LatinPipe: a comparison of monolingual and multilingual pre-trained language models and their concatenations; initial pretraining on the frozen Transformer weights; adding two BiLSTM layers on top of the Transformers; and using the gold UPOS from the shared task data on the network input. A considerable focus is directed at multi-treebank training, as well as the harmonization of annotation styles among the seven publicly available Latin treebanks.

2. Related Work

The EvaLatin 2024 Dependency Parsing shared task (Sprugnoli et al., 2024) builds upon the two previous editions of EvaLatin, which focused respectively on lemmatization and POS tagging (Sprugnoli et al., 2020) and lemmatization, POS tagging, and features identification (Sprugnoli et al., 2022). UDPipe 2 won the EvaLatin 2020 shared task (Straka and Straková, 2020); previously, it participated in the 2018 CoNLL Shared Tasks on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018), which encompassed also Latin, and placed among the winning systems (Straka, 2018).

Latin Dependency Parsing In recent years, Nehrdich and Hellwig (2022) developed a graph-based dependency parser specifically for Latin. Their approach modifies the architecture of the biaffine parser proposed by Dozat and Manning (2017) by incorporating a character-based convolutional neural network (CharCNN), and exploits Latin BERT embeddings (Bamman and Burns, 2020).

Fantoli and de Lhoneux (2022) trained a POS tagging and parsing model using the deep biaffine parser (Dozat and Manning, 2017) implementation of MaChAmp (van der Goot et al., 2021) and exploiting treebank embeddings in the encoder.

Karamolegkou and Stymne (2021) explored Latin parsing in a low-resource scenario and found ancient Greek to be most effective as transfer language, likely due to its syntactic similarity with Latin.

3. Data

Latin Treebanks We train LatinPipe on the training portions of the following seven publicly available Latin corpora:

- ITTB of UD 2.13 (Passarotti, 2019);
- LLCT of UD 2.13 (Cecchini et al., 2020a);

Corpus	Training tokens
ITTB	391K
LLCT	194K
PROIEL	178K
UDante	31K
Perseus	18K
Sab	11K
Arch	1K
UD 2.13	812K
UD 2.13+Sab+Arch	824K

Table 1: Training data sizes in tokens.

- PROIEL in either of these two versions: UD 2.13 (Haug and Jøhndal, 2008), and a UD-style harmonized version (Gamba and Zeman, 2023a,b);[1]
- UDante of UD 2.13 (Cecchini et al., 2020b);
- Perseus of UD 2.13 (Bamman and Crane, 2011);
- UD-style annotated text of *De Latinae Linguae Reparatione* by Marcus Antonius Sabellicus (Gamba and Cecchini, 2024);
- *Archimedes Latinus* UD-style treebank (Fantoli and de Lhoneux, 2022), based on the Latin translation of the Greek mathematical work *The Spirals* of Archimedes;[2]

where UD 2.13 stands for the Universal Dependencies project (Nivre et al., 2020), version 2.13 (Zeman et al., 2023). We denote the former five corpora distributed by UD 2.13 as *UD 2.13* and all seven corpora including additionally *Arch* and *Sab* as *UD 2.13+Arch+Sab* in our experiments. The treebank training data sizes are presented in Table 1.

For the shared task, we train in multi-treebank setting, in which the examples from the abovementioned corpora are sampled into training batches proportionally to the square root of the number of their sentences, similarly to van der Goot et al. (2021).

Harmonization of Annotation Styles We noticed that the PROIEL treebank stands out most in terms of annotation style from the rest of the other treebanks, so much so that the differences in annotation style result in varying performance. We therefore experimented with the following three settings:

- training with a harmonized version of PROIEL by Gamba and Zeman (2023a,b), submitted as *ÚFAL LatinPipe 1*;
- training without PROIEL altogether, submitted as *ÚFAL LatinPipe 2*;
- training with the original PROIEL annotation by Haug and Jøhndal (2008), not submitted due to the two-runs-per-team limit.

The harmonized version of PROIEL resulted from the harmonization carried out by Gamba and Zeman (2023a,b), who observed persisting differences in the annotation scheme of the five Latin treebanks, annotated by different teams and in different stages of the development of UD guidelines. Divergences were observed at all annotation levels, from word segmentation to lemmatization, POS tags, morphology, and syntactic relations. The implemented harmonization process led to substantial improvements in parsing performances, confirming the need for a truly standardized annotation style. Notably, among the five treebanks, in the case of PROIEL a lower degree of accordance with the UD guidelines was observed. For instance, in compound numerals like *viginti quattuor* 'twenty-four' the second number is attached to the first one through a `fixed` relation; in the harmonized version, such dependencies are reannotated as `flat`. Moreover, PROIEL makes use of the `dep` relation, intended for cases where a more precise deprel cannot be assigned. Through POS tags and morphology, in the harmonized version `dep` is replaced with a more appropriate one.

4. Methods

LatinPipe is a graph-based dependency parser. First, a deep learning neural network is used to score the graph edge values, and then a global optimization Chu-Liu/Edmonds' algorithm (Chu and Liu, 1965; Edmonds, 1967) for finding the minimum spanning tree problem is run on the graph.

For scoring the graph edge values, LatinPipe pursues a deep learning approach and consists of a fine-tuned pre-trained LM (or a concatenation of them) with a dot-product parsing attention head. In addition, morphology softmax classification heads are also used, so LatinPipe jointly learns both dependency parsing and morphological analysis.

The general overview of the architecture is given in Figure 1 and the details are outlined in the following paragraphs.

Pre-trained LMs Our baselines are either fine-tuned LaBerta or PhilBerta, the Latin monolingual RoBERTa base language models by Riemenschneider and Frank (2023); or the fine-tuned XLM-RoBERTa large (Conneau et al. (2020); 355M parameters), which was pretrained on 390M

[1] Available for download at https://github.com/fjambe/Latin-variability/tree/main/morpho_harmonization/morpho-harmonized-treebanks.

[2] Available at https://github.com/mfantoli/ArchimedesLatinus.

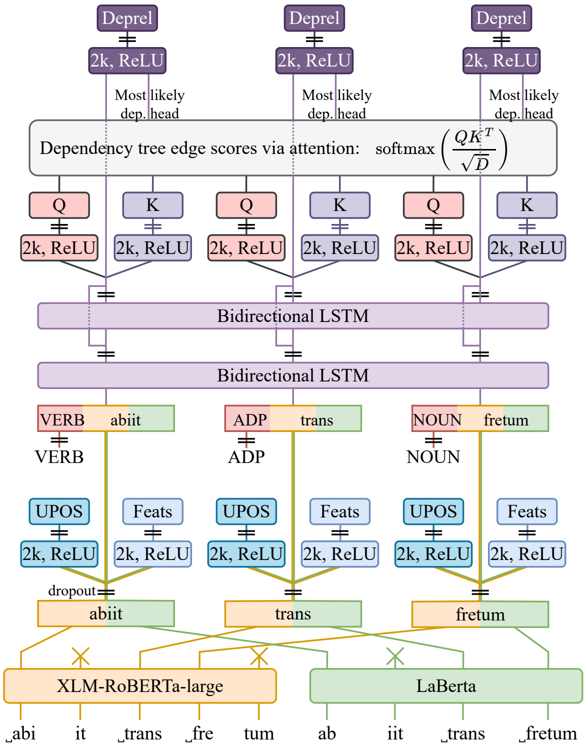

Figure 1: LatinPipe architecture overview.

Latin tokens among other languages. Apart from using the single fine-tuned PLMs, we also experimented with a concatenation of the contextualized embeddings yielded by multiple fine-tuned encoders: *LaBerta+PhilBerta* and *XLM-R large+LaBerta+PhilBerta*.

Frozen Pretraining Before fine-tuning the PLMs' weights, we can optionally freeze the pre-trained Transformer weights, and optimize solely the remaining weights of the architecture for a few initial epochs, namely the heads and the stacked BiLSTM layers. The objective of frozen pretraining is to facilitate the commencement of the fine-tuning optimization from a favorable starting point.

Adding LSTMs We incorporate two bidirectional LSTM layers (Hochreiter and Schmidhuber, 1997; Gers et al., 2000) on top of the Transformer(s) to enhance the modeling of relative short-distance relationships between the tokens and to contextualize the embedded UPOS tags.

Gold UPOS on Input We leverage the gold morphological analysis provided in the shared task data as an additional input to the neural network. The trainable word embeddings of UPOS are concatenated with the contextualized embeddings yielded from the fine-tuned PLM(s), and together, the concatenation of embeddings is processed by the LSTM layers.

Ensembling For the final submission, we ensemble output probability distributions from seven randomly instantiated networks by averaging the probabilities in the corresponding dimensions.

Handling punctuation The shared task test data do not contain punctuation. This causes concern in settings when training without PROIEL, which is the only representative of a treebank without punctuation. Training solely on data containing punctuation is expected to lead to inferior performance on test data without it. Therefore in this particular setting, we artificially add punctuation to the test data by appending periods at sentence ends, and after the model prediction, we remove the dummy punctuation again.[3]

Architecture Details In the LatinPipe architecture (Figure 1), every classification layer and computation of queries and keys is preceded by a hidden layer of size 2 048 with ReLU activation. The dimensionality of the queries and keys is 512, and the LSTM dimensionality is 256. When predicting dependency relations, we also concatenate the LSTM-generated representation of the most likely dependency head according to the predicted scores (which is not necessarily the one chosen by the Chu-Liu/Edmonds' algorithm).

Training Details The model is trained with the Adam optimizer (Kingma and Ba, 2015) for 30 epochs, each comprising 1 000 batches with a batch size of 32. The learning rate is first linearly increased from 0 to 2e-5 in the first two epochs and then decays to 0 according to the cosine schedule. Optionally, we perform 10-epoch pretraining with frozen Transformer weights utilizing a constant learning rate of 1e-3. On a single A100 GPU with 40GB, the training takes 9 hours. The exact training configuration, including the exact command used to train the models, is available in the released source code.

5. Results

We present the evaluation on the UD 2.13 test data in Tables 2 and on the EvaLatin 2024 test data in Table 3. All the results are averages of three runs.

Table 2.A evaluates the baseline fine-tuned PLMs on the UD 2.13 test sets. Increasing PLM size from base to large clearly improves the results across the board and on average, even if the large model is not a monolingual but a multilingual one.

[3] Obviously, the other option would be to remove the punctuation from the training data and retrain the models, an expensive and unavailable option due to the restricted time span of the shared task testing period.

Experiment	Avg	ITTB	LLCT	PROIEL	UDante	Perseus
A) PLMs Evaluation						
LaBerta	83.20	90.91	94.54	86.75	66.71	77.08
PhilBerta	82.87	91.09	94.19	86.13	66.42	76.51
LaBerta+PhilBerta	83.99	91.31	94.74	87.29	68.18	**78.42**
XLM-R large	84.19	91.60	95.33	87.18	71.17	75.67
XLM-R large+LaBerta+PhilBerta	**84.67**	**91.78**	**95.35**	**87.57**	**71.95**	76.70
B) Incremental Architecture Improvements w.r.t. the Previous Row						
+ Frozen training for 10 epochs	86.09	92.29	95.34	88.64	74.20	79.98
+ Two bidirectional LSTM layers	86.33	92.81	94.70	89.05	74.78	80.32
+ Gold UPOS on parser input	**86.97**	**93.18**	**95.64**	**89.78**	**74.99**	**81.28**
C) Multi-treebank Training w.r.t. the Previous Row						
Single-treebank training	86.97	**93.18**	**95.64**	**89.78**	74.99	81.28
UD 2.13 training	88.05	92.25	95.60	88.74	79.84	**83.84**
UD 2.13+Sab+Arch training	**88.09**	92.18	95.44	88.43	**80.56**	83.81
D) Ensembles of the Models in the Previous Section						
Single-treebank training, 7 models	87.31	**93.38**	95.78	**90.23**	75.51	81.66
UD 2.13 training, 7 models	88.51	92.65	**95.89**	89.10	80.91	84.02
UD 2.13+Sab+Arch training, 7 models	**88.63**	92.45	95.78	89.23	**81.47**	**84.22**
E) Previous work						
UDPipe 2 (Straka, 2018), UD 2.12		*89.35*	*94.39*	*79.55*	*68.65*	*71.91*
MaChAmp (van der Goot et al., 2021), UD 2.8		*92.45*	*95.41*	*86.97*	*74.01*	*74.67*
Nehrdich and Hellwig (2022), UD 2.8-2.9		*92.99*	—	*86.34*	—	*80.16*

Table 2: UD 2.13 test sets LAS evaluation. Avg denotes the LAS macro average over the UD 2.13 corpora. Section E shows previous work on older UD versions.

Experiment	Avg	Poetry	Prose
A) Single-treebank Training			
ITTB	59.96	57.84	62.08
LLCT	47.93	45.12	50.74
PROIEL original	68.87	68.47	69.26
PROIEL harmonized	**73.88**	**72.37**	**75.40**
UDante	60.23	59.11	61.36
Perseus	59.22	58.43	60.02
B) Multi-treebank with PROIEL Versions			
UD 2.13, original	72.31	72.10	72.52
UD 2.13, none	66.16	64.03	68.29
UD 2.13, harmonized	75.22	**74.65**	75.78
UD 2.13+Sab+Arch, original	72.75	72.35	73.14
UD 2.13+Sab+Arch, none	66.64	64.50	68.79
UD 2.13+Sab+Arch, harmo.	**75.48**	74.52	**76.43**
C) Multi-treebank w/ and wo/ Gold UPOS			
w/ gold UPOS	**75.48**	**74.52**	**76.43**
wo/ gold UPOS	74.19	73.28	75.09
D) Ensembles of 7 Models			
UD 2.13+Sab+Arch, original	73.76	73.57	73.95
UD 2.13+Sab+Arch, none	68.16	65.71	70.60
UD 2.13+Sab+Arch, harmo.	**76.58**	**75.75**	**77.41**
E) Adding Punctuation Before Prediction			
UD 2.13+Sab+Arch, none	71.87	70.68	73.07

Table 3: EvaLatin 2024 test set LAS evaluation. Avg denotes the LAS macro average over Poetry and Prose.

The only exception is Perseus, on which we suspect the XLM-R large to overtrain due to the small size of the corpus (see Table 1). Finally, a concatenation of models yields further gains over their single components in all cases.

Table 2.B shows a notable macro average gain of +1.42 percent points when pretraining with frozen weights for initial 10 epochs before fine-tuning. Also the addition of the two bidirectional LSTM layers helps marginally on average by +0.24. Unsurprisingly, the addition of gold UPOS on input brings +0.64 percent points in the UD 2.13 macro average, as well as it improves performance in all single UD 2.13 treebanks. On the EvaLatin test set, the addition of the gold UPOS straightforwardly improved the results by +1.2 on Poetry and +1.3 on Prose, as measured on the non-ensembled model (Table 3.C).

Table 2.C compares multi-treebank training vs. single-treebank training. In accord with previous literature (Nehrdich and Hellwig, 2022), we observed the greatest benefits from the multi-treebank training for the smaller datasets (UDante and Perseus), indecisive results for the middle-sized datasets (LLCT and PROIEL), and a decrease for the largest dataset (ITTB). However, in macro average, we gained +0.51 percent point by multi-treebank training. While the addition of the two new small datasets, the Sab and Arch, is indecisive on the

Experiment	Avg	ITTB	LLCT	PROIEL	UDante	Perseus
A) Best Single-model Results						
Single-treebank training	**97.33**	**99.37**	**99.77**	**98.32**	**93.61**	95.55
UD 2.13 training	97.23	99.25	**99.77**	98.10	93.18	**95.85**
B) Best 7-Model Ensemble Results						
Single-treebank training, 7 models	**97.43**	**99.39**	99.78	**98.47**	**93.61**	95.89
UD 2.13 training, 7 models	97.42	99.33	**99.79**	98.31	93.58	**96.09**
C) Previous work						
UDPipe 2 (Straka, 2018), UD 2.12		*99.03*	*99.75*	*97.02*	*92.95*	*91.18*
MaChAmp (van der Goot et al., 2021), UD 2.8		*98.62*	*99.68*	*97.84*	*91.44*	*90.46*
Nehrdich and Hellwig (2022), UD 2.8-2.9		*97.3*	—	*94.2*	—	*90.8*
Bamman and Burns (2020), UD 2.6		*98.8*	—	*98.2*	—	*94.3*

Table 4: UD 2.13 test sets UPOS evaluation, with Avg denoting the UPOS macro average.

Experiment	Avg	ITTB	LLCT	PROIEL	UDante	Perseus
A) Best Single-model Results						
Single-treebank training	92.45	**98.57**	97.33	**94.68**	83.06	88.61
UD 2.13 training	**93.68**	98.26	**97.36**	94.05	**88.27**	**90.49**
B) Best 7-Model Ensemble Results						
Single-treebank training, 7 models	92.68	**98.62**	97.42	**95.04**	83.37	88.94
UD 2.13 training, 7 models	**94.19**	98.45	**97.52**	94.56	**89.16**	**91.24**
C) Previous work						
UDPipe 2 (Straka, 2018), UD 2.12		*97.12*	*97.16*	*91.43*	*84.38*	*84.65*
MaChAmp (van der Goot et al., 2021), UD 2.8		*96.95*	*96.79*	*92.56*	*69.72*	*84.32*

Table 5: UD 2.13 test sets UFeats evaluation, with Avg denoting the UFeats macro average.

UD 2.13 macro average in Table 2.C, which is in alignment with their modest size (Table 1), on EvaLatin 2024 (Table 3.B), we observed a marginal improvement when incorporating Sab and Arch, which might probably be attributed to similarity of the EvaLatin test data to these treebanks.

Table 3 shows the evaluation on the EvaLatin test data, both Poetry and Prose, and their LAS macro average; with focus on the effect of data harmonization. In all paired experiments, the harmonized PROIEL version clearly improved results over the version with the original PROIEL dataset from UD 2.13, when evaluated on the EvaLatin 2024 test data. However, using at least the original PROIEL dataset in the multi-treebank training is still better than excluding the PROIEL treebank altogether.

As evidenced by both Table 2.D and Table 3.D, an ensemble is always stronger than its individual components. Ensembling adds on average +0.45 percent points on the UD 2.13 LAS macro average over three experimental settings (compare sections C and D in Table 2). In the shared task, ensembling adds +1.26 percent points (compare sections B and D in Table 3). Our best entry, submitted as *ÚFAL LatinPipe 1*, corresponds to the row *UD 2.13+Sab+Arch, harmo.* in Table 3.D.

Finally, when training without PROIEL in a multi-treebank setting, we have to mitigate the punctuation mismatch between the training and the shared task test data, as described in Section 4. Row *UD 2.13+Sab+Arch* in Table 3.E shows our second submission to the shared task, *ÚFAL LatinPipe2*, in which we corrected for missing punctuation in the shared task test data.

UPOS and UFeats Tagging Since our model performs full morphosyntactic analysis, we present also the accuracy of UPOS tagging and UFeats tagging in Tables 4 and 5, respectively. LatinPipe surpasses the previous systems and sets new state-of-the-art results for all treebanks.

6. Conclusion

We described LatinPipe, the winning entry to the EvaLatin 2024 Dependency Parsing shared task, and we provided the evaluation and rationale behind our system design choices. The source code for LatinPipe is available at `https://github.com/ufal/evalatin2024-latinpipe`. Our future work will entail drawing insights from the methodologies presented in this context for the development of UDPipe 3.

7. Acknowledgements

This work has been supported by the Grant Agency of the Czech Republic under the EXPRO program as project “LUSyD” (project No. GX20-16819X), and by the Grant Agency of Charles University as project GAUK No. 104924 “Adapting Uniform Meaning Representation (UMR) for the Italic/Romance languages”. The work described herein uses resources hosted by the LINDAT/CLARIAH-CZ Research Infrastructure (projects LM2018101 and LM2023062, supported by the Ministry of Education, Youth and Sports of the Czech Republic).

8. Bibliographical References

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology. *CoRR*, abs/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 Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020a. 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.

Flavio Massimiliano Cecchini, Rachele Sprugnoli, Giovanni Moretti, and Marco Passarotti. 2020b. UDante: First Steps Towards the Universal Dependencies Treebank of Dante’s Latin Works. In *Proceedings of the Seventh Italian Conference on Computational Linguistics*, pages 1–7. Italian Association for Computational Linguistics (AILC).

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14(10):1396–1400.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking Embedding Coupling in Pre-trained Language Models. In *International Conference on Learning Representations*.

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.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *5th International Conference on Learning Representations*, pages 1–8.

Jack Edmonds. 1967. Optimum Branchings. *Journal of Research of the National Bureau of Standards, B*, 71(4):233–240.

Margherita Fantoli and Miryam de Lhoneux. 2022. Linguistic Annotation of Neo-Latin Mathematical Texts: A Pilot-Study to Improve the Automatic Parsing of the Archimedes Latinus". In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 129–134, Marseille, France. European Language Resources Association.

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 Ltd., Shoumen, Bulgaria.

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, Washington, D.C. Association for Computational Linguistics.

Felix Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural computation*, 12(10):2451–2471.

Dag Trygve Truslew Haug and Marius Jøhndal. 2008. Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Antonia Karamolegkou and Sara Stymne. 2021. Investigation of Transfer Languages for Parsing Latin: Italic Branch vs. Hellenic Branch. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 315–320, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Marco Passarotti. 2019. The Project of the Index Thomisticus Treebank. *Digital Classical Philology*, 10:299–320.

Frederick Riemenschneider and Anette Frank. 2023. Exploring Large Language Models for Classical Philology. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto, Canada. Association for Computational Linguistics.

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 2024*, Torino, Italy. European Language Resources Association.

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.

Rachele Sprugnoli, Marco Passarotti, Flavio Massimiliano Cecchini, and Matteo Pellegrini. 2020. Overview of the EvaLatin 2020 evaluation campaign. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 105–110, Marseille, France. European Language Resources Association (ELRA).

Milan Straka. 2018. UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2020. UDPipe at EvaLatin 2020: Contextualized Embeddings and Treebank Embeddings. In *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Languages*, pages 124–129, Marseille, France. European Language Resources Association (ELRA).

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive Choice, Ample Tasks (MaChAmp): A Toolkit for Multi-task Learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael

Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

9. Language Resource References

Gamba, Federica and Cecchini, Flavio Massimiliano. 2024. *De Latinae Linguae Reparatione treebank*. PID http://hdl.handle.net/11234/1-5438. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Zeman, Daniel and Nivre, Joakim and others. 2023. *Universal Dependencies 2.13*. PID http://hdl.handle.net/11234/1-5287. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.