Behr at EvaLatin 2024: Latin Dependency Parsing Using Historical Sentence Embeddings

Rufus Behr
Research Computing at Northeastern University
r.behr@northeastern.edu

Abstract

This paper identifies the system used for my submission to EvaLatin's shared dependency parsing task as part of the LT4HALA 2024 workshop. EvaLatin presented new Latin prose and poetry dependency test data from potentially different time periods, and imposed no restriction on training data or model selection for the task. This paper, therefore, sought to build a general Latin dependency parser that would perform accurately regardless of the Latin age to which the test data belongs. To train a general parser, all of the available Universal Dependencies treebanks were used, but in order to address the changes in the Latin language over time, this paper introduces historical sentence embeddings. A model was trained to encode sentences of the same Latin age into vectors of high cosine similarity, which are referred to as historical sentence embeddings. The system introduces these historical sentence embeddings into a biaffine dependency parser with the hopes of enabling training across the Latin treebanks in a more efficacious manner, but their inclusion shows no improvement over the base model.

Keywords: Natural Language Processing (NLP), Dependency Parsing, Latin

1. Introduction

EvaLatin's (Sprugnoli et al., 2024) dependency parsing task, which makes use of the Universal Dependency Parsing framework[1], permitted the use of any models and combination of training data to parse new test data created for this task, consisting of both prose and poetic texts from different time periods. One of the main challenges for this task, therefore, is identifying which combination of treebanks and data to use.

There are two main complications regarding dependency parsing data for Latin: its comparatively low-resource nature and the evolution of the language over time. Nehrdich and Hellwig (2022), citing Passarotti and Ruffolo (2010) and McGillivray and Passarotti (2009), explain that prior works on dependency parsing for Latin have domain transfer issues, where the training on one treebank yields poorer results on others. The authors explain, citing Dinkova-Bruun (2011) and Vincent (2016), that this issue stems from the linguistic evolution of the language over time, which, for instance, can be seen when comparing Classical Latin to Medieval, and this change is reflected in the respective dependency parsing treebanks for those time periods. Consequently, even though Latin is well-studied and has sizable extant text compared to other low-resource languages, the change in the language over time can make it prohibitive to use all the data that is available.

The two most widely-used forms of dependency parsing algorithms are graph-based and transition-based. In the latter, the parser moves across the sentence, adding words to a stack, and, given the top elements of the stack and its prior transitions, predicts if there's a dependency arc (Jurafsky and Martin). Graph-based parsing algorithms, however, encode a given sentence into a fully connected, weighted, and directed graph, where each vertex is a word and each edge a possible relation, and the parser then assigns scores for each edge. Afterwards, they find the maximum spanning tree for this graph, which is deemed the best parse tree (Jurafsky and Martin; Altıntaş and Tantuğ, 2023). A notable downside to transition-based parsing is that it necessarily creates a projective tree (Jurafsky and Martin), whereas graph-based parsing can produce non-projective trees. As Nehrdich and Hellwig (2022) explain, one reason graph-based parsing is preferred for Latin is the freedom of word order, resulting in possibly non-projective dependency trees.

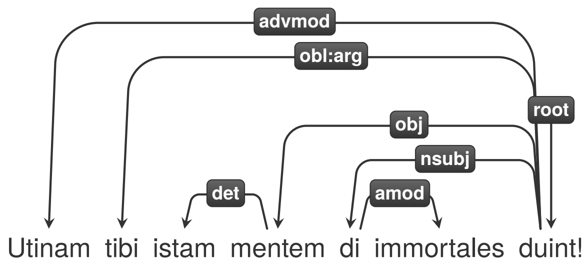

Figure 1: An example of dependency parsed Latin text from Cicero's *in Catilinam*

The predominant neural graph-based dependency parsing architecture comes from Dozat and

[1] www.universaldependencies.org

Manning (2017). This parser takes the words of the sentences and creates 100-dimensional uncased word vectors concatenated with their part-of-speech tag vectors (Dozat and Manning, 2017; Altıntaş and Tantuğ, 2023). These are then processed by three Bidirectional Long Short-Term Memory (BiLSTM) layers, the output of which is passed into four Multilayer perceptrons (MLP). Two of the MLPs are used to identify head and dependent arcs and the other two to identify their labels (Dozat and Manning, 2017; Altıntaş and Tantuğ, 2023). The vectors of the MLPs are passed into two biaffine classifiers, which produce score matrices for the dependency arcs and their label probabilities (Dozat and Manning, 2017; Altıntaş and Tantuğ, 2023).

This architecture by Dozat and Manning (2017) was the base architecture used for the Latin dependency parsing done by Nehrdich and Hellwig (2022), which achieved state-of-the-art results. The authors modified this architecture by employing contextualized Latin word embeddings from Latin BERT (Bamman and Burns, 2020).

This paper uses the dependency parser model architecture and code from Attardi et al. (2021) as its base. That model is a modified version of the semantic dependency parser proposed by Dozat and Manning (2018), which was an extension of the authors' prior work for semantic dependency parsing. The modification by Attardi et al. (2021) was in its loss function, using softmax cross-entropy rather than sigmoid.

The model in this paper builds on top of this architecture by introducing a historical sentence embedding produced by a Sentence-BERT (SBERT) model (Reimers and Gurevych, 2019), trained for this submission. The sentence embedding is concatenated with the output of the BiLSTM before being passed into the four MLPs. This embedding is introduced with the hope that the model might yield better results when trained on the Latin treebanks that span different periods in the history of Latin.

2. Model Architecture and Resources

Universal Dependencies has five Latin treebanks available: Index Thomisticus Treebank (ITTB) (Passarotti, 2019), Late Latin Charter Treebank (LLCT) (Cecchini et al., 2020b), Perseus (Bamman and Crane, 2011), UDante treebank (Cecchini et al., 2020a), and PROIEL (Haug and Jøhndal, 2008).

Gamba and Zeman (2023) experimented with a new workflow that involves harmonising all the Universal Dependency Latin treebanks before training with UDPipe and Stanza, but they found that the parsing accuracy only improved slightly after applying the harmonisation process.

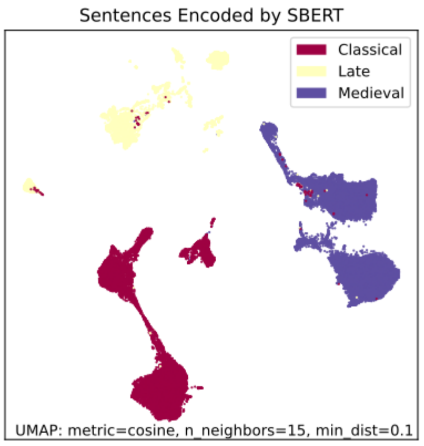

Figure 2: The data encoded using the SBERT model and then visualised with UMAP

This paper also makes use of all the available Universal Dependency Latin treebanks — in particular, the Universal Dependency version 2.13 release of ITTB (Passarotti, 2019), LLCT (Cecchini et al., 2020b), Perseus (Bamman and Crane, 2011), UDante (Cecchini et al., 2020a), and PROIEL (Haug and Jøhndal, 2008), but rather than modification of the treebanks prior to training, it introduces a historical sentence embedding. This idea is inspired by the work done by Altıntaş and Tantuğ (2023), where they show improved dependency parsing performance through concatenating features, including sentence representation, to the tokens before the MLP layer. The authors did not use SBERT themselves, but they did list it as a possible sentence representation.

Filename	Kept Sentences	Sentences Skipped
la_udante-ud-train.conllu	926	0
la_udante-ud-dev.conllu	375	1
la_udante-ud-test.conllu	419	0
la_ittb-ud-test.conllu	1879	222
la_ittb-ud-dev.conllu	1936	165
la_ittb-ud-train.conllu	21107	1668
la_llct-ud-dev.conllu	752	98
la_llct-ud-train.conllu	6189	1100
la_llct-ud-test.conllu	715	169
la_proiel-ud-train.conllu	15515	681
la_proiel-ud-test.conllu	1201	59
la_proiel-ud-dev.conllu	1171	62
la_perseus-ud-train.conllu	1324	10
la_perseus-ud-test.conllu	935	4
Total	54444	4239

Table 1: Treebank data for the experiments, loaded and displayed sequentially

In preparation for both the dependency parsing training and the SBERT training, the five treebanks were merged with exact duplicates removed. To identify these duplicates, sentences were compared against previously processed sentences, and if a new sentence's text was found previously, it was skipped in the merging process. Table 1 shows the data loaded in order during the merging process.

Age	Number of Sentences	Percentage
Classical	19192	49%
Late	8610	16%
Medieval	26642	35%

Table 2: Sentence Distribution by Age

In addition to removing duplicates, the sentences are sorted by their Latin age, the resulting distribution of which can be seen in Table 2.

2.1. Historical Sentence Embedding

As the intended purpose of including a historical sentence embedding is guiding the parser dependent on the text's corresponding Latin age to allow training on all treebanks, encoded sentences of the same age should have a higher cosine similarity, whereas sentences of other ages should be dissimilar.

As stated, this paper uses SBERT (Reimers and Gurevych, 2019), a fine-tuned version of BERT — in this case Latin BERT (Bamman and Burns, 2020) — designed for encoding sentences, to create these historical sentence embeddings. To prepare the training data, 50,000 random unique sentence pairs were selected from the five treebanks' 54,444 unique sentences, and each pair was assigned a similarity label: 1.0 if the authors were the same, .8 if they were from the same Latin age, and 0.0 if they were from different Latin ages.

The SBERT model, trained on that data, was able to embed sentences into 256-dimensional vectors, where sentences of the same Latin age are similar. You can see the historical sentence embeddings of the data from the five treebanks, mapped to two dimensions using UMAP (McInnes et al., 2020), in Fig. 2.

2.2. Model Architecture

The parser architecture, as described at the end of Section 1, is a modification of the dependency parser from Attardi et al. (2021) with the notable incorporation of the SBERT model, trained as described in Section 2.1.

Given a sentence, the model creates the historical sentence embedding and then creates the word embeddings and Latin BERT (Bamman and Burns, 2020) embeddings, which are concatenated together. These embeddings are then passed through the BiLSTMs, whose outputs are then concatenated with the historical sentence embedding. At this point, the values are run through the MLPs and biaffine classifiers as in the base model.

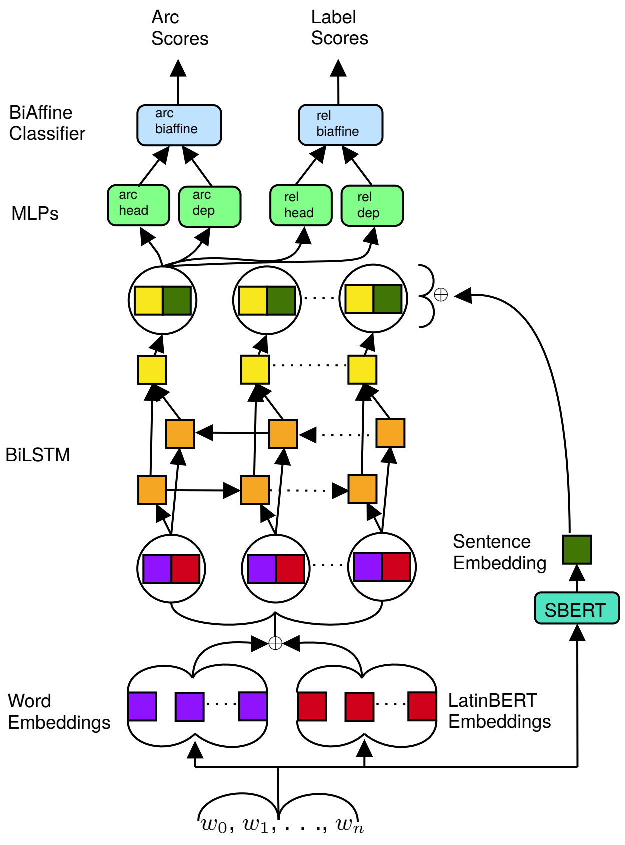

Figure 3: The Model Architecture

The model's architecture diagram can be seen in Fig. 3 with the omission of layer repetition (i.e., there are three BiLSTM layers, but the figure shows only one).

The data for training the model was selected through stratified sampling from the merged treebanks with respect to the Latin age. The training data uses 67% of the total data, and the remaining 33% were split evenly between test and development sets.

3. Experiments

Three models were selected for the experiments: the proposed model with SBERT, trained on the 5 treebanks; the base diaparser model (Attardi et al., 2021), trained on the 5 treebanks; and the pretrained Latin diaparser model, which was trained on ITTB (Passarotti, 2019) and LLCT (Cecchini et al., 2020b).

The only change to the hyperparameters from the original diaparser implementation was to change Adam's epsilon value to 1e-6 from the original 1e-12.

These three models were evaluated on the test data created for EvaLatin using the provided script[2]

[2]https://github.com/CIRCSE/LT4HALA/blob/master/2024/conll18_ud_eval.py

		Poetry				Prose			
Model	Metric	Precision	Recall	F1	AligndAcc	Precision	Recall	F1	AligndAcc
Diaparser with SBERT	CLAS	67.31	68.45	67.87	68.45	66.31	66.74	66.53	66.74
	LAS	68.33	68.33	68.33	68.33	69.72	69.72	69.72	69.72
	UCM				35.50				14.38
	LCM				15.14				3.68
Diaparser without SBERT	CLAS	67.33	68.59	67.95	68.59	65.83	66.60	66.21	66.60
	LAS	68.28	68.28	68.28	68.28	68.28	68.28	68.28	68.28
	UCM				33.87				14.38
	LCM				13.69				3.68
Pretrained Diaparser	CLAS	24.35	24.11	24.23	24.11	33.26	33.29	33.27	33.29
	LAS	26.45	26.45	26.45	26.45	39.39	39.39	39.39	39.39
	UCM				2.34				2.34
	LCM				0.00				0.0

Table 3: The results of the different models evaluated on the EvaLatin gold conllu files

to find the CLAS and LAS, and then the UCM and LCM were found for each model using diaparser's built-in evaluate function[3].

All of the code for the experimentation and data preparation is available on GitHub[4].

4. Results and Analysis

The results presented in Table 3 are a combination of the official ones for the EvaLatin submission, which used the historical embeddings, and subsequent evaluation runs done using the same script with the performance metrics as described in Section 3.

The results show no significant improvement with the inclusion of the historical sentence embedding proposed. Both models that were trained on the totality of the text provided did outperform the pretrained model, which is likely reflective of the lack of Classical Latin text in the model's training dataset compared to its proportion in the EvaLatin test set.

5. Conclusions

This paper experimented with the application of a historical Latin sentence embedding to help guide a Latin dependency parser, inspired by Altıntaş and Tantuğ (2023). Although the inclusion of this sentence embedding did not improve the overall performance of the parser, future research might focus on the inclusion of other features to guide Latin graph-based dependency parsing to enable better training across the treebanks.

6. Acknowledgements

This work was completed in part using the Discovery cluster, supported by Northeastern University's Research Computing team.

[3]https://github.com/Unipisa/diaparser/?tab=readme-ov-file#evaluation

[4]https://github.com/SufurElite/LatinDependencyParser

7. Bibliographical References

Mücahit Altıntaş and A. Cüneyd Tantuğ. 2023. Improving the performance of graph based dependency parsing by guiding bi-affine layer with augmented global and local features. *Intelligent Systems with Applications*, 18:200190.

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. 2021. Biaffine dependency and semantic graph parsing for EnhancedUniversal dependencies. In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 184–188, Online. Association for Computational Linguistics.

David Bamman and Patrick J. Burns. 2020. Latin BERT: A contextual language model for classical philology. *CoRR*, abs/2009.10053.

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

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Federica Gamba and Daniel Zeman. 2023. 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.

Dan Jurafsky and James H. Martin. *Chapter 18: Dependency Parsing*, 3rd edition, pages 5–6,15–16.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. 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 â€“ LT4HALA 2024*, Torino, Italy. European Language Resources Association.

8. Language Resource References

Bamman, David and Crane, Gregory. 2011. *The Ancient Greek and Latin Dependency Treebanks*. Springer Berlin Heidelberg, Theory and Applications of Natural Language Processing.

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

Cecchini, Flavio Massimiliano and Korkiakangas, Timo and Passarotti, Marco. 2020b. *A New Latin Treebank for Universal Dependencies: Charters between Ancient Latin and Romance Languages*. European Language Resources Association.

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

Passarotti, Marco. 2019. *The Project of the Index Thomisticus Treebank*.