# KU Leuven / Brepols-CTLO at EvaLatin 2024: Span extraction approaches for Latin dependency parsing

**Wouter Mercelis**
KU Leuven / Brepols-CTLO
KU Leuven: Blijde Inkomststraat 21, B-3000 Leuven, Belgium
Brepols-CTLO: Begijnhof 39, B-2300 Turnhout, Belgium
wouter.mercelis@kuleuven.be

**Abstract**

This report describes the KU Leuven / Brepols-CTLO submission to EvaLatin 2024. We present the results of two runs, both of which try to implement a span extraction approach. The first run implements span-span prediction, rooted in Machine Reading Comprehension, while making use of LaBERTa, a RoBERTa model pretrained on Latin texts. The first run produces meaningful results. The second, more experimental run operates on the token-level with a span-extraction approach based on the Question Answering task. This model finetuned a DeBERTa model, pretrained on Latin texts. The finetuning was set up in the form of a Multitask Model, with classification heads for each token's part-of-speech tag and dependency relation label, while a question answering head handled the dependency head predictions. Due to the shared loss function, this paper tried to capture the link between part-of-speech tag, dependency relation and dependency heads, that follows the human intuition. The second run did not perform well.

**Keywords:** Latin, NLP, dependency parsing, span extraction, question answering

## 1. Introduction

This short report describes the two runs of the KU Leuven / Brepols-CTLO team for the EvaLatin 2024 Evaluation Campaign (Sprugnoli, Iurescia and Passarotti, 2024), specifically for the Latin dependency parsing task. For each of the dependency parsing runs, this report will discuss the methodology (including the pre-trained language model), the actual results and a short discussion of the results.

## 2. MRC-based span-span prediction

### 2.1 Methodology

One of the first aims of our run was to look for an alternative to Dozat and Manning's (2017) Biaffine parser. Gan et al. (2022) propose a two-step method, called MRC-based span-span prediction, which firstly tries to predict subtrees in a dependency tree of a sentence, and secondly predicts the links between these proposed subtrees. The authors claim state-of-the-art performance on various benchmarks. In addition to this, the method also works with non-projective dependency trees, which is important for languages with a relatively free word order such as Latin.

Gan et al's (2022) method requires a pretrained language model as a starting point. We opted for the RoBERTa-like LaBERTa (Riemenschneider and Frank, 2023) for the following reasons. Firstly, we encountered some technical difficulties using our own DeBERTa-based model, as the tokenizer approach of Gan et al. (2022) was not compatible with our DeBERTa-based model. Due to time constraints, we decided to switch to a model with broader support. Furthermore, we chose LaBERTa because the original paper performed best with a similar XLM-RoBERTa (Conneau *et al.*, 2020) model. Therefore, we decided to use a model which has an equivalent architecture and training process.

For the training data, we took advantage of the work of Gamba and Zeman (2023), in which harmonization measures were introduced to reduce the disparity between the five Latin Universal Dependencies (UD) (de Marneffe *et al.*, 2021) treebanks. We opted to train on the Perseus (UD v2.13)(Bamman and Crane, 2011) and the ITTB (UD v2.13) (Passarotti, 2019) treebanks, as the Perseus treebank aligns the most with the test data. The ITTB treebank is mainly included because of its large size.

Concerning training parameters, we used the default parameters out-of-the-box, with a reduced batch size of 4 to prevent CUDA out-of-memory errors.

### 2.2 Results

| | Poetry | | | |
|---|---|---|---|---|
| | Precision | Recall | F1 | AligndAcc |
| CLAS | 57.26 | 57.42 | 57.34 | 57.42 |
| LAS | 59.02 | 59.02 | 59.02 | 59.02 |
| | Prose | | | |
| | Precision | Recall | F1 | AligndAcc |
| CLAS | 63.93 | 63.49 | 63.71 | 63.49 |
| LAS | 67.32 | 67.32 | 67.32 | 67.32 |

Table 1: KU Leuven/Brepols-CTLO run 1 results



*LT4HALA 2024@LREC-COLING-2024*, pages 203–206

| | Poetry | | | |
|---|---|---|---|---|
| | Precision | Recall | F1 | AligndAcc |
| CLAS | 74.34 | 74.72 | 74.53 | 74.72 |
| LAS | 75.75 | 75.75 | 75.75 | 75.75 |
| | Prose | | | |
| | Precision | Recall | F1 | AligndAcc |
| CLAS | 73.58 | 72.80 | 73.19 | 72.80 |
| LAS | 77.41 | 77.41 | 77.41 | 77.41 |

Table 2: ÚFAL LatinPipe_1 results

In Table 1, the results of our first run are summarized, while in Table 2, the results of the best-performing team are shown in comparison.

### 2.3 Discussion

To start with, the Chu-Liu-Edmonds algorithm failed once to generate a proper graph, resulting in a dependency tree with two roots, which is not well-formed. This was solved by considering the second root as a conjunction of the first one.

Apart from this slight mishap, the results were quite disappointing. For a large part, this can be explained by a misinterpretation of the guidelines from our part. As the guidelines contained information about all the main relations, and referred to the UD website for more information about the subrelations, we wrongly interpreted this as supplementary information, meaning that the subrelations would not be taken into account during evaluation. This had a considerable impact on our accuracy numbers. For example, almost half of the wrongly predicted dependency relation labels contained a subrelation in the gold data (913 out of 1871 wrong predictions).

Another problematic notion are coordinating constructions. For the 805 “conj” instances in the prose gold data, in 305 cases the wrong head was predicted. Similarly, for the 605 “cc” instances, 175 receive a wrong head relation. The same method reveals that 96 of the 299 roots do not receive the correct head relation. This is can possibly be attributed to differences in annotation between test and training data. Furthermore, with regards to ellipsis in clauses, the UD framework prefers assigning the root label to non-verbs in verb-final languages such as Latin. Our model has trouble taking this into account, preferring to use the final verb as a root instead.

## 3. Multitask Question Answering

### 3.1 Methodology

Our second run was much more experimental. During work on word alignment, we used a span extraction approach that is also used in Question Answering. As an experiment, we tried to apply this naively to dependency parsing as well. In fact, the first run can be seen as a more elaborate approach to this problem, in a way that is more suited to the task as well.

For this second run, we made use of a Multitask Model, in which a pretrained language model is finetuned using different classification heads, with a shared loss function, as shown in Figure 1. For a theoretical survey, see Crawshaw (2020). Due to this shared loss function, the model is not only very efficient, it also quantifies the learning of inter-task dependencies and generalizes well, following our intuition that the relation labels, the relations themselves and the part-of-speech tags all influence each other.

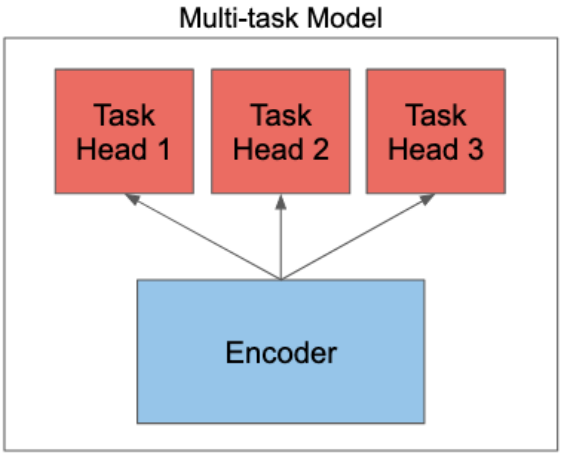


Figure 1: Architecture of a multi-task model

In this task we used our own DeBERTa-model (He et al., 2023). Starting with DeBERTa v3, the pretraining approach is very similar to the ELECTRA approach (Clark *et al.*, 2020), with better results. This Latin DeBERTa model is the successor of the ELECTRA model that we used in the 2022 Evalatin Competition (Mercelis and Keersmaekers, 2022). It is also trained on Brepols’ Library of Latin Texts[1], in addition to various online corpora such as the CAMENA project[2] and web data such as the Latin Wikisource[3] and Wikipedia[4]. As copyright rests on the Library of Latin Texts, this model is not publicly available.

1 https://www.brepols.net/series/LLT-O
2 https://mateo.uni-mannheim.de/camenahtdocs/camena_e.html
3 https://la.wikisource.org/wiki/Pagina_prima
4 https://la.wikipedia.org/wiki/Vicipaedia:Pagina_prima

For the finetuning data, we experimented with only ITTB (UD v2.13) (Bamman and Crane, 2011) as our training data in the first place, planning to add more data if the experiments were fruitful. Unfortunately, these addition have not yet taken place due to time constraints.

As the multitask model performed well during experiments with jointly predicting morphological tags, we tried extending it to dependency parsing as well. Crucially, this task is fundamentally different from morphological tagging in the sense that on the one hand, tokens cannot be predicted in a vacuum: they are inherently part of a sentence. On the other hand, due to the nature of our span extraction task, we have to input the tokens one by one. By contrast, in token classification tasks, the input is an entire sentence of tokens that are predicted in one go.

This has severe complications for the training process of our model. Table 3 shows in a simplified way how our model processes the data. For clarity, the same sentence is used in the example. Note that during the experiment, this data is shuffled at random, so the same sentence will be spread throughout the data for each of the finetuning tasks.

| Tokens | Training task |
|---|---|
| [CLS] unde et dicit … [SEP] [PAD] ... | token classification (POS) |
| [CLS] unde et dicit … [SEP] [PAD] ... | token classification(deprel) |
| [CLS] [SEP] unde et dicit … [SEP] [PAD] ... | question answering, first token |
| [CLS] unde [SEP] et [SEP] dicit … [SEP] [PAD] ... | question answering, second token |
| [CLS] unde et [SEP] dicit [SEP] … [SEP] [PAD] ... | question answering, third token |

Table 3: Overview of the training data structure

As the data are shuffled at random, the part-of-speech tagging, the dependency relations and the dependency heads are not learned at the same stage in the training process.

Adding to this, we encountered more technical difficulties, as said above, resulting in a batch size of 1, which is also not ideal. We did not have enough time to have an in-depth look into these issues. Also, due to these time constraints, we could not try this approach with the LaBERTa model as well.

### 3.2 Results

| | Poetry | | | |
|---|---|---|---|---|
| | Precision | Recall | F1 | AligndAcc |
| CLAS | 5.36 | 5.33 | 5.34 | 5.33 |
| LAS | 5.44 | 5.44 | 5.44 | 5.44 |
| | Prose | | | |
| | Precision | Recall | F1 | AligndAcc |
| CLAS | 3.79 | 3.76 | 3.78 | 3.76 |
| LAS | 3.70 | 3.70 | 3.70 | 3.70 |

Table 4: KU Leuven/Brepols-CTLO run 2

Table 4 shows the results of our second run. See table 2 for a comparison with the results of the best-performing team.

### 3.3 Discussion

As seen above, the results are not meaningful at all. Unexpectedly, the model performs worse on prose than on poetry. However, the obtained results are so low that this does not tell anything about the performance of the model. In fact, we only included this run so we could discuss the architecture *in se*. We could see that the implementation of the Chu-Liu-Edmonds algorithm had difficulties providing a meaningful graph, resulting in many sentences with multiple predicted roots. We used the same algorithm as in the previous model to reduce them to well-formed sentences. This however resulted in many wrongly predicted heads. However, the dependency relation labels did not suffer from this approach at all. For the prose data, 4402 tokens out of 5840 received the right dependency relation label, outperforming our first run, which labeled only 3969 tokens correctly. This leads us to believe that the multi-task approach is not the problem, but rather the current question-answering implementation that predicts the dependency heads.

Thus, we believe that with a proper technical implementation, there is something to say for this approach. However, the focus needs to shift from the token level to the sentence level.

## 4. Conclusion

In conclusion, our first run performed reasonably well, unfortunately hampered by the subrelation issue. This shows that there are performant alternatives to Dozat and Manning's Biaffine parser. Our second run did not perform well, but can serve as a building block for further research, as this multi-task model shows promise especially in the prediction of dependency labels.

## 5. Acknowledgments

Our work has been funded by grant no. HBC.2021.0210 of Flanders Innovation and Entrepreneurship.

## 6. Bibliographical References


Clark, K. *et al.* (2020) 'ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators', *arXiv:2003.10555 [cs]* [Preprint]. Available at: http://arxiv.org/abs/2003.10555 (Accessed: 4 October 2021).Conneau, A. et al. (2020) 'Unsupervised Cross-lingual Representation Learning at Scale', in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online: Association for Computational Linguistics, pp. 8440–8451. Available at: https://doi.org/10.18653/v1/2020.acl-main.747.

Crawshaw, M. (2020) 'Multi-Task Learning with Deep Neural Networks: A Survey'. arXiv. Available at: http://arxiv.org/abs/2009.09796 (Accessed: 10 March 2024).

Dozat, T. and Manning, C.D. (2017) 'Deep Biaffine Attention for Neural Dependency Parsing', *arXiv:1611.01734 [cs]* [Preprint]. Available at: http://arxiv.org/abs/1611.01734 (Accessed: 19 March 2021).

Gamba, F. and Zeman, D. (2023) 'Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD', in L. Grobol and F. Tyers (eds) *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*. *UDW-SyntaxFest 2023*, Washington, D.C.: Association for Computational Linguistics, pp. 7–16. Available at: https://aclanthology.org/2023.udw-1.2 (Accessed: 10 March 2024).

Gan, L. *et al.* (2022) 'Dependency Parsing as MRC-based Span-Span Prediction', in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland: Association for Computational Linguistics, pp. 2427–2437. Available at: https://doi.org/10.18653/v1/2022.acl-long.173.

He, P., Gao, J. and Chen, W. (2023) 'DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing'. arXiv. Available at: https://doi.org/10.48550/arXiv.2111.09543.

Mercelis, W. and Keersmaekers, A. (2022) 'An ELECTRA Model for Latin Token Tagging Tasks', in R. Sprugnoli and M. Passarotti (eds) *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*. *LT4HALA 2022*, Marseille, France: European Language Resources Association, pp. 189–192. Available at: https://aclanthology.org/2022.lt4hala-1.30 (Accessed: 10 March 2024).

Riemenschneider, F. and Frank, A. (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)*. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada: Association for Computational Linguistics, pp. 15181–15199. Available at: https://doi.org/10.18653/v1/2023.acl-long.846.

Sprugnoli, R., Iurescia, F. and Passarotti, M. (2024) 'Overview of the EvaLatin 2024 Evaluation Campaign', in R. Sprugnoli and M. Passarotti (eds) *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages @LT4HALA 2024*. Torino, Italy: European Language Resources Association.


## 7. Language Resource References


Bamman, D. and Crane, G. (2011) 'The Ancient Greek and Latin Dependency Treebanks', in C. Sporleder, A. van den Bosch, and K. Zervanou (eds) *Language Technology for Cultural Heritage*. Heidelberg: Springer, pp. 79–98. Available at: https://doi.org/10.1007/978-3-642-20227-8_5.

de Marneffe, M.-C. *et al.* (2021) 'Universal Dependencies', *Computational Linguistics*, 47(2), pp. 255–308. Available at: https://doi.org/10.1162/coli_a_00402.

Passarotti, M. (2019) 'The Project of the Index Thomisticus Treebank', in M. Berti (ed.) *Digital Classical Philology. Ancient Greek and Latin in the Digital Revolution*. Berlin - Boston: De Gruyter Saur, pp. 299–320. Available at: https://doi.org/10.1515/9783110599572-017.