Extending `omnes flores` for the EvaLatin 2026 Dependency Parsing Tasks

Hiroshi Matsuda, Masayuki Asahara
Megagon Labs, Tokyo, Recruit Co., Ltd., National Institute for Japanese Language and Linguistics
Marunouchi 1-9-2, Chiyoda, Tokyo, Japan, Midorichou 10-2, Tachikawa, Tokyo, Japan
hiroshi_matsuda@megagon.ai, masayu-a@ninjal.ac.jp

Abstract

`omnes flores` is an NLP framework based on Universal Dependencies (UD) that utilizes multilingual Large Language Models (LLMs), and its default model is trained on data from 40 UD languages comprising 40 treebanks. For the EvaLatin 2026 Dependency Parsing Tasks, we extended the training data of `omnes flores` by incorporating six public Latin treebanks from UD and trained a dependency parsing model using the extended training data. The dependency parser of `omnes flores` normally takes a list of word FORM values as input. However, since the EvaLatin 2026 test data includes an UPOS column, we investigated whether incorporating both FORM and UPOS during both training and inference could improve parsing accuracy. Our experiments show that training using both FORM and UPOS improves performance by 0.5-1.0 LAS points on Prose compared with training using only FORM, but decreases performance by 5 points on Poetry.

Keywords: Universal Dependencies, dependency parsing, LoRA SFT, EvaLatin

1. Introduction

In recent years, Large Language Models (LLMs) have improved to the point that they can be applied to tasks that were traditionally handled by Natural Language Processing (NLP) techniques. Tuning LLMs for specific tasks or domains has become easier in recent years. This is largely due to few-shot prompting and the spread of Parameter-Efficient Fine-Tuning (PEFT) techniques used for supervised fine-tuning (SFT) on instruction—response pairs. As a result, end users can increasingly treat LLMs as black boxes and apply them to a wide range of downstream tasks.

Quantitative analysis and NLP technology. Because LLMs are large models, inference is computationally expensive, and LLMs by themselves perform poorly on quantitative analysis of text (they struggle even with simple tasks such as word-frequency statistics). As a result, the use of LLMs for mining tasks over large-scale text collections has not progressed very far, and conventional NLP techniques continue to be used for this kind of task.

Dependency parsing is often used not only in computational linguistics research but also in industrial applications such as text mining, and Universal Dependencies (UD) (Nivre et al., 2016, 2020) is widely used for training and evaluating dependency parsing models. UD began to be developed around 2015, and the latest version at the time of writing (r2.17) supports 186 languages.

Sustainability of NLP technology. Most open-source NLP frameworks in active use today were designed before the widespread adoption of LLMs. They mainly implement functional components such as part-of-speech tagging and dependency parsing with encoder-based models, and assemble those components into a processing pipeline. Developing and maintaining such conventional NLP pipelines requires advanced specialized knowledge and engineering skills. However, most market demand has already shifted from conventional NLP to LLMs, and it may become difficult to maintain the organizational capacity needed to improve and support conventional NLP pipelines.

Syntactic parsing with LLMs. With an eye toward ensuring the sustainability of NLP development in the technology market, we propose a practical approach in which conventional NLP pipelines are replaced with LLMs while improving accuracy. Based on the method of the prior work (Matsuda et al., 2025), we perform SFT with Low-Rank Adapters (LoRA) (Hu et al., 2022) on a multilingual LLM whose weights are publicly available by using a further improved version of the dependency parsing prompt from Matsuda et al. (2025).

2. Related Work

2.1. NLP Frameworks

Below we briefly survey NLP frameworks whose parsing functionality continues to be improved at the time of writing, in order of public release. `spaCy` performs deterministic or beam-search dependency parsing with the Non-Monotonic Arc-Eager Transition System (Honnibal and Johnson, 2015). Multitask learning is possible between the transformer and the parsing components. `Stanza`

Proceedings of LT4HALA 2026 @ LREC 2026, pages 448–452.

can learn the entire pipeline end to end, from language identification to dependency parsing (Qi et al., 2020). Its dependency parser uses an extended biaffine model with Bi-LSTMs. In recent years, constituency parsing has also been addressed (Bauer and Manning, 2025). `Spark NLP` is an NLP framework that claims support for more than 200 languages and a wide range of development environments (Python/Scala/Java/R) (Kocaman and Talby, 2021). Details such as the dependency parsing algorithm are not publicly available. `HanLP` performs multitask learning between a shared encoder and multiple parsing components. A biaffine-derived model is used for dependency parsing (He and Choi, 2021).

2.2. Dependency Parsing

Many of the NLP frameworks mentioned above perform dependency parsing with variants of the biaffine model (Dozat and Manning, 2017). Recently, methods such as Hexatagger (Amini et al., 2023), which represent local tree structure with a small number of tags and learn it with an encoder model, have been reported to achieve high parsing accuracy. For dependency parsing with autoregressive decoder models such as LLMs, methods that express dependency structure through word indices in a tabular format such as `CoNLL-U` are considered more accurate than methods that output trees in bracketing notation (Matsuda et al., 2025). In this paper, we extend the method of the prior work (Matsuda et al., 2025) and propose a method in which processing steps other than dependency parsing are also carried out by LLMs.

3. Proposed Method

We modified `omnes flores`[1], a unified NLP framework which employs LLMs for tasks such as language identification, sentence delimitation, word segmentation, part-of-speech tagging, and dependency parsing, in terms of both training data and input features for EvaLatin Dependency Parsing Tasks (Iurescia et al., 2026). For details of the methods of SFT and inference of the underlying `omnes flores`, refer to the official website.

3.1. Adding public UD Latin Treebanks

The default model of `omnes flores` uses the following 40 treebanks each of which has at least 40,000 words in its training split and a license that permits commercial use.

UD_Armenian-ArmTDP, UD_Belarusian-HSE, UD_Bororo-BDT, UD_Chinese-GSD, UD_Chinese-GSDSimp, UD_Croatian-SET, UD_Czech-CAC, UD_Danish-DDT, UD_Dutch-Alpino, UD_English-EWT, UD_Estonian-EWT, UD_Finnish-TDT, UD_French-GSD, UD_German-GSD, UD_Haitian_Creole-Adolphe, UD_Hebrew-IAHLTwiki, UD_Icelandic-GC, UD_Indonesian-GSD, UD_Irish-IDT, UD_Japanese-GSDLUW, UD_Korean-Kaist, UD_Latvian-LVTB, UD_Lithuanian-ALKSNIS, UD_Naija-NSC, UD_Norwegian-Nynorsk, UD_Persian-PerDT, UD_Portuguese-Porttinari, UD_Romanian-RRT, UD_Russian-GSD, UD_Scottish_Gaelic-ARCOSG, UD_Serbian-SET, UD_Sindhi-Isra, UD_Slovak-SNK, UD_Slovenian-SSJ, UD_Spanish-GSD, UD_Swedish-Talbanken, UD_Thai-TUD, UD_Turkish-BOUN, UD_Ukrainian-ParlaMint, UD_Western_Armenian-ArmTDP.

We added the following six Latin treebanks to the 40 treebanks listed above and finally performed LoRA SFT on the merged training set.

UD_Latin-ITTB, UD_Latin-LLCT, UD_Latin-UDante, UD_Latin-CIRCSE, UD_Latin-Perseus, UD_Latin-PROIEL.

3.2. Dialog Template

The dialog template used for dependency parsing is shown in Figure 1. We input `language`, `sentence`, and `indexed word list` together with a prompt instructing UD part-of-speech tagging, head prediction, and dependency relation labeling, and train the correspondence between the input and the response, which is a TSV-formatted dependency structure output.

Prior work (Matsuda et al., 2025) reported that the best accuracy is obtained when prompts are used to perform UD part-of-speech tagging, head prediction, and dependency relation labeling in three steps. However, increasing the number of steps lengthens the output context, which strongly affects inference efficiency. In addition, because of LLM context-length limits, truncation can occur at the end of the output for long inputs. In order to shorten the output context, `omnes flores` performs all dependency parsing subtasks in a single step and trains the model with the word-form column omitted from the output TSV.

In preliminary experiments, we confirmed that the context length of the proposed method (instruction plus response) is less than half that of Matsuda et al. (2025), while achieving accuracy comparable to Matsuda et al. (2025) on three metrics: UD part-of-speech tagging accuracy (UPOS), unlabeled attachment score (UAS), and labeled attachment score (LAS).

[1] https://megagonlabs.github.io/omnes-flores

{role: system} `You are a linguist and specialize in dependency parsing based on Universal Dependencies.`

{role: user} `We will now perform dependency parsing on` **`Latin`** `sentence.`
`After splitting the input sentence into words with indexes as shown below, create a TSV with four fields:`
`word index from 1 to` **`7`** `+ part of speech + the dependent word index + the Universal Dependencies relation.`

`input sentence:`
species intelligibilis similitudo est alicuius intellecti.

`words:`
1 species
2 intelligibilis
3 similitudo
4 est
5 alicuius
6 intellecti
7 .

{role: assistant}

```
1   NOUN    3   nsubj
2   ADJ     1   amod
3   NOUN    0   root
4   AUX     3   cop
5   PRON    3   nmod
6   VERB    5   acl
7   PUNCT   3   punct
<eos>
```

Figure 1: An example of the **1-step-no-form** dialog instance. The parts that change from instance to instance are shown in **`bold`**. The `shaded region` in the assistant-role corresponds to the range over which the loss gradient is computed during training, and to the decoded text during inference. At inference time, the span from the system-role up to the assistant-role header is provided as input, and decoding of the subsequent segment continues until `<eos>` is generated.

Train	Test - LAS						
Dataset	circse	ittb	llct	perseus	proiel	udante	Macro AVG
circse	51.3	25.4	23.5	26.7	53.6	21.2	33.6
ittb	24.3	90.5	56.8	71.0	30.5	70.1	57.2
llct	1.0	63.2	94.1	60.8	1.4	55.4	46.0
perseus	13.9	57.2	42.5	65.0	10.2	45.8	39.1
proiel	59.4	51.5	47.6	53.6	84.6	43.4	56.7
udante	5.3	69.3	52.5	65.9	9.5	63.3	44.3
latin6	61.6	85.8	90.6	74.2	78.7	71.0	77.0
40lang+latin6	61.9	86.5	90.8	75.0	80.5	72.8	77.9

Table 1: LAS scores under various settings when using FORM as input. `latin6` is obtained by merging the training sets of `circse`, `ittb`, `llct`, `perseus`, `proiel`, and `udante`. `40lang+latin6` is the model trained on data obtained by merging the 40-language training data of `omnes flores` and `latin6`.

4. Experiments

4.1. Settings and Accuracy Measures

Hardware. Google Cloud G4 instance (8 x RTX Pro 6000 GPUs, 384-core AMD Turin CPUs, 768 GB RAM), 250 GB Hyperdisk, 10 TB Filestore.
Software. Ubuntu 24.04, CUDA 12.8, Python 3.12.11, PyTorch 2.8.0, Transformers 4.57.0, Tokenizers 0.22.1, TRL 0.19.1, PEFT 0.17.1.
Base model. We use gemma-2-9b, which showed good accuracy in Matsuda et al. (2025).
Hyperparameters. We follow the settings of Matsuda et al. (2025), but reduce the number of epochs from 3 to 2 in order to shorten training time. The remaining settings are as follows: max_seq_length: 8,192, lr: 3e-4, lr_scheduler: cosine_with_min_lr, min_lr: 0.1, lora_r: 8, lora_dropout: 0.05, target_modules: all-linear.
Accuracy measures. We use UD part-of-speech tagging accuracy (UPOS), unlabeled attachment score (UAS), and labeled attachment score (LAS).

4.2. Evaluation on public UD Latin Treebanks

We performed LoRA SFT on `gemma-2-9b` with the dialog instances generated in the way explained in Section 3.2, and then carried out inference and evaluation with various settings. The results are shown in Tables 1 and 2.

Table 1 shows the following types of results when using FORM on input:
- Trained on each of the UD Latin treebanks
- Trained on a merged set of the six UD Latin treebanks (`latin6`)
- Trained on the default `omnes flores` model with the addition of `latin6` to the training data for 40 languages (`40lang+latin6`)

Table 2 shows the same types of results as Table 1 when using both FORM and UPOS.

The `40lang+latin6` model shows the best LAS across all public UD Latin treebanks. In addition, when UPOS is used together with FORM as input, an improvement in LAS of 2.1 points is achieved.

Regarding the addition of `40lang`, the following

Train	Test - LAS							Effect of Adding UPOS
Dataset	circse	ittb	llct	perseus	proiel	udante	Macro AVG	
circse	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-33.6
ittb	0.1	80.0	55.5	67.6	0.0	63.8	44.5	-12.7
llct	9.3	57.8	81.2	57.6	9.5	51.3	44.4	-1.5
perseus	0.1	3.7	0.2	3.7	0.0	2.1	1.6	-37.5
proiel	54.2	58.6	50.5	61.2	72.5	49.3	57.7	1.0
udante	6.2	7.5	5.2	3.4	9.5	8.0	6.6	-37.7
latin6	61.5	82.3	88.2	73.9	77.8	70.0	75.6	-1.4
40lang+latin6	65.3	87.2	91.3	77.9	81.8	76.6	80.0	2.1

Table 2: The LAS scores when using both FORM and UPOS on input.

System	Setting	Poetry		Prose	
		CLAS (P/R/F1/Rank)	LAS (P/R/F1/Rank)	CLAS (P/R/F1/Rank)	LAS (P/R/F1/Rank)
Omnes Flores_1	with subtypes	57.65 / 64.29 / 60.79 / 5	60.83 / 60.83 / 60.83 / 5	81.32 / 79.90 / 80.60 / 5	83.26 / 83.26 / 83.26 / 4
Omnes Flores_1	no subtypes	63.22 / 63.11 / 63.17 / 5	65.64 / 65.64 / 65.64 / 5	84.08 / 84.26 / 84.17 / 5	86.10 / 86.10 / 86.10 / 5
Omnes Flores_2	with subtypes	52.36 / 58.74 / 55.37 / 6	54.74 / 54.74 / 54.74 / 6	82.69 / 80.05 / 81.35 / 3	**83.74 / 83.74 / 83.74 / 2**
Omnes Flores_2	no subtypes	58.16 / 58.57 / 58.37 / 6	60.20 / 60.20 / 60.20 / 6	85.14 / 85.42 / 85.28 / 4	**87.09 / 87.09 / 87.09 / 4**

Table 3: Summary of EvaLatin 2026 Syntactic Parsing Results submitted by the Omnes Flores team. Model names are as follows: models that reference only FORM in the input are named Omnes Flores_1, and models that reference both FORM and UPOS are named Omnes Flores_2. Each cell reports Precision / Recall / F1 / Rank among all 8 models submitted to the EvaLatin 2026 Dependency Parsing Tasks. The evaluation results for all models submitted by all teams are publicly available in the official EvaLatin repository.

three reasons can be considered for the improved accuracy in Latin, even though almost all of the treebanks comprising 40lang are modern languages:

- Increased training volume
- Generalization from other Romance languages
- Generalization from languages with many Latin-derived words, such as English

Based on these results, we submitted the outputs of the 40lang+latin6 model to the EvaLatin 2026 Dependency Parsing Tasks (Iurescia et al., 2026). Model names are as follows: models that reference only FORM in the input are named Omnes Flores_1, and models that reference both FORM and UPOS are named Omnes Flores_2.

4.3. EvaLatin 2026 Results

The official evaluation results for Omnes Flores team submissions in EvaLatin 2026 Dependency Parsing Tasks are shown in Table 3. The Omnes Flores_1 refers to the outputs from the model trained with reference to FORM only, and Omnes Flores_2 refers to the outputs from the model trained with reference to both FORM and UPOS.

Across the two submissions, performance is clearly stronger on Prose than on Poetry. Removing subtypes improves every official F1 score, with gains ranging from 2.4 to 5.5 points. Omnes Flores_1 is consistently stronger on Poetry, whereas Omnes Flores_2 is consistently stronger on Prose.

The best Poetry result among the two systems is 65.6 LAS F1 from Omnes Flores_1 without subtypes, while the best Prose result is 87.1 LAS F1 from Omnes Flores_2 without subtypes. In the final ranking tables, Omnes Flores_2 is relatively competitive on Prose—it places 3rd in CLAS with subtypes and ties for 2nd in LAS with subtypes—while both models remain 5th or 6th in the Poetry rankings.

5. Conclusion

We propose a method that extends omnes flores by adding both training data and input features and show that it improves dependency parsing accuracy on Latin treebanks.

We submitted the outputs of the following two models to EvaLatin 2026 Dependency Parsing Tasks: (1) a model trained by adding six Latin treebanks to the training data of the default model of omnes flores, and (2) a model trained by adding UPOS to the input features of the model in (1).

In the final ranking table, one of our systems tied for 2nd in Prose LAS with subtypes, whereas both systems ranked 5th or 6th on Poetry.

Based on these results, we released an expanded version of the omnes flores model[2], trained on 84 languages comprising 99 treebanks, by adding 44 languages (including Latin), represented by 59 treebanks with independent training sets in the UD v2.17 treebank suite, in order to support languages not included among the 40 languages used to train the standard omnes flores model[3].

Future work will pursue two directions. First, we will investigate whether incorporating morphological features during both training and inference can further improve parsing accuracy. Second, we will analyze the factors underlying the performance degradation observed on the Poetry data when UPOS information is used in addition to FORM.

6. Acknowledgment

This work was conducted as part of a collaborative research project between Recruit Co., Ltd. and the National Institute for Japanese Language and Linguistics.

[2] https://huggingface.co/megagonlabs/omnes-flores-84-lang-99-treebank-non-commercial-v0

[3] https://huggingface.co/megagonlabs/omnes-flores-40-lang-41-treebank-v0

7. Bibliographical References

Afra Amini, Tianyu Liu, and Ryan Cotterell. 2023. Hexatagging: Projective dependency parsing as tagging. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1453–1464, Toronto, Canada.

John Bauer and Christopher D. Manning. 2025. High-accuracy transition-based constituency parsing. In *Proceedings of the 18th International Conference on Parsing Technologies (IWPT, SyntaxFest 2025)*, pages 26–39, Ljubljana, Slovenia.

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

Han He and Jinho D. Choi. 2021. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5555–5577, Punta Cana, Dominican Republic.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Federica Iurescia, Marco Passarotti, and Rachele Sprugnoli. 2026. Overview of the Dependency Parsing Task at EvaLatin 2026. In *Proceedings of the Fourth Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2026)*, Palma, Mallorca (Spain). ELRA.

Veysel Kocaman and David Talby. 2021. Spark NLP: Natural language understanding at scale. *Software Impacts*, page 100058.

Hiroshi Matsuda, Chunpeng Ma, and Masayuki Asahara. 2025. Step-by-step instructions and a simple tabular output format improve the dependency parsing accuracy of LLMs. In *Proceedings of the 18th International Conference on Parsing Technologies (IWPT, SyntaxFest 2025)*, pages 11–19, Ljubljana, Slovenia.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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. European Language Resources Association.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.