Typology-aware Multilingual Morphosyntactic Parsing with Functional Node Filtering

Kutay Acar and **Gülşen Eryiğit**
ITU NLP Research Group, Istanbul Technical University, Türkiye
{acarku18, gulsen.cebiroglu}@itu.edu.tr

Abstract

This paper presents a system for the UniDive Morphosyntactic Parsing (MSP) Shared Task, where it ranked second overall among participating teams. The task introduces a morphosyntactic representation that jointly models syntactic dependencies and morphological features by treating content-bearing elements as graph nodes and encoding functional elements as feature annotations, posing challenges for conventional parsers and necessitating more flexible, linguistically informed approaches. The proposed system combines a typology-aware, multitask parser with a multilingual content/-function classifier to handle structural variance across languages. The architecture uses adapter modules and language embeddings to encode typological information. Evaluations across 9 typologically varied languages confirm that the system can accurately replicate both universal and language-specific morphosyntactic patterns.

1 Introduction

Morphosyntactic parsing aspires to integrate syntactic structure with fine-grained morphological annotation to offer a deeper and linguistically neutral understanding of sentence structure. The UniDive Morphosyntactic Parsing (MSP) Shared Task offers a novel paradigm that challenges conventional parsing assumptions by restructuring dependency trees around content-bearing elements and functional grammatical units. In this new schema, only content nodes—such as lexical verbs, nouns, and adjectives—are represented explicitly in the graph, while functional elements like auxiliaries, clitics, and determiners are removed and represented as features of the content words. Moreover, the format integrates abstract nodes for dropped or elided arguments, which are syntactically required but not present on the surface, as commonly seen in pro-drop languages.

Such a shift from surface-token-based syntax to deeper morphosyntactic abstraction makes this task both linguistically rich and technically challenging. Traditional parsers must be adapted to filter out functional nodes and accommodate missing heads, necessitating new modeling strategies. In response to the novel task format, this study adapts the UDapter model (Üstün et al., 2020, 2022), a typologically informed multilingual dependency parser. The original architecture is extended with a content/function classifier and decoding routines are modified accordingly, while multitask learning is leveraged for both dependency parsing and morphological tagging. This approach not only conforms to the structural assumptions of the MSP task but also exploits cross-lingual signals across 9 diverse languages.

Evaluated in the official shared task, the proposed system ranks second overall. As the first adaptation of UDapter to the MSP framework, it introduces a content/function classifier to align parsing with the task's structure. By combining multilingual pretraining, typological conditioning, and multitask learning, the system effectively integrates syntax and morphology beyond surface-level representations, offering a robust solution for typologically diverse parsing.

This paper is structured as follows: Section 2 reviews related work on dependency parsing and morphosyntactic modeling. Section 3 presents the system architecture. Section 4 details the experimental setup and results. Section 5 concludes the paper and outlines directions for future research.

2 Related Work

Dependency parsing methods are traditionally grouped into two paradigms: transition-based and graph-based approaches. Transition-based parsers, such as those by Nivre (2003); Nivre et al. (2006); Hall et al. (2007), incrementally construct depen-

Proceedings of The UniDive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing, pages 27–33
August 26, 2025

dency trees through local decisions. These models are computationally efficient but often suffer from error propagation. A significant advancement in this line was the biaffine parser of Dozat and Manning (2016), built on Kiperwasser and Goldberg (2016), which introduced attention-based arc and label scoring and achieved state-of-the-art results across many languages.

Multilingual dependency parsing has gained traction due to the Universal Dependencies (UD) framework (de Marneffe et al., 2021), which standardizes syntactic annotation across more than 100 languages. Multilingual benchmarks enabled by UD treebanks include CoNLL-X (Buchholz and Marsi, 2006), CoNLL 2007 (Nivre et al., 2007), and CoNLL 2018 (Zeman et al., 2018). Full parsing pipelines from raw text to dependency structures in 75 languages were evaluated in the CoNLL 2018 shared task.

Modern approaches increasingly rely on multilingual pretrained language models like mBERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020). UDify (Kondratyuk and Straka, 2019) was among the first to use mBERT for joint multitask prediction of POS tags, morphological features, lemmas, and dependencies across all UD languages. Its universal parameter sharing, however, made it less flexible for languages with low resources or typological distance. By introducing lightweight adaptor modules in between mBERT layers, UDapter (Üstün et al., 2020, 2022) addressed this issue and preserved the advantages of multilingual pretraining while enabling typology-aware, language-specific transformation. This approach improved generalization, especially when resources were scarce or the setting is zero shot.

The Morphosyntactic Parsing (MSP) Shared Task (Goldman et al., 2025) presents an updated parsing approach in which function words are appended as morphological features and only content words are represented as syntactic nodes. Abstract nodes are also integrated to represent dropped or implicit arguments, such as pro-drop pronouns. This annotation strategy decouples word segmentation from syntactic structure, enabling more typologically robust morphosyntactic parsing.

3 System Architecture

The system integrates a multilingual content/function classifier with a universal dependency parser to handle the structural transformations introduced by the MSP shared task. In this format, functional nodes—such as determiners, auxiliaries, adpositions, and punctuation—are excluded from the dependency graph by assigning them null heads, rendering them incompatible with standard parsing methods. As illustrated in Figure 1, adpositions like *sonra* and clitic constructions like *–kine* are not linked via dependency arcs. Instead, their morphological contributions are absorbed into the parent node: for instance, the combination *gittikten sonra* alters the original *Case=Abl* to *Case=Tps*, and *sizin* + *–kine* merges into a single node with *Case=Gen;Dat*. To support this abstraction, a BERT-based classifier is applied in preprocessing to identify and remove functional tokens before parsing. The UDapter model, equipped with typology-aware adapters and multitask heads for both morphological tagging and dependency parsing, then processes the remaining content nodes under this structurally modified scheme.

3.1 UDapter Model Architecture

Built on top of mBERT, UDapter is a multilingual, multitask neural architecture intended for morphological tagging and universal dependency parsing across typologically disparate languages. By combining shared task heads with language-specific adapter modules, it facilitates effective cross-lingual generalization, especially in situations with limited resources and a rich morphology.

The architecture consists of three main components: (1) a frozen mBERT encoder that provides deep multilingual token representations; (2) adapter modules that introduce language-specific transformations between encoder layers; and (3) shared task heads for parsing and tagging that operate over the adapter-enhanced embeddings.

Language embeddings are learned during training by projecting URIEL typological features (Littell et al., 2017) with a multi-layer perceptron, following Üstün et al. (2020, 2022). This projection allows structurally sensitive adaptation without relying on fixed encodings, enabling UDapter to optimize language embeddings for parsing quality.

Adapter Modules UDapter uses residual bottleneck adapters inserted after each transformer layer, following the formulation in Houlsby et al. (2019). Each adapter transforms the hidden state $h \in \mathbb{R}^d$ as:

$$\text{Adapter}(h) = h + W_u f(\text{LN}(h)W_d) \tag{1}$$

Original Representation

```
# sent_id = 00099161_102
# text = Kadınlar gittikten sonra sizinkine veririm.
1 Kadınlar kadın ADJ NAdj Case=Nom|Number=Plur|Person=3 2 nsubj _ _
2 gittikten git VERB Verb Aspect=Perf|Case=Abl|Mood=Ind|Polarity=Pos|Tense=Past|VerbForm=Part 6 advcl _ _
3 sonra sonra ADP PCAbl _ 2 case _ _
4-5 sizinkine _ _ _ _ _ _ _ _
4 sizin siz PRON Pers Case=Gen|Number=Plur|Person=2|PronType=Prs 6 iobj _ _
5 kine ki ADP Rel Case=Dat|Number=Sing|Person=3 4 case _ _
6 veririm ver VERB Verb Aspect=Hab|Mood=Ind|Number=Sing|Person=1|Polarity=Pos|Tense=Pres 0 root _ SpaceAfter=No
7 . . PUNCT Punc _ 6 punct _ _
```

MSP-Adapted Representation

```
# sent_id = 00099161_102
# text = Kadınlar gittikten sonra sizinkine veririm.
1 Kadınlar kadın ADJ _ Case=Nom|Number=Plur|Person=3 2 nsubj _ _
2 gittikten git VERB _ Aspect=Perf|Case=Tps|Mood=Ind|Polarity=Pos|Tense=Past|VerbForm=Part 6 advcl _ _
3 sonra sonra ADP _ _ _ _ _ _
4-5 sizinkine _ _ _ _ _ _ _ _
4 sizin siz PRON _ Case=Gen;Dat|Number=Sing|Person=3|PronType=Prs 6 iobj _ _
5 kine ki ADP _ _ _ _ _ _
6 veririm ver VERB _ Aspect=Hab|Mood=Ind|Polarity=Pos|Tense=Pres 0 root _ _
6.1 _ _ PRON _ Case=Nom|Number=Sing|Person=1|PronType=Prs 6 nsubj _ _
7 . . PUNCT _ _ _ _ _ _
```

Figure 1: Data Formats

Here, LN denotes layer normalization, f is a non-linearity (typically ReLU or GELU), and $W_d \in \mathbb{R}^{d\times b}$, $W_u \in \mathbb{R}^{b\times d}$ are projection matrices defining the bottleneck structure. This configuration enables efficient language-specific adaptation while keeping the main encoder frozen.

Task Heads UDapter includes two task heads shared across languages. The Dependency Parsing Head uses a biaffine attention mechanism to predict syntactic arcs and labels. For each token pair, head and dependent projections are computed as:

$$r_i^{\text{head}} = \text{MLP}_{\text{head}}(h_i), \quad r_j^{\text{dep}} = \text{MLP}_{\text{dep}}(h_j)$$

The score of an arc from token i to token j is given by:

$$s(i,j) = r_i^{\text{head}\top} W_{\text{arc}} r_j^{\text{dep}} + U_{\text{arc}}^\top [r_i^{\text{head}}; r_j^{\text{dep}}] + b_{\text{arc}}$$

A separate biaffine classifier is used to assign dependency labels to each scored arc.

The Morphological Tagging Head follows a multi-label setup, predicting the value of each morphological attribute (e.g., Case, Number, Tense) independently. For each attribute f, a dedicated softmax layer is applied:

$$\hat{y}_i^{(f)} = \text{softmax}(W^{(f)} h_i + b^{(f)})$$

where $W^{(f)}, b^{(f)}$ are task-specific parameters. This factored approach allows the model to generalize better on rare tag combinations compared to predicting a concatenated tag string.

3.2 Content/Function Classifier (CF-BERT)

To identify functional nodes in a language-agnostic way, `bert-base-multilingual-cased` is fine-tuned on a binary token classification task. Functional nodes (e.g., AUX, DET, ADP) are excluded from the standard parsing graph, as they are assigned null heads in the MSP format and their contribution is represented through morphological features. The classifier computes:

$$p(y_i|x_i) = \text{softmax}(W_c h_i + b_c) \tag{2}$$

where h_i is the contextual embedding of token x_i from mBERT, and W_c, b_c are learned parameters.

Training data is constructed from the MSP shared task data by labeling tokens with null heads as functional and others as content. All languages are used jointly during training, resulting in a multilingually trained model that achieves high accuracy and enables reliable identification of functional nodes prior to dependency parsing.

4 Experimental Setup & Results

In this section, the multilingual parsing system's test set results, evaluation methods, and training configuration are shown. Key hyperparameters, implementation details, and necessary preparation steps for the MSP shared task format are explained. Three metrics—MSLAS, LAS, and morphological feature (Feats) F1—are used to compare the empirical performance across languages in the multilingual and monolingual setups.

4.1 Experimental Setup

The models are trained on an NVIDIA L40S GPU using the AllenNLP framework (Gardner et al., 2018). While the training set consists of uncovered Universal Dependencies treebanks, the test set is displayed in covered format[1]. Tokenization and segmentation are recovered during evaluation using UDPipe 2.0 (Straka, 2018) just during test time.

The `bert-base-multilingual-cased` model is used as the shared backbone, frozen throughout, with adapter modules and task-specific decoders trained on top. Input embeddings incorporate language-specific adapter representations using syntax, phonology, and phoneme inventory features. Morphological features are modeled with factored outputs using separate softmax layers for each attribute.

Dropout is applied at multiple levels: 0.15 in BERT adapters, 0.2 in word dropout, and 0.5 in decoders. Layer dropout and language embedding dropout are both set to 0.1. Language embeddings are 32-dimensional vectors learned from typological features. The batch size is dynamically adjusted using a maximum amount of 3200 tokens per batch. Training is performed for up to 80 epochs with early stopping and gradient clipping ($\|\nabla\| \leq 5$). Total training time was approximately 12.3 hours, with peak GPU memory usage reaching 28.3GB. No additional hyperparameter tuning was performed; the configurations were adopted directly from the original UDapter work (Üstün et al., 2020, 2022).

4.2 Results & Discussion

Tables 1a–1c report performance on the covered test set using MSLAS, LAS, and Feats F1 metrics. As the test data omits structural and morphological annotations, UDPipe is used during evaluation to recover segmentation and token boundaries only. This ensures compatibility with the uncovered training format while allowing test-time evaluation against the shared task metrics.

The submitted system corresponds to the multilingual configuration, where all languages are trained jointly with shared parameters and language-specific adapters. For comparison, a monolingual baseline is included, consisting of separately trained models for each language without cross-lingual transfer.

Multilingual Superiority The multilingual model consistently outperforms its monolingual counterparts across all metrics, demonstrating the effectiveness of cross-lingual transfer in morphosyntactic parsing. On average, it yields a relative improvement of +6.20 in MSLAS F1, +7.45 in LAS F1, and +7.21 in Feats F1 (Tables 1a–1c). These gains are particularly pronounced in English (+7.61 MSLAS, +7.19 LAS, +4.40 Feats), Italian (+7.58, +7.66, +7.19), and Serbian (+7.69, +6.82, +4.06), suggesting that typological proximity and morphological richness play key roles in enhancing multilingual adapter-based learning.

Such improvements also indicate that the shared parameter space of UDapter—augmented with language-specific adapters and multitask supervision—facilitates better generalization in low- to medium-resource settings. Even for languages with complex morphology and flexible word order, such as Turkish, notable gains are achieved (+7.50 MSLAS, +7.50 LAS, +4.36 Feats), confirming the model's robustness within MSP's revised structural paradigm.

Abstract Node Omission A key limitation of the current architecture is its inability to model abstract nodes, despite their presence in the training data. These nodes represent syntactic elements with no surface realization—such as dropped subjects or objects—but still function as content nodes with syntactic heads and morphological features. For example, in the sentence provided in Figure 1, the subject pronoun *ben* is not expressed in the surface form but is represented by an abstract node with ID 6.1. This node functions syntactically as the subject of the verb *veririm* and carries person and number features.

Since the test set is provided in covered format, abstract nodes must have been generated before parsing, which requires nontrivial modifications to standard pipelines. As our current system lacks this capability, recall is penalized in languages where such structures are common. Turkish is particularly affected due to its reliance on pro-drop constructions and agglutinative morphology. As shown in Table 2, Turkish exhibits the highest proportion of abstract nodes (13.45%), contributing to its relatively lower evaluation gains. Omitting such content nodes impacts both dependency arc and

[1] The "covered" version, as referred to throughout the paper, includes only the # text = "..." line for each sentence in the data files, with all remaining annotations removed. The "uncovered" version of the data can be seen in the example provided in Figure 1.

System	AVG	cz	en	he	it	pl	pt	sr	sv	tr
Monolingual	55.08	70.30	52.11	37.62	49.95	54.47	63.53	68.35	58.56	40.85
Multilingual	**61.28**	**73.02**	**59.72**	**43.44**	**57.53**	**60.40**	**68.07**	**76.04**	**64.99**	**48.35**
Diff	↑6.20	↑2.72	↑7.61	↑5.82	↑7.58	↑5.93	↑4.54	↑7.69	↑6.43	↑7.50

(a) MSLAS F1 scores

System	AVG	cz	en	he	it	pl	pt	sr	sv	tr
Monolingual	58.86	74.87	58.60	43.53	53.98	59.60	69.53	73.79	63.39	45.20
Multilingual	**66.31**	**77.57**	**65.79**	**49.68**	**61.64**	**65.62**	**73.97**	**80.61**	**69.67**	**52.70**
Diff	↑7.45	↑2.70	↑7.19	↑6.15	↑7.66	↑6.02	↑4.44	↑6.82	↑6.28	↑7.50

(b) LAS F1 scores

System	AVG	cz	en	he	it	pl	pt	sr	sv	tr
Monolingual	73.29	85.41	76.35	63.48	69.54	73.19	79.88	85.47	81.09	71.15
Multilingual	**80.50**	**87.22**	**80.75**	**68.94**	**76.73**	**78.46**	**83.12**	**89.53**	**84.56**	**75.51**
Diff	↑7.21	↑1.81	↑4.40	↑5.46	↑7.19	↑5.27	↑3.24	↑4.06	↑3.47	↑4.36

(c) Feats F1 scores

Table 1: Test set performance per language using covered CoNLL-U and predicted content/function labels. Each subtable reports one metric.

morphological feature prediction.

Lang.	Abs.	Total	Rate (%)
Czech	2441	87857	2.78
English	30	7732	0.39
Hebrew	171	5717	2.99
Italian	161	9956	1.62
Polish	1238	34310	3.61
Portuguese	915	32625	2.80
Serbian	45	11466	0.39
Swedish	14	20128	0.07
Turkish	**1553**	**11544**	**13.45**

Table 2: Rates of abstract nodes per language in the test sets. Turkish shows the highest omission rate.

Functional Node Filtering with CF-BERT Prior to parsing, functional nodes are filtered using a dedicated content/function classifier, CF-BERT. This preprocessing step is essential for the MSP task, where functional nodes—such as auxiliaries, conjunctions, and determiners—are excluded from the dependency graph. To align with the MSP annotation scheme, CF-BERT is trained on the uncovered training split and validated on the uncovered development split. That is, whether a token had a head annotation in the CoNLL-U file was used as the class label in our content/function classification task. To maintain direct supervision and avoid dependence on POS tags or language-specific heuristics, we used only surface forms of words as input features.

As shown in Table 3, the classifier achieves excellent results across all metrics, maintaining above 99% accuracy, precision, recall, and F1 score. These scores confirm the classifier's effectiveness in consistently identifying and filtering out non-content elements. With high-confidence functional filtering in place, the UDapter parser receives clean, content-bearing structures, enhancing both arc prediction and cross-lingual generalization.

Metric	Score
Accuracy	99.57%
Precision	99.04%
Recall	99.04%
F1 Score	99.04%

Table 3: CF-BERT performance on functional node classification using the uncovered training (train) and development (dev) splits of the MSP dataset.

Structural Universality through Multitask Learning UDapter's multitask design aligns closely with the shared task's goal of simultaneously modeling syntactic dependencies and morphological features. The system predicts both arc structures and token-level features in a unified framework, allowing complementary signals to guide representation learning. By using factored morphological decoders and typology-aware adapters, the model generalizes well to the structural diversity present in the 9 target languages.

As seen in Tables 1b and 1c, UDapter achieves strong performance across both syntactic and morphological dimensions. LAS gains demonstrate robust parsing capabilities, with improvements such as +7.66 in Italian and +7.19 in English, while Feats F1 results such as +7.19 in Italian, +5.27 in Polish, +4.36 in Turkish show accurate modeling of rich morphological systems. The factored decoding architecture enables efficient learning over sparse feature combinations, particularly beneficial in morphologically complex settings. These results affirm that structurally aware multitask systems can offer a linguistically grounded and scalable solution to cross-lingual morphosyntactic parsing.

5 Conclusion

This work presents the first successful adaptation of the UDapter model to the UniDive MSP Shared Task, which challenges traditional parsing by introducing structurally flexible and typologically informed dependency representations. The task format—featuring abstract nodes and functional node eliminations—necessitates substantial revisions to conventional parsing pipelines.

To address these challenges, the proposed system combines a BERT-based functional node classifier (CF-BERT) with UDapter's multilingual adapter architecture and factored multitask decoders. CF-BERT aligns training and test conditions by filtering out non-content elements with near-perfect accuracy, allowing the parser to focus exclusively on content-bearing structures. This setup enhances both syntactic and morphological prediction under cross-lingual supervision.

Experimental results show that the multilingual system consistently outperforms monolingual baselines across MSLAS, LAS, and Feats F1 metrics, with particularly strong gains in morphologically rich languages like Turkish and typologically adjacent languages like Italian and Serbian. The results affirm the system's core strengths: typology-aware representation, functional node filtering, and multitask structural learning. Future improvements may focus on integrating abstract node generation to better capture pro-drop phenomena and further enhance recall in structurally underspecified contexts.

Limitations

Despite the fact that it works well across languages in the MSP Shared Task, some limitations affect the scope and architecture of the proposed system.

Although abstract nodes are present in the training data, the current model architecture does not learn or predict them. These nodes typically correspond to syntactic elements that are not explicit in surface form, such as dropped subjects or pronouns in pro-drop languages like Turkish. Since the test data is covered, special mechanisms are required to incorporate abstract nodes into parsing and decoding. Designing models that can effectively handle such structures remains an open direction for future research.

Second, the content/function classifier is only used as a preprocessing step and is not integrated into the multitask learning process. A more unified framework may be able to jointly learn this classification in addition to parsing and labeling, which could improve task interaction.

Additionally, the system just slightly alters its default hyperparameters. Language-specific typological embeddings, dropout rates, and adaption sizes are all fixed. Custom configurations can further enhance performance, particularly in environments with complex morphology or constrained resources.

Lastly, the experiments only use the 9 languages that were included in the challenge. The robustness and universality of the model would be supported by further evaluation on datasets with greater diversity of typologies.

These limitations open up a number of paths for future study, such as deeper task integration, structural modeling of abstract nodes, and more comprehensive multilingual testing.

Acknowledgments

We would like to sincerely thank the Department of Technology and Innovation at Kariyer.net for providing the GPU resources that made this research possible. We express our gratitude to the UniDive consortium and UniDive MSP Shared work organizers, as well as the data providers, for their invaluable contributions to the creation of the dataset and shared work.

References

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, New York City. Association for Computational Linguistics.

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. *Preprint*, arXiv:1911.02116.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Omer Goldman, Leonie Weissweiler, Kutay Acar, Diego Alves, Arianna Bienati, Gülşen Eryiğit, Adriana Pagano, Ludovica Pannitto, Tanja Samardžić, Luigi Talamo, Alina Wróblewska, Daniel Zeman, Joakim Nivre, and Reut Tsarfaty. 2025. Findings of the UniDive 2025 shared task on multilingual morpho-syntactic parsing. In *Proceedings of The UniDive 2025 shared task on multilingual morpho-syntactic parsing*.

Johan Hall, Jens Nilsson, Joakim Nivre, Gülşen Eryiğit, Beáta Megyesi, Mattias Nilsson, and Markus Saers. 2007. Single malt or blended? a study in multilingual parser optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 933–939, Prague, Czech Republic. Association for Computational Linguistics.

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. *Preprint*, arXiv:1902.00751.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

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.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. Uriel and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160, Nancy, France.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, Prague, Czech Republic. Association for Computational Linguistics.

Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 221–225, New York City. Association for Computational Linguistics.

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.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2022. UDapter: Typology-based language adapters for multilingual dependency parsing and sequence labeling. *Computational Linguistics*, 48(3):555–592.

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.