UNIDIVE 2025

Workshop of The UniDive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing

Proceedings of the Workshop

August 26, 2025

The UNIDIVE organizers gratefully acknowledge the support from the following sponsors.

Sponsored by

UniDive

As part of

SYNTAX
FEST
Ljubljana
August 26-29 2025

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-320-3

Organizing Committee

Organizers

Omer Goldman, Bar-Ilan University
Leonie Weissweiler, Uppsala University
Joakim Nivre, Uppsala University
Reut Tsarfay, Bar-Ilan University

Publication Chair

Omer Goldman, Bar-Ilan University

Local SyntaxFest 2025 Organizing Committee

Kaja Dobrovoljc, University of Ljubljana, SDJT
Špela Arhar Holdt, University of Ljubljana
Luka Terčon, University of Ljubljana
Marko Robnik-Šikonja, University of Ljubljana
Matej Klemen, University of Ljubljana
Sara Kos, University of Ljubljana
Timotej Knez, University of Ljubljana, SDJT
Tinca Lukan, University of Ljubljana, SDJT

Program Committee

Program Chairs

Omer Goldman, Bar Ilan University
Leonie Weissweiler, Uppsala University

Table of Contents

Findings of The UniDive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing

Omer Goldman[1], Leonie Weissweiler[2], Kutay Acar[3], Diego Alves[4], Anna Bączkowska[5], Gülşen Eryiğit[3], Lenka Krippnerová[6], Adriana Pagano[7], Tanja Samardžić[8], Luigi Talamo[4], Alina Wróblewska[9], Daniel Zeman[6], Joakim Nivre[2], and Reut Tsarfaty[1]

[1]Bar-Ilan University [2]Uppsala University [3]Istanbul Technical University
[4]Saarland University [5]University of Gdansk [6]Charles University
[7]Federal University of Minas Gerais [8]University of Zurich [9]Polish Academy of Sciences
omer.goldman@gmail.com, leonie.weissweiler@lingfil.uu.se, reut.tsarfaty@biu.ac.il

Abstract

This paper details the findings of the 2025 UniDive shared task on multilingual morpho-syntactic parsing. It introduces a new representation in which morphology and syntax are modelled jointly to form dependency trees of contentful elements, each characterized by features determined by grammatical words and morphemes. This schema allows bypassing the theoretical debate over the definition of "words" and it encourages the development of parsers for typologically diverse languages. The data for the task, spanning 9 languages, was annotated based on existing Universal Dependencies (UD) treebanks that were adapted to the new format. We accompany the data with a new metric, MSLAS, which combines syntactic LAS with F1 over grammatical features. The task received two submissions, which, together with three baselines, give a detailed view on the ability of multi-task encoder models to cope with the task at hand. The best performing system, UM, achieved 78.7 MSLAS macro-averaged over all languages, improving by 31.4 points over the few-shot prompting baseline.

1 Introduction

Syntactic and morphological tasks, such as parsing (Sakai, 1961; Zeman et al., 2018) and linearization (Filippova and Strube, 2007; Shimorina et al., 2021), analysis (Koskenniemi, 1983; McCarthy et al., 2019) and inflection (Durrett and DeNero, 2013; Goldman et al., 2023), have a long history in NLP research. Collectively, these tasks aim to provide structured representations of free text to facilitate further research and applications. However, the distinction between morphology and syntax, and hence the definitions of these tasks, rely on the definition of a "word" (Dixon and Aikhenvald, 2002) – a unit that is notoriously ill-defined from a cross-lingual perspective (Haspelmath, 2011).

This shared task draws on previous works which attempt to avoid relying on words, in either syntax (Bārzdiņš et al., 2007; Nivre et al., 2022) or morphology (Goldman and Tsarfaty, 2022), and presents a new representation that models morphosyntax in a unified and harmonized fashion. The representation used here closely resembles dependency trees from Universal Dependencies (UD; de Marneffe et al., 2021), but instead of the distinction between words and morphemes, it adopts the distinction between *content* and *function*. In this structure, every sentence is represented by a dependency tree whose nodes are content-bearing elements. On the other hand, function elements, words and morphemes, are represented as grammatical features on the nodes, in a manner similar to morphological features in UD.

The shared task includes data in 9 languages: Czech, English, Hebrew, Italian, Polish, Brazilian Portuguese, Serbian, Swedish, and Turkish, with several thousands of sentences as training data for each. The data was converted from UD dependency trees in a semi-automatic fashion. Submitted systems were tested on a held-out test set for which participants got only the raw text as input. The systems' predictions were evaluated using three metrics. LAS, as defined by Nivre et al. (2004), was used to measure the systems' success in connecting the nodes to their correct parent with the correct relation. F1 over morpho-syntactic features measured the systems' ability to correctly characterize the functions relating to each node. But as the main metric, we defined MSLAS, which combines both metrics to measure overall success in modelling both node relations and node content.

This report includes the results of five systems, three baselines and two submitted systems. Both submitted systems and one of the baselines utilize multi-task training, where several classifiers are trained on top of a frozen encoder-only model such

Proceedings of The UniDive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing, pages 1–18
August 26, 2025 ©2025 Association for Computational Linguistics

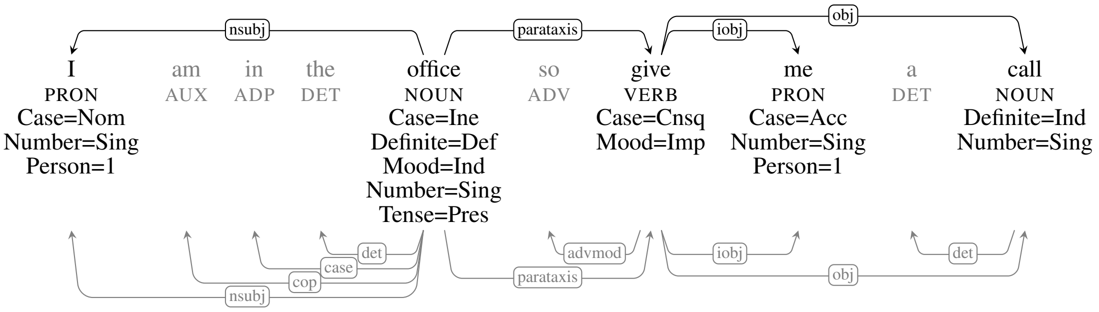

Figure 1: An example of a morphosyntactic tree above the sentence, with the full UD tree below for reference. Function words do not participate in the morphosyntactic tree and are coloured grey.

as BERT (Devlin et al., 2019) or XLM-R (Conneau et al., 2020). The systems differ mostly in the way they incorporate the prediction of contentfulness into the system, and whether they were trained jointly or separately across languages. In addition, two baselines prompted Gemini 2.5 (Comanici et al., 2025) in a few-shot setting to output the entire prediction as a string.

The best performing system, UM (Inostroza Améstica et al., 2025), achieved 78.7 MSLAS over the covered test set macro-averaged over all languages. This score is a 31.4-point improvement over the best baseline. Although impressive, the results show that there is still some headroom, for training better models to excel in the task. Specifically, non-Indo-European languages, like Turkish and Hebrew, seem to pose a greater challenge to the submitted systems that are based on dependency parsing models.

All in all, this shared task introduces a more holistic representation that takes into account the typological variety in word definitions across languages, and shows that predicting this structure poses a challenge for parsing technology and to large language models alike.

2 Task Description

The concept of "word" is known to have a highly disputed definition, ever since (at least) the formation of modern linguistics as a field (Krámský, 1969; Juilland and Roceric, 1972). Despite that, words have had a crucial role in delineating syntax and morphology, both from a theoretical (Dixon and Aikhenvald, 2002) and practical (de Marneffe et al., 2021; Batsuren et al., 2022) standpoint. The result in UD is that words are defined inconsistently across languages, so the ability to compare structures across languages is severely hindered. On the other hand, the mere attempt to cross-lingually define words entails a tokenization process that is unnatural to some languages that differ greatly from the prototypical Western language.

Morpho-syntactic parsing is designed to bypass the issue by modelling syntax and morphology in tandem, getting rid of the necessity to define words in order to separate the two. In this task, systems are required to predict a morpho-syntactic dependency tree for each sentence, where nodes are contentful elements. These nodes include lexemes from open-class parts of speech, such as nouns and verbs, and their arguments. Grammatical, or functional, elements do not appear as nodes in the tree, even if they are written as independent words. Instead, they contribute morpho-syntactic features that characterize the content nodes, akin to morphological features in UD.

All predicates and arguments are included in the morpho-syntactic tree, including arguments in the form of pronouns, clitics and agreement morphemes.[1] On the other hand, elements that signify relations between content nodes are deemed function nodes, as well as elements that specify some grammatical additional meaning to content nodes.

The result is a dataset in which sentences in different languages are structured much more similarly, owing their differences to structural linguistic variations between languages, and not to variations in orthography or grammatical tradition. Systems solving morpho-syntactic parsing must then detect contentful elements in a similar fashion across all languages, without the benefit of a more natural tokenization process given to languages that are more similar to English and a few other Western languages.

Figure 1 has an example tree. It shows several function words in gray that are not part of the

[1] See Zwicky and Pullum (1983) for a discussion on the wordhood of these elements.

Lang	Train	Dev	Test	Annotator	Treebank
CS	10,000	4,681	5,885	Daniel Zeman & Lenka Krippnerová	PDT & PUD
EN	2,576	472	492	Omer Goldman	EWT
HE	2,267	374	253	Omer Goldman	HTB
IT	7,773	537	461	Luigi Talamo & Arianna Bienati & Ludovica Pannitto	All UD treebanks
PL	9,914	2,180	3,180	Alina Wróblewska & Anna Bączkowska	PDB & PUD
PT	5,591	806	1,596	Diego Alves & Adriana Pagano	Porttinari
SR	3,312	536	520	Tanja Samardžić	SET
SV	4,128	492	2,168	Joakim Nivre & Victor Norrman	Talbanken & PUD
TR	3,409	1,081	1,082	Kutay Acar & Gülşen Eryiğit	IMST

Table 1: The number of sentences in each split for each language.

morpho-syntactic tree, but contribute features to the content nodes, like the conjunction *so* and the preposition *in* that contribute a Case feature, and the copula *am* that contributes verbal features to the nominal predicate *office*.

2.1 Data Annotation Process

The data for this task was semi-automatically converted from UD treebanks. A conversion script was written for every language, based on some shared infrastructure. The scripts used POS tags and dependency relations to detect function words and used the languages' grammars to decide on the morpho-syntactic features of the nodes given the function words and morphological features in UD. Manual decisions were made, for example in cases of ambiguity, like English *would* that can denote either a conditional mood or past prospective, or in cases of categories that are not strictly function or content, like adverbs.

The annotation of UD treebanks into the morpho-syntactic schema followed several principles. First and foremost, the resulting data structure had to be independent of word definitions. Specifically, all predicates and arguments should have corresponding tree nodes.[2] For practical reasons, the data was annotated in a way that deviates as little as possible from the existing UD data. We only replaced the morphological features with morpho-syntactic features, removed function words from the tree, and occasionally added *abstract nodes* for unrealized arguments. In other words, we did not alter the heads of the nodes nor the arc labels. The full annotation guidelines are given in Appendix A.

All in all, we ended up with about 70k annotated sentences in 9 languages. Of which, about 45k were used for training, and about 12.5k constitute a held-out test whose gold trees were not revealed to the participants. The data was split according to the splits in the UD treebanks it is based on. The statistics of the entire dataset are given in Table 1.

2.2 Evaluation

Systems were evaluated using three metrics: *LAS*, *Feats-F1*, and *MSLAS*, with the latter being the main metric that combines the two others others.

Since the input for this task is raw text, systems may predict a different number of nodes compared to the ground truth. And because all metrics compare nodes to one another, an alignment mechanism is needed between the prediction and the gold trees. Nodes are then considered correctly predicted under any of the metrics only if they are aligned with a node in the gold tree and have the same dependency arc, morpho-syntactic features, or both, depending on the metric. The alignment of content nodes to gold nodes is done sequentially for each sentence twice, from right to left or left to right, then metrics are computed for each alignment, and the better score is taken for that sentence.

The evaluation script is based on that of Zeman et al. (2018), and can be found at `https://github.com/UniDive-MSP/MSP-shared-task`.

Labelled Attachment Score (LAS) is the standard evaluation metric in dependency parsing. It is the percentage of predicted nodes that are assigned the same parent and relation type (arc label) as the corresponding gold node. In our task, LAS is calculated only over content nodes. Note that we did not take subtypes into account. For example, an arc labelled `acl:relcl` was considered correct as long as the corresponding gold arc had `acl` as its main type.

Features F1 Score (Feats-F1) is a standard metric in morphological analysis. It is the F1 score when comparing the predicted set of features of a specific node to the features of its aligned node in the gold tree. We applied this metric to all morpho-syntactic features assigned to a node, not only to the morphological features.

[2] This also bypasses the debate on whether pronouns are a functional feature bundle or contentful elements with definite semantic meaning.

Morphosyntactic Labelled Attachment Score (MSLAS) is the main metric of the task that combines both other metrics to evaluate the systems' trees as well as the quality of the characterization of each node. This metric averages the per-node Feats-F1 score only for nodes that are considered correct according to the LAS metric.

3 Languages

The data for each language was prepared by individuals or teams who are speakers of that language (see Table 1). Below is a short description of the languages included in the shared task.

3.1 Czech

Czech is a West Slavic language with rich fusional morphology. Nouns, adjectives, pronouns, determiners and numerals inflect for up to 7 case forms. Morphology of finite verbs cross-references the person and number of the subject; participles cross-reference gender and number. Consequently, subject nominals can be dropped. Some verb forms (past tense, future tense, conditional, and passive) are composed periphrastically using various forms of the auxiliary *být* 'to be'. Modal verbs are treated as main verbs in UD, and that approach is kept in the shared task. The reflexive clitics *se, si* are used, besides marking true reflexive arguments, also in so-called reflexive passive construction (and, with certain verbs, as a particle modifying the meaning of the verb).

The shared task data is based on the Prague Dependency Treebank (PDT; Hajič et al., 2020),[3] automatically modified to the shared task format using rule-based heuristics implemented in the Udapi framework[4] (Popel et al., 2017). In the first stage, periphrastic verb forms and copular constructions were identified and converted into features, based in part on Krippnerová and Zeman (2025).[5]

In the second stage, the value of the `Case` feature was determined for nominals. First, by determining the morphological case of some non-straightforward cases: uninflected loanwords, abbreviations, and numeral-modified nouns. Then, we combined the morphological case with prepositions to generate the morphosyntactic `Case` value that reflects the most salient meaning of that combination. For example, *za* 'behind' + `Case=Ins` would result in `Case=Pst` (postessive), *za* + `Case=Acc` in `Case=Psl` (postlative), and *za* 'under' + `Case=Gen` would yield `Case=Der` (durative). In an analogy, the `Case` feature was also used to encode meanings of subordinating and even coordinating conjunctions. See Appendix A.1 for the full label inventory; out of them, 79 values are attested in the Czech data.

Finally, abstract nodes were created to represent dropped subject pronouns in finite clauses; their features (`Number`, `Person` and/or `Gender`) were taken from the verb. If an overt subject was present in the sentence, no abstract node was created, but the agreement features were still removed from the verb.[6]

3.2 English

While English is often described as morphologically "poor", its morpho-syntax is considerably rich. Verbs, whose features now include information from auxiliaries and particles, are inflected to 3 tenses, 4 aspects, and multiple moods. Nouns are inflected for number and definiteness, as well as for a wide array of cases, mostly using prepositions. And adjectives are also inflected for degree.

Although grammatical functions are usually expressed by concatenative means, some non-concatenative operations are also employed by English: ablauts for past inflections and word/morpheme order for interrogative mood.

In terms of marginally grammatical structures, the treebank we started with, EWT (Silveira et al., 2014), considers modal verbs are inflections of the main verb (expressed using the `Mood` feature), but *going to*, *used to* and similar constructions are considered semantic compositions of two verbs rather than a grammatical tense marking.

3.3 Italian

Italian presents quite rich morphology, especially in the verbal system, while being primarily tense-based, aspect and mood also play crucial roles. However, while tense is morphologically marked or explicit through construction with auxiliaries, aspect is not fully grammaticalized and has often

[3] We only took documents that had tectogrammatical annotation in the original treebank, as their UD conversion already has abstract nodes for some dropped pronouns. We excluded all sentences containing the `orphan` relation because they have other types of abstract nodes, not expected in the shared task.

[4] `https://udapi.github.io/`

[5] This tool is ready to process any Slavic language currently covered in UD, including Polish and Serbian; however, in the present shared task, it was used to prepare the Czech data only.

[6] The features were not copied to the overt subject (i.e., `Person=3` can disappear completely if the subject is a noun or a subject clause).

to be inferred from lexical choices or contextual cues. Nonetheless, in the indicative mood, tenses bear a tendential association with aspect, which we implemented in the task data.

The Italian data was selected by randomly sampling 10K sentences from all released Italian UD treebanks (Bosco et al., 2013; Sanguinetti et al., 2018; Alfieri and Tamburini, 2016; Zeman et al., 2017b). Sentences containing `parataxis`, `orphan`, `dep` or `discourse` relations were excluded, as they introduced trees too difficult to process automatically for the sake of the task. The resulting data is composed of 8771 sentences. Each dependency tree was traversed via depth first search (DFS), yielding head tokens and their non-content children. Specialized modules handled the information depending on the head UPOS.

Adpositions, subordinating conjunctions and coordinating conjunctions are mapped by lemmas to the `Case` feature. Values were assigned based on the lemma's meaning retrieved from a dictionary. If polysemous, the value was assigned deterministically to the most 'basic' or 'shared' meaning across languages, prioritizing spatial and temporal meanings. This decision is made according to the literature, when available (e.g., Luraghi, 2009), otherwise it relies on the intuition of a proficient speaker.

3.4 Hebrew

As a Semitic language, Hebrew grammar is characterized by the extensive use of ablauts for verbal inflection and the assignment of lexical meaning to consonantal roots. The Hebrew verbal paradigm is somewhat limited, with only indicative and imperative moods and one periphrastic aspect, but it has 7 inflectional cases and many irregulars. Many prepositions and conjunctions are fused onto the following word, possessive pronouns are sometimes fused onto the previous words, and some other prepositions are inflected when applied to personal pronouns.

The segmentation strategy taken in HTB (Sade et al., 2018) made its conversion to the morpho-syntactic schema relatively straightforward. Most fused elements were segmented from their parents so `Case` were determined based on the table in Appendix A.1 and applied to the heads. The Hebrew verbal system means that periphrastic inflections are extremely rare, although they had to be disambiguated manually. Lastly, nominal and adjectival predicates were harmonized in structure regardless of whether a copula exists or not.

3.5 Portuguese

Portuguese, like Italian, has a rich morphological system, particularly in its verb forms. To address this, rule-based adaptations were implemented for the five auxiliary verbs in Portuguese and their combinations, taking into account tense, mood, and aspect.

Portuguese is also distinctive for having two copulas *ser* and *estar*, which differ in aspect: *ser* typically marks stative situations, while *estar* conveys more dynamic or temporary states. The distinction between them was marked using the `Aspect` feature of their nominal head.

The Brazilian Portuguese corpus for the shared task was derived from the Porttinari corpus (Pardo et al., 2021; Duran et al., 2023), incorporating adaptations based on the changes proposed for the English corpus, while accounting for Portuguese-specific characteristics.

A set of features was defined for prepositions, adverbs, conjunctions, and fixed expressions. Special attention was given to avoid mislabelling homographic forms, such as *se*, which can function either as a clitic, as a pronoun or as a subordinate conditional conjunction. In addition, some manual decisions were made when dealing with degree markers of adjectives.

3.6 Polish

Analogous to Czech, Polish is a highly inflectional and fusional West Slavic language, characterised by the possibility of dropping subject nominals and by the use of analytical verb forms in the past tense, future tense, conditional, reflexive, and passive constructions. Modal verbs, inherently impersonal verb forms, and predicative words are treated as main verbs. A distinctive feature of Polish morphosyntax is the phenomenon called *mobile inflection*: auxiliary morphemes may detach from participles and move to another syntactically licensed position, e.g., attached to conjunctions or pronouns.

The shared task data is derived from the Polish PDB-UD and PUD-PL treebanks (Wróblewska, 2018), both automatically converted from the Polish Dependency Bank (Wróblewska, 2014). We excluded sentences with `orphan` relation that were proven too difficult for automatic annotation. The assignment of morpho-syntactic features then followed a rule-based procedure. Simple structural rules are applied to assign features to content words based on the subtrees they head. In addition, `Case` values were assigned based on a predefined repertoire (see Table 4) to all nouns, their adjectival dependents that show morphological agreement, and their conjuncts. Verbs were also assigned features based on suboridnators and other markers that modified them.

3.7 Serbian

Like the other Slavic languages, Serbian grammar is characterized by a preference for periphrastic verb forms. Of Serbian's seven tenses, and four moods, indicative present tense and imperative mood are the only frequently used inflections that do not require auxiliary verbs. Nouns inflect to 7 morphological cases with distinct singular vs. plural forms for most of the cases. Together with a wide array of prepositions, they form many more morpho-syntactic cases. Serbian has a relatively free word order, with many clitics, i.e., auxiliaries and some pronouns, tending to occupy the second position in the sentence in a fixed relative order.

The data for the shared task is based on the SET treebank (Batanović et al., 2023; Samardžić et al., 2017). The conversion process largely followed the logic set in the other languages, incorporating auxiliaries, conjunctions, and prepositions into features on their contentful parents. The lack of articles in Serbian meant that determiners were treated solely as content nodes.

3.8 Swedish

Swedish is a moderately inflected language belonging to the North Germanic branch of the Indo-European family. The case system for nouns has been reduced to two cases, nominative (which subsumes the old nominative, accusative and dative) and genitive, while the pronominal system still distinguishes three cases (nominative, accusative/dative, and genitive). The verbal inflection system has been simplified by dropping number and person agreement (except for past participles) and subjunctive mood (except for a few frequent verbs). Unusual features include a suffixed definite article (which means that all nouns are inflected for number, definiteness and case) and two passive constructions, one inflectional and one periphrastic.

The Swedish data sets are based on the UD treebanks Swedish-Talbanken (Einarsson, 1976; train, dev and test) and Swedish-PUD (Zeman et al., 2017a; only test). The conversion is based on the English conversion script, which was adapted to Swedish and complemented by a number of language-specific post-processing steps. The conversion has gone through a number of iterations involving a combination of automatic consistency checks and manual spot checks.

All adpositions, subordinating conjunctions and coordinating conjunctions have been mapped deterministically to the most frequent semantic category, with no attempt to disambiguate polysemous expressions. Since Swedish is not a pro-drop language, abstract nodes have been inserted only in cases where the head of an adposition or conjunction is elided.

3.9 Turkish

Turkish is the most agglutinative language in our selection. Morphemes are added to verbs, and occasionally to nouns and adjectives, to express tense and a wide array of compounding aspects and moods. Nouns inflected for case and possession, and subordinated verbs are inflected to convey their relation with their parent clause.

In order to convert the original IMST-UD treebank (Sulubacak and Eryiğit, 2018) to the shared task format, we manually annotated some functional categories (i.e., adverbials, adpositions, conjunctions, determiners, and auxiliaries) with the morphological features they should contribute to their contentful heads. Overall, we introduced 23 morphological features for adpositions, 10 for conjunctions, 10 for adverbials, 4 for determiners, and 1 for auxiliaries. These new morpho-syntactic features were then systematically transferred upwards to contentful nodes via recursive syntactic tree traversal, ensuring accurate representation of implicit grammatical structures. Abstract nodes were explicitly introduced to represent dropped pronouns, frequent in Turkish.

Appendix B provides a sample sentence annotated in UD and in the morpho-syntactic schema. It illustrates double-case marking on the particle *-ki* in *Karşınızdakine* (nodes 1 and 2) and multiple abstract nodes for dropped pronouns (nodes 1.1, 5.1, 9.1, and 11.1).

4 Baseline Systems

We provide three baseline systems, of which one is a fine-tuned mBERT model and two are on the basis of prompting Gemini 2.5.

4.1 MC: Fine-tuning Baseline

The fine-tuning baseline uses the MaChAmp toolkit (van der Goot et al., 2021), a multi-task learning library optimised specifically for handling ConLL-U data. MaChAmp encodes the data using mBERT (Devlin et al., 2019), and then trains a separate decoding head for each task. We do not predict separately which words are content words, but rather use the SEQ head for the prediction of the morphosyntactic features (as would be standard for the prediction of regular UD morphological features) and set words with no features as function words. In addition, the DEPENDENCY head predicts dependencies between all words. This resulted in overprediction of dependency arcs for function nodes, which does not affect the scores as they are ignored in the evaluation. We trained separate models per language for 20 epochs with early stopping, and applied postprocessing to ensure that the node determined by the dependency parser as the root, if it had not been assigned any morphosyntactic features, is assigned the empty feature set to mark it as a content node ('|').

4.2 Prompting Baselines

In addition, we provided two prompting-based baselines where no model was trained. Instead, we gave Gemini 2.5 Pro the annotation guidelines from Appendix A (without the cases table from Appendix A.1) together with 3 example parse trees, randomly chosen from the dev sets for each test example. The model was then asked to output, as a text string, the correct parse tree for that test example. Minor post processing was done to correct some formatting issues, like replacing whitespaces with tabs. But for the most part, when the model output a tree in the wrong format, it received zero credit.

The difference between the two baselines is in the source of the examples given with the input sentence:

- **Cross** is a cross-lingual baseline, where the examples for each test input were taken strictly from three other languages.
- **Mono** is where the examples for each test input were taken solely from the dev set of the same language.

System		MSLAS	LAS	Feats
baselines	MC	33.0	36.1	52.3
	Cross	36.7	51.2	50.6
	Mono	47.3	55.4	64.2
submissions	ITU	61.3	66.4	80.5
	UM	**78.7**	**80.1**	**90.3**

Table 2: Scores for each of the systems, averaged over all 9 languages.

5 Submitted Systems

5.1 UM: University of Melbourne

The system submitted by the University of Melbourne (Inostroza Améstica et al., 2025) is based on an XLM-RoBERTa encoder augmented with character embeddings (Akbik et al., 2018), which is shared for all languages. This is combined with three specialised decoders, where the first classifies words into either content or function, and the other two operate only on the function words. The content words identification system is a BiLSTM combined with a linear layer, while the morphosyntactic feature decoder is a single multi-label classification layer, which notably predicts each feature value separately. The parsing decoder uses multilayer perceptrons for arc and relation prediction with biaffine attention mechanisms (Dozat and Manning, 2017).

5.2 ITU: Istanbul Technical University

The system submitted by Istanbul Technical University (Acar and Eryiğit, 2025) is similarly based on mBERT (Devlin et al., 2019). Specifically, it uses UDapter (Üstün et al., 2020), which introduces adapter modules for language-specific transformations between encoder layers. These adapter modules are informed by language embeddings derived from the URIEL database (Littell et al., 2017). Prediction is handled by a dependency parsing head with biaffine attention, and a morphological tagging head which predicts the value of each feature separately. Classification into content and function words is handled by a separate model, a fine-tuned instance of mBERT.

6 Results

We present overall evaluation results in Table 2. Out of the two submitted systems, UM achieved the highest MSLAS on the test set, macro-averaged

Lg.	Metric	Baseline			UM	ITU
		Mono	Cross	MC		
CS	MSLAS	32.9	43.9	34.7	87.1	73.0
	LAS	39.5	58.3	36.7	88.0	77.6
	Feats	48.0	58.8	53.0	95.2	87.2
EN	MSLAS	51.4	43.1	37.7	83.8	59.7
	LAS	59.4	58.5	41.2	85.1	65.8
	Feats	69.0	55.3	53.9	94.9	80.7
IT	MSLAS	46.5	3.5	33.6	73.0	57.6
	LAS	52.8	8.1	35.1	73.7	61.6
	Feats	62.7	15.3	54.1	84.7	76.9
HE	MSLAS	56.3	37.3	22.4	68.7	43.4
	LAS	65.0	53.1	26.4	71.4	49.7
	Feats	71.9	54.0	43.3	83.4	68.9
PT	MSLAS	46.5	40.0	31.4	88.9	68.1
	LAS	53.9	53.1	33.5	89.5	74.0
	Feats	60.6	51.8	48.3	94.8	83.1
PL	MSLAS	43.0	39.6	31.8	75.0	60.4
	LAS	52.8	54.0	35.3	76.5	65.6
	Feats	60.3	54.8	50.2	86.2	78.5
SR	MSLAS	49.6	42.4	41.0	86.6	76.0
	LAS	58.3	58.7	43.5	88.3	80.6
	Feats	70.0	59.2	60.1	95.6	89.5
SV	MSLAS	55.1	45.7	47.5	86.6	65.0
	LAS	64.3	63.5	49.9	87.7	69.7
	Feats	67.8	56.0	61.5	95.7	84.6
TR	MSLAS	44.8	35.1	17.0	58.7	48.3
	LAS	52.3	53.2	23.6	60.9	52.7
	Feats	67.2	50.4	46.8	82.1	75.5

Table 3: Results for each language

across all languages, with 78.7. Both submissions significantly outperformed all baselines across all scores. Of the baselines, the best-performing system was the Mono system, which used few-shot prompting to learn from examples in the same language.

Considering the results per language in Table 3, the UM system consistently outperformed the others in all languages. All systems notably struggled with Turkish and Hebrew, and scores did not seem to correlate in an obvious way with the number of sentences in the training data in Table 1.

7 Discussion

7.1 Abstract Nodes

Neither of the submitted systems was able to handle abstract nodes. They were also not handled by the fine-tuning baseline. Acar and Eryiğit (2025) note that the Turkish data has by far the highest rate of abstract nodes (13.45% as opposed to 3.61% in the second-highest, Polish). This is significant because all systems that do not model abstract nodes report their worst performance on Turkish. We cannot differentiate whether this is due to the agglutinative nature of Turkish, as we do not include any other agglutinative language in our sample, or simply due to points lost to missing abstract nodes, as well as cascading errors.

7.2 Tokenisation

The test data was provided in an untokenised format, and systems and baselines therefore had to make choices about the tokenisation, introducing errors which inevitably propagated down the line. The UM system made an attempt at re-engineering the tokenisation of the original data by selecting the stanza tokenisation model that results in the highest downstream performance. Because the evaluation script aligns either from the beginning or the end, one extra (or missing) word towards the middle of the sentence will cause the rest of the sentence to be misaligned and the score to drop significantly. This points to the improvement of tokenisation as an opportunity for improvement for parsers.

7.3 Multilinguality

All systems were based on a multilingual encoder model and then finetuned with the MSP training data on each language separately. Future systems could explore something more multilingual, especially for related languages like Italian and Portuguese, given how little data is available in total.

8 Conclusion

This paper presents the results of the UniDive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing, introducing a novel representation and evaluation metric designed to be more inclusive of typologically diverse languages. By shifting the focus from traditional distinction between “words” and “morphemes” to a distinction between content and function elements, the task encourages the development of new parsing technologies that operate in a more equitable fashion across all languages.

The results from the five evaluated systems demonstrate that the task is challenging yet solvable for modern multi-task learning models. The top-performing system, UM, achieved an impressive MSLAS score of 78.7, significantly outperforming all baselines and showcasing the potential of dedicated architectures for this problem. Nevertheless, challenges remain, particularly in handling abstract nodes and parsing non-Indo-European languages like Turkish and Hebrew, indicating clear directions for future research. Overall, this shared task successfully established a new benchmark for morpho-syntactic analysis and paved the way for more linguistically comprehensive parsing models.

Future iterations of the shared task will have the opportunity to cover a more diverse set of languages to allow a better evaluation, as well as better applications for studies in computational typology and morpho-syntax.

Acknowledgements

The Unidive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing was supported by the UniDive COST action (CA21167). Leonie Weissweiler was supported by a postdoctoral fellowship from the German Research Foundation (DFG, WE 7627/1-1). The Charles University team was supported by the grants *Language Understanding: from Syntax to Discourse* of the Czech Science Foundation (Project No. 20-16819X) and *LINDAT/CLARIAH-CZ* (Project No. LM2023062) of the Ministry of Education, Youth, and Sports of the Czech Republic. Adriana Pagano has a grant from the National Council for Scientific and Technological Development (CNPq 404722/2024-5; 313103/2021-6) and Minas Gerais State Agency for Research and Development (FAPEMIG). Reut Tsarfaty and Omer Goldman were supported by a generous grant from the Israel Science Foundation (ISF grant No. 670/23) and by a KAMIN grant from the Israel Innovation Authority, for which we are grateful.

References

Kutay Acar and Gülşen Eryiğit. 2025. Typology-aware multilingual morphosyntactic parsing with functional node filtering. In *Proceedings of The UniDive 2025 shared task on multilingual morpho-syntactic parsing*.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Linda Alfieri and Fabio Tamburini. 2016. (Almost) Automatic Conversion of the Venice Italian Treebank into the Merged Italian Dependency Treebank Format. In Anna Corazza, Simonetta Montemagni, and Giovanni Semeraro, editors, *Proceedings of the Third Italian Conference on Computational Linguistics CLiC-it 2016 : 5-6 December 2016, Napoli*, Collana dell'Associazione Italiana di Linguistica Computazionale, pages 19–23. Accademia University Press, Torino.

Guntis Bārzdiņš, Normunds Grūzītis, Gunta Nešpore, and Baiba Saulīte. 2007. Dependency-based hybrid model of syntactic analysis for the languages with a rather free word order. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 13–20, Tartu, Estonia. University of Tartu, Estonia.

Vuk Batanović, Nikola Ljubešić, Tanja Samardžić, and Tomaž Erjavec. 2023. Serbian linguistic training corpus SETimes.SR 2.0. Slovenian language resource repository CLARIN.SI.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, and 76 others. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Cristina Bosco, Simonetta Montemagni, and Maria Simi. 2013. Converting Italian Treebanks: Towards an Italian Stanford Dependency Treebank. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 61–69, Sofia, Bulgaria. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3278 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

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.

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. In *Proceedings of the 2019 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert MW Dixon and Alexandra Y Aikhenvald. 2002. Word: a typological framework. *Word: A cross-linguistic typology*, pages 1–41.

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

Magali Sanches Duran, Lucelene Lopes, Maria das Graças Volpe Nunes, and Thiago Alexandre Salgueiro Pardo. 2023. The dawn of the porttinari multigenre treebank: Introducing its journalistic portion. *Anais*.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia. Association for Computational Linguistics.

Jan Einarsson. 1976. Talbankens skriftspråkskonkordans.

Katja Filippova and Michael Strube. 2007. Generating constituent order in German clauses. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 320–327, Prague, Czech Republic. Association for Computational Linguistics.

Omer Goldman, Khuyagbaatar Batsuren, Salam Khalifa, Aryaman Arora, Garrett Nicolai, Reut Tsarfaty, and Ekaterina Vylomova. 2023. SIGMORPHON–UniMorph 2023 shared task 0: Typologically diverse morphological inflection. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 117–125, Toronto, Canada. Association for Computational Linguistics.

Omer Goldman and Reut Tsarfaty. 2022. Morphology without borders: Clause-level morphology. *Transactions of the Association for Computational Linguistics*, 10:1455–1472.

Jan Hajič, Eduard Bejček, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague dependency treebank - consolidated 1.0. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5208–5218, Marseille, France. European Language Resources Association.

Martin Haspelmath. 2011. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia Linguistica*, 45(1):31–80.

Demian Inostroza Améstica, Meladel Mistica, Ekaterina Vylomova, Chris Guest, and Kemal Kurniawan. 2025. A joint multitask model for morpho-syntactic parsing. In *Proceedings of The UniDive 2025 shared task on multilingual morpho-syntactic parsing*.

Alphonse Juilland and Alexandra Roceric. 1972. *The Linguistic Concept of Word: Analytic Bibliography*. De Gruyter Mouton, Berlin, Boston.

Kimmo Koskenniemi. 1983. Two-level model for morphological analysis. In *International Joint Conference on Artificial Intelligence*.

Jiří Krámský. 1969. *The word as a linguistic unit*. Janua Linguarum. Series Minor. De Gruyter.

Lenka Krippnerová and Daniel Zeman. 2025. Periphrastic verb forms in Universal Dependencies. In *Proceedings of the Eighth International Conference on Dependency Linguistics (Depling, SyntaxFest 2025)*, pages 140–149, Ljubljana, Slovenia. 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, Valencia, Spain. Association for Computational Linguistics.

Silvia Luraghi. 2009. A model for representing polysemy: The Italian preposition da. In *Actes Du Colloque "Autour de La Préposition"*, pages 167–178, Caen. Presses Universitaires de Caen.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Joakim Nivre, Ali Basirat, Luise Dürlich, and Adam Moss. 2022. Nucleus composition in transition-based dependency parsing. *Computational Linguistics*, 48(4):849–886.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Thiago Alexandre Salgueiro Pardo, Magali Sanches Duran, Lucelene Lopes, Ariani di Felippo, Norton Trevisan Roman, and Maria das Graças Volpe Nunes. 2021. Porttinari: a large multi-genre treebank for brazilian portuguese. *Anais*.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for Universal Dependencies. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 96–101, Gothenburg, Sweden. Association for Computational Linguistics.

Shoval Sade, Amit Seker, and Reut Tsarfaty. 2018. The Hebrew Universal Dependency treebank: Past present and future. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 133–143, Brussels, Belgium. Association for Computational Linguistics.

Itiroo Sakai. 1961. Syntax in universal translation. In *Proceedings of the International Conference on Machine Translation and Applied Language Analysis*, National Physical Laboratory, Teddington, UK.

Tanja Samardžić, Mirjana Starović, Željko Agić, and Nikola Ljubešić. 2017. Universal Dependencies for Serbian in comparison with Croatian and other Slavic languages. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 39–44, Valencia, Spain. Association for Computational Linguistics.

Manuela Sanguinetti, Cristina Bosco, Alberto Lavelli, Alessandro Mazzei, Oronzo Antonelli, and Fabio Tamburini. 2018. PoSTWITA-UD: An Italian Twitter Treebank in Universal Dependencies. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Anastasia Shimorina, Yannick Parmentier, and Claire Gardent. 2021. An error analysis framework for shallow surface realization. *Transactions of the Association for Computational Linguistics*, 9:429–446.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Umut Sulubacak and Gülşen Eryiğit. 2018. Implementing universal dependency, morphology, and multiword expression annotation standards for turkish language processing. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(3):1662–1672.

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.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multitask 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.

Alina Wróblewska. 2018. Extended and enhanced Polish dependency bank in Universal Dependencies format. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 173–182, Brussels, Belgium. Association for Computational Linguistics.

Alina Wróblewska. 2014. *Polish Dependency Parser Trained on an Automatically Induced Dependency Bank*. Ph.D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

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, and 43 others. 2017a. 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.

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, and 43 others. 2017b. 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.

Arnold M Zwicky and Geoffrey K Pullum. 1983. Cliticization vs. inflection: English n't. *Language*, 59(3):502–513.

A Annotation Guidelines

These are official annotation guidelines as shared with the participants in the shared task's official GitHub repo.[7]

Introduction:

In this documentation, we first explain the general principles of MSP, and then elaborate on the feature set and the file format.

Motivation

Words have long been an essential concept in the definition of treebanks in Universal Dependencies (UD), since the first stage in their construction is delimiting words in the language at hand. This is done due to the common view in theoretical linguistics of words as the dividing line between syntax, the grammatical module of word combination, and morphology, that is word construction.

We suggest defining the content-function boundary to differentiate 'morphological' from 'syntactic' elements. In our morpho-syntactic data structure, content words are represented as separate nodes on a dependency graph, even if they share a whitespace-separated word, and both function words and morphemes contribute morphology-style features to characterize the nodes.

Principles

* **Independence from Word Boundaries:** Delimiting syntactically relevant words gets exponentially more complicated, the less isolating the languages are. Thus, this operation, which is as simple as breaking the text on white spaces for English, is borderline impossible for polysynthetic languages, in which a single word may be composed of several lexemes that have predicate-argument relations. This reflects the fact that, despite the presumed role of words in contemporary linguistics, there is no consensus on a coherent cross-lingual definition of words. We will thus avoid (most) theoretical debates on word boundaries, and solve much of the word segmentation inconsistencies that occur in UD, either across languages, e.g., Japanese is treated as isolating and Korean as agglutinative, even though they are very similar typologically, or across treebanks of the same language, e.g., the different treebanks for Hebrew segment and attribute different surface forms for clitics.

* **Content-Function Divide:** The central divide in an MS graph is between content words (or morphemes) and function words (or morphemes). Content words form the nodes, while the information from function words is represented as features modifying the content nodes.

* **Crosslingual Parallelism:** Morphosyntactic Annotation will bring the trees of very different languages much closer together and thus enable new typological studies. In isolating languages, the data will explicitly surface MS features that are expressed periphrastically. Morpho-syntactic data will be more inclusive towards languages that are currently treated unnaturally, most prominently noun-incorporating languages. Morpho-syntactic models will be able to parse sentences in more languages and enable better cross-lingual studies.

* **Minimal Deviation from CoNLL-U:** We will use the well-established CoNLL-U file format for our data. The morpho-syntactic features are replacing the morphological features, and function nodes are recognizable by the lack of content in this column. This format is a variant of the CoNLL-U Plus format from which the data was constructed.

Schema Description

File Format

The format for morpho-syntactic parsing data is a simple alternation of UD's CoNLL-U format. It includes a replacement of a single column, morphological features, with morpho-syntactic features (named: MS-FEATS) for every UD node that contains a content word. UD nodes that contain function words should have empty (i.e. _) MS features. In addition, columns that are irrelevant for MSP, like MISC and XPOS, are also left empty.

[7] https://github.com/UniDive-MSP/MSP-shared-task

Morpho-Syntactic Features

As the key characteristics of morpho-syntactic dependency trees, morpho-syntactic features (MS features) are modelled after the morphological features in UD and may be viewed as a generalization of them. Like in UD, the features are an alphabetically ordered set of name and value pairs separated by pipes, of the structure `Name1=Value1|Name2=Value2`.[8] Most feature names and values are equivalent to those in UD, for example `Gender=Masc`, `Voice=Pass`, etc.

However, MS features also differ from morphological features in a couple important characteristics:

* The features are only defined for content nodes (see below).

  - Function words should not have MS features. All the information they convey should be expressed as features on the relevant content node.
  - Note: since the file format is a modified version of UD's CoNLL-U, function words may appear in the final output, their MS-feats column should be `_`. This is in contrast with content words that happen to have no MS-feats that should contain an orphan pipe `|`.

* The features are defined not only by morphemes but by any grammatical function marker, be it a morpheme or a word. So the content node *go* in *will go* should bear the feature `Tense=Fut`.

  - All applicable features should be marked on the respective content nodes, even if expressed by non-concatenative means (as long as they are grammatical). E.g., the node go in *did you go?* should be marked with `Mood=Ind;Int` even though the interrogative mood is expressed mostly by word order.

* Features should be applied only to their relevant node. In other words, no agreement features are needed, and in a phrase like *he goes* only *he* should bear `Number=Sing|Person=3`, and *goes* should have only `Tense=Pres` (and other features if relevant).

* The feature structure is not flat. In other words, features are not necessarily single strings. They can contain:

  - a list of values separated by a semicolon, for example `Aspect=Perf;Prog` on the verb of the English clause *I have been walking*,
  - a negation of a value, for example `Mood=not(Pot)` on the Turkish verb *yürüyemez* ("he can't walk") where the negation refers to the ability,[9]
  - a conjunction of values. This mechanism is to be used only in cases of explicit conjunction of grammatical constructions, for example `Case=and(Cnd,Temp)` is the manifestation of the English phrase *if and when* when connecting two clauses (see below for discussion on the `Case` feature),
  - and a disjunction of values, `Tense=or(Fut,Pas)`

* If a feature includes multiple values in any kind of order or structure, they are ordered alphabetically in accordance with the general UD guidelines.

The mapping from morpho-syntactic constructions to features does not have to be one-to-one. In cases where several constructions have the exact same meaning (e.g., they differ in geographic distribution, register or personal preferences), it is perfectly suitable to assign the same feature combination to both of them. For example, in Spanish, both *comiera* and *comiese* will be assigned `Aspect=Imp|Mood=Sub|Tense=Past|VerbForm=Fin` (remember that the agreement features should appear only on the relevant argument).

The categories of words to be "consumed" into MS features are usually: auxiliaries, articles, adpositions, conjunctions and subordinators, and some particles. These categories may not neatly correspond to UD POS tags. Some clearly do, like auxiliaries (POS tag `AUX`), while others, like `DET`, may include also contentful word, like *all* and *every*. Some POS tags like `ADV` mix many contentful words (*nicely*, *rapidly*, *often*, etc.) with a few that serve as conjunctions (*when*, *then*, etc.), and in rare cases the same word may be considered functional or contentful depending on the context.[10]

[8] The feature set is unordered in theory, but in practice the features are ordered alphabetically by feature name, just to make the annotations consistent.

[9] This is in contrast with the verb *yürümebilir* (literally "he is able to not walk", i.e., he may not walk), where the negation pertains to the verb itself and should be tagged as `Mood=Pot|Polarity=Neg`.

[10] Compare the word *then* in the sentence *if you want, then I'll do it* (functional) to the same word in *I didn't know what to do,*

Feature Inventory

Since the MS features are a generalization of UD's morphological features, their types and possible values are also highly similar with that of UD's features. Therefore, for most features, the list in UD is sufficient in characterizing content nodes in MS trees as well. The most prominent exception to this is the expansion of the `Case` feature.

Originally, the `Case` feature characterized the relation between a predicate and its argument, almost always a nominal, but for MS trees its role is expanded twice. First, in line with the principle of independence from word boundaries, in MS trees this feature corresponds to traditional case morphemes as well as adpositions (these usually have `case` as `DEPREL` in UD trees) and coverbs when such exist. The inclusion of adpositions in determining the `Case` feature entails the expansion of cases possible in almost any language. Nominals in German, for example, now have an elative case (indicating motion from the inside of the argument) expressed by the combination of the synthetic dative case and the periphrastic *aus* preposition.

The second expansion of the `Case` feature is that in MS trees this feature is also used to characterize predicate-predicate relations, hence it is applicable also to verbal nodes and it "consumes" also conjunctions and subordinators. So *fell* in *I cried until I fell asleep* and *today* in *It is true until today* will both get a `Case=Ttr` because they are both marked by the function word *until*.

In general, the same function word/morpheme combination should be mapped to the same `Case` value, even if it serves multiple functions. For example, the Swahili preposition *na* should be mapped only to `Case=Conj` even when it serves a function of introducing the agent of a passive verb.

Table 4 details a set of universal values for the `Case` feature. These features do not cover all possible relations, and in some cases when there are adpositions or conjunctions that do not correspond to any of the features, the value of the respective feature should be the canonical citation form of the function word transliterated into Latin letters.

Content Nodes

Content nodes, to which morpho-syntactic features are to be defined, are all words or morphemes from open classes (like nouns, verbs and adjectives) that do not convey a grammatical modification of another word.[11] These content words form a morpho-syntactic tree.

Note that copulas are not content words. In sentences with copulas refer to the nominal as the predicate and tag it with the features expressed by the copula.

For example, in the sentence *the quick brown fox jumps over the lazy dog* there are 6 content words (quick, brown, fox, jump, lazy, dog) and 3 function words (the, over, the).

In compounds or headless expressions, i.e., cases where one of the `fixed`, `flat` or `goeswith` `DEPREL`s are used, all words are judged together to either be of content or of function. Usually, such cases will be contentful, but sometimes a fixed expression can be a multi-word adposition, for example *as well as* and *because of*.

Abstract Nodes

In addition to words from open classes, content nodes also include all arguments and predicates in the sentence. The implications of this are twofold:

1. Pronouns should always be represented as nodes with MS features, regardless of your theoretical position on whether pronouns are contentful or a mere bundle of features.
2. Arguments that do not appear explicitly in a sentence but are expressed implicitly (i.e., by agreement of their predicate) should also be represented by their own node. However, this node lacks `FORM` or `LEMMA` fields and is therefore an abstract node. Abstract nodes should appear after the node from which they inherit their features and should have a special ID in the form of X.1, X.2 etc.

The most common use case of abstract nodes is when pronouns are dropped. For example, in Basque, the UD nodes:

```
4 ziurtatu ziurtatu VERB _ Aspect=Perf|VerbForm=Part 0 root _ _
5 zuten edun AUX _ Mood=Ind|Number[abs]=Sing|Number[erg]=Plur|Person[abs]=3|Person[erg]=3|Tense=Past|VerbForm=Fin 4 aux _
      ReconstructedLemma=Yes
```

then I understood (then stands for "after some time" hence contentful).

[11] In most languages, content nodes are equivalent to words. However, in some noun incorporating languages open class nouns can appear as morphemes concatenated to another content node that is the verb.

is be tagged as:

```
4 ziurtatu ziurtatu VERB _ Aspect=Perf|Mood=Ind|Tense=Past|VerbForm=Fin 0 root _ _
5 zuten edun AUX _ _ _ _ _ _
5.1 _ _ _ _ Case=Erg|Number=Plur|Person=3 4 nsubj _ _
5.2 _ _ _ _ Case=Abs|Number=Sing|Person=3 4 obj _ _
```

Note that node 5 now doesn't have MS-feats and therefore it will be dropped from the MS tree.

This example underlines that the abstract nodes may be viewed as a replacement for feature layering. The advantage of this mechanism is that it equates the representation of agreement morphemes, clitics and full pronouns, and removes the need to decide which is which.

The same mechanism is used whenever an argument is missing from the clause as an independent word, but expressed in other means, i.e., not when an argument was dropped for pragmatic reasons or was otherwise not detectable from the surface forms. For example, the annotation of the Japanese sentence 宣言したのだ ('(he) proclaimed') should not contain an abstract node for the non-existent subject, although one is understood.

Abstract nodes are also to be used when the argument is outside the clause.

Gaps

Abstract nodes are also to be used in simple gaps, when there are function words referring to some missing argument. For example, a phrase like books to choose from, should be annotated as:

```
4 books book NOUN NN Number=Plur 2 obj _ _
5 to to PART TO _ _ _ _ _
6 choose choose VERB VB VerbForm=Inf 4 acl _ _
7 from from ADP IN _ _ _ _ _
7.1 _ _ _ _ Case=Abl 6 obl _ _
```

So node `7.1` is created to carry the feature of the function word *from*.

Cases of more complex gaps are largely excluded from this shared task's data.

A.1 Case Values

Table 4: The inventory of `Case` values with examples from several shared task languages. The examples on the same line are often translation equivalents but this is not guaranteed, as sometimes a different example is more appropriate in a particular language.

		EN	**CS**	**TR**
Argument alignment				
`Nom`	nominative	*he*	*on*	*o*
`Acc`	accusative	*him*	*jeho*	*onu*
`Abs`	absolutive	—	—	—
`Erg`	ergative	—	—	—
`Dat`	dative	—	*jemu*	*ona*
`Agt`	agentive	—	—	—
Static location				
`Loc`	locative	*at school*	—	*okulda*
`Ine`	inessive	*in the house*	*v domě*	—
`Ces`	interessive	*among the students*	*uprostřed lesa*	—
`Int`	intrative	*between us*	*mezi námi*	—
`Ext`	external	*outside the house*	*vně domu*	—
`Ade`	adessive	*on the table*	*na stole*	—
`Adt`	superadessive	*atop the mountain*	—	—
`Adh`	lateradessive	—	—	—
`Apu`	apudessive	*beside the house*	*vedle domu*	—
`Chz`	chezative	—	*u Martina*	—
`Cir`	circumessive	*around the house*	*kolem domu*	—
`Prx`	proximative	*near the house*	*blízko domu*	—
`Dst`	distantive	*far from the house*	*daleko od domu*	—

Table 4: The inventory of Case values with examples from several shared task languages. The examples on the same line are often translation equivalents but this is not guaranteed, as sometimes a different example is more appropriate in a particular language.

		EN	**CS**	**TR**
Sup	superessive	*above the house*	*nad domem*	*bir yılı aşkın*
Sub	subessive	*under the house*	*pod domem*	—
Ant	antessive	*in front of the house*	*před domem*	—
Pst	postessive	*behind the house*	*za domem*	*amacından öte*
Ori	orientative	—	—	—
Rev	revertive	—	—	—
Opp	oppositive	*opposite the house*	*naproti domu*	—
Tot	total	*throughout the house*	—	—
Direction focused on origin				
Abl	ablative	*from the school*	*od školy*	—
Egr	egressive	—	—	—
Ela	elative	—	*z domu*	—
Cne	interelative	—	*zprostřed lesa*	—
Ite	intraelative	*from between*	—	—
Exe	exelative	—	—	—
Del	delative	*off the table*	*se stolu*	—
Ape	apudelative	—	—	—
Spe	superelative	*from above the house*	—	—
Sbe	subelative	*from under the house*	*zpod domu*	—
Ane	antelative	—	—	—
Pse	postelative	*from behind the house*	*zpoza domu*	—
Direction focused on path				
Per	perlative	—	*po ulici*	—
Crs	perlative across	*across the lake*	*napříč jezera*	—
Lng	perlative along	*along the river*	*podél řeky*	—
Pro	prolative	*via Berlin*	—	—
Inx	inprolative	*through the house*	*skrz dům*	—
Cnx	interprolative	—	—	—
Adx	adprolative	—	—	—
Apx	apudprolative	—	—	—
Spx	superprolative	*over the bridge*	*přes most*	—
Sbx	subprolative	—	—	—
Cix	circumprolative	—	*ob dům*	—
Asc	ascentive	*up the river*	—	—
Dsc	descentive	*down the river*	—	—
Direction focused on destination				
Lat	lative	*to the house*	—	—
Ter	terminative	*up to that house*	*po tamten dům*	—
Ill	illative	*into the house*	*do domu*	—
Cnl	interlative	—	*doprostřed lesa*	—
Itl	intralative	—	*mezi nás*	—
Exl	exlative	—	—	—
All	allative	*onto the table*	*na stůl*	—
Apl	apudlative	—	*k domu*	—
Spl	superlative	—	*nad dům*	—
Sbl	sublative	*to under the house*	*pod dům*	—
Anl	antlative	—	*před dům*	—

Table 4: The inventory of `Case` values with examples from several shared task languages. The examples on the same line are often translation equivalents but this is not guaranteed, as sometimes a different example is more appropriate in a particular language.

		EN	**CS**	**TR**
`Psl`	postlative	—	*za dům*	—
Temporal				
`Tan`	temporal antessive	*before lunch*	*dokud neobědvá*	*yemekten önce*
`Ttr`	temporal terminative	*until lunch*	—	*yemek saatine kadar*
`Lim`	limitative	—	—	—
`Tem`	temporal	*upon doing X*	*v sobotu*	—
`Tpx`	temporal approximative	*circa 9*	*v období dešt’ů*	—
`Din`	durative initiative	—	*počátkem zimy*	—
`Dur`	durative	*during winter*	*během zimy*	*yol boyunca*
`Der`	durative era	—	*za Caesara*	—
`Dtr`	durative terminative	—	*koncem zimy*	—
`Tdi`	temporal distributive	—	—	—
`Tps`	temporal postessive	*after lunch*	*jakmile doobědvá*	*yemekten sonra*
`Teg`	temporal egressive	*since winter*	*počínaje zimou*	*kıştan beri*
`Tbt`	temporal interessive	—	—	—
Relation				
`Atr`	complement / attribute	*that*	*že*	*-arak*
`Gen`	genitive	*of the house*	*domu*	—
`Psd`	possessed	—	—	—
`Par`	partitive	—	—	—
`Dis`	distributive	—	—	—
`Com`	comitative	*with Martin*	*s Martinem*	*Martin ile birlikte*
`Orn`	ornative	—	—	*uzun parmaklı*
`Abe`	abessive	*without Martin*	*bez Martina*	*Martin olmadan*
`Inc`	inclusive	*including Martin*	*včetně Martina*	—
`Add`	additive	—	—	—
`Exc`	exclusive	*except Martin*	*kromě Martina*	—
`Sbs`	substitutive	*instead of Martin*	*místo Martina*	—
Similarity				
`Ess`	essive	*as a teacher*	*jako učitel*	—
`Equ`	equative	—	—	*öğretmen kadar*
`Sem`	semblative	*like a teacher*	—	*öğretmen gibi*
`Rpl`	replicative	—	—	—
`Dsm`	dissemblative	*unlike a teacher*	*oproti učiteli*	—
`Cmp`	comparative	*than a teacher*	*než učitel*	—
`Dif`	differential	—	*o dva metry*	—
`Tra`	translative	—	—	—
`Exe`	exessive	—	—	—
`Cmt`	comment	*whereas*	*kdežto*	*halbuki*
Cause, consequence, circumstance, other				
`Cau`	causative	*because of rain*	*kvůli dešti*	*yağmurdan dolayı*
`Pur`	purposive	*in order to survive*	*aby přežil*	*hayatta kalmak için*
`Cns`	considerative	*considering the rain*	*na základě doporučení*	—
`Ign`	ignorative	*regardless of the rain*	*at’ prší nebo ne*	—
`Ccs`	concessive	*despite the rain*	*navzdory dešti*	*yağmura rağmen*
`Cnd`	conditional	*in case of rain*	*pokud bude pršet*	*eğer yağmur yağarsa*
`The`	themative	*about the rain*	*o dešti*	*yağmurla ilişkin*

Table 4: The inventory of Case values with examples from several shared task languages. The examples on the same line are often translation equivalents but this is not guaranteed, as sometimes a different example is more appropriate in a particular language.

		EN	**CS**	**TR**
Quo	quotative	*according to the law*	*podle zákona*	*yasaya göre*
Ins	instrumental	*using a hammer*	*kladivem*	—
Ben	benefactive	*for Martin*	*pro Martina*	—
Mal	malefactive	—	—	—
Adv	adversative	*against Martin*	*proti Martinovi*	*Martin'e karşı*
Evi	evitative	—	—	—
Voc	vocative	—	*Martine!*	—
Paratactic relations (CCONJ type)				
Conj	conjunctive	*and*	*a*	*ve*
Nnor	negative conjunctive	*nor*	*ani*	*ne*
Disj	disjunctive	*or*	*nebo*	*veya*
Advs	adversative	*but*	*ale*	*ama*
Reas	reason	*for*	*nebot'*	*çünkü*
Cnsq	consequence	*so*	*tedy*	*ki*

B Example Sentence Annotation

Original Representation

```
# sent_id = 00058111_26
# text = Karşınızdakine Sizi işe alıyorum, demek geçer aklınızın bir köşesinden.
1-2 Karşınızdakine _ _ _ _ _ _ _ _
1 Karşınızda karşı ADJ NAdj Case=Loc|Number=Sing|Number[psor]=Plur|Person=3|Person[psor]=2 7 iobj _ _
2 kine ki ADP Rel Case=Dat|Number=Sing|Person=3 1 case _ _
3 Sizi siz PRON Pers Case=Acc|Number=Plur|Person=2|PronType=Prs 4 obj _ _
4 işe iş NOUN Noun Case=Dat|Number=Sing|Person=3 7 ccomp _ _
5 alıyorum al VERB Verb Aspect=Prog|Mood=Ind|Number=Sing|Person=1|Polarity=Pos|Polite=Infm|Tense=Pres 4 compound _ SpaceAfter=No
6 , , PUNCT Punc _ 7 punct _ _
7 demek de VERB Verb Aspect=Perf|Case=Nom|Mood=Ind|Polarity=Pos|Tense=Pres|VerbForm=Vnoun 8 nsubj _ _
8 geçer geç VERB Verb Aspect=Hab|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Pres 0 root _ _
9 aklınızın akıl NOUN Noun Case=Gen|Number=Sing|Number[psor]=Plur|Person=3|Person[psor]=2 8 compound _ _
10 bir bir NUM ANum NumType=Card 8 compound _ _
11 köşesinden köşe NOUN Noun Case=Abl|Number=Sing|Number[psor]=Sing|Person=3|Person[psor]=3 8 compound _ SpaceAfter=No
12 . . PUNCT Punc _ 8 punct _ _
```

MSP-Adapted Representation

```
# sent_id = 00058111_26
# text = Karşınızdakine Sizi işe alıyorum, demek geçer aklınızın bir köşesinden.
1-2 Karşınızdakine _ _ _ _ _ _ _ _
1 Karşınızda karşı ADJ _ Case=Loc;Dat|Number=Sing|Person=3 7 iobj _ _
1.1 _ _ PRON _ Case=Gen|Number=Plur|Person=2|PronType=Prs 1 nmod:poss _ _
2 kine ki ADP _ _ _ _ _ _
3 Sizi siz PRON _ Case=Acc|Number=Plur|Person=2|PronType=Prs 4 obj _ _
4 işe iş NOUN _ Case=Dat|Number=Sing|Person=3 7 ccomp _ _
5 alıyorum al VERB _ Aspect=Prog|Mood=Ind|Polarity=Pos|Polite=Infm|Tense=Pres 4 compound _ _
5.1 _ _ PRON _ Case=Nom|Number=Sing|Person=1|PronType=Prs 5 nsubj _ _
6 , , PUNCT _ _ _ _ _ _
7 demek de VERB _ Aspect=Perf|Case=Nom|Mood=Ind|Number=Sing|Person=3|Polarity=Pos|Tense=Pres|VerbForm=Vnoun 8 nsubj _ _
8 geçer geç VERB _ Aspect=Hab|Mood=Ind|Polarity=Pos|Tense=Pres 0 root _ _
9 aklınızın akıl NOUN _ Case=Gen|Number=Sing|Person=3 8 compound _ _
9.1 _ _ PRON _ Case=Gen|Number=Plur|Person=2|PronType=Prs 9 nmod:poss _ _
10 bir bir NUM _ NumType=Card 8 compound _ _
11 köşesinden köşe NOUN _ Case=Abl|Number=Sing|Person=3 8 compound _ _
11.1 _ _ PRON _ Case=Gen|Number=Sing|Person=3|PronType=Prs 11 nmod:poss _ _
12 . . PUNCT _ _ _ _ _ _
```

A Joint Multitask Model for Morpho-Syntactic Parsing

Demian Inostroza, Mel Mistica, Ekaterina Vylomova, Chris Guest, Kemal Kurniawan
University of Melbourne
{inostrozaad, misticam, ekaterina.vylomova,
chris.guest, kurniawan.k}@unimelb.edu.au

Abstract

We present a joint multitask model for the UniDive 2025 Morpho-Syntactic Parsing shared task, where systems predict both morphological and syntactic analyses following novel UD annotation scheme. Our system uses a shared XLM-RoBERTa encoder with three specialized decoders for content word identification, dependency parsing, and morphosyntactic feature prediction. Our model achieves the best overall performance on the shared task's leaderboard covering nine typologically diverse languages, with an average MSLAS score of 78.7%, LAS of 80.1%, and Feats F1 of 90.3%. Our ablation studies show that matching the task's gold tokenization and content word identification are crucial to model performance. Error analysis reveals that our model struggles with core grammatical cases (particularly `Nom-Acc`) and nominal features across languages.[1]

1 Introduction

The UniDive 2025 Morpho-Syntactic Parsing shared task (Goldman et al., 2025) introduces a novel framework for dependency parsing that seeks to bridge the traditional divide between morphological and syntactic analysis. In conventional Universal Dependencies (Nivre et al., 2020), morphology and syntax are treated as distinct modules operating at different linguistic levels, with word boundaries serving as the interface between them. However, this separation has led to significant inconsistencies in how different languages and even different treebanks for the same language handle word segmentation and grammatical analysis. The shared task proposes to address these long-standing challenges by reorganizing grammatical representation around the content-function distinction rather than relying on theoretically problematic word boundaries, proposing a more typologically consistent approach to multi-linguistic parsing. For instance, in the sentence 'From the AP comes this story' shown in Table 1, traditional UD treats 'From' as a dependent of 'AP' with the deprel `case`, while the new framework transfers the grammatical meaning of 'From' as a morphosyntactic feature `Case=Abl` (Ablative) directly onto the content word 'AP'.

ID	Token	FEATS	HEAD	DEPREL
1	From	_	_	_
2	the	_	_	_
3	AP	Case=Abl|Definite=Def| Number=Sing	4	obl
4	comes	Mood=Ind|Polarity=Pos| Tense=Pres|VerbForm=Fin| Voice=Act	0	root
5	this	Number=Sing| PronType=Dem	6	det
6	story	Number=Sing	4	nsubj
7	:	_	_	_

Table 1: Example of the new annotation scheme used in the shared task

The task requires systems to predict both labeled dependency arcs and morphosyntactic features, but with a difference from standard Universal Dependencies parsing: the dependency tree consists only of content words (lexical words carrying semantic meaning like nouns, verbs, and adjectives), while function words (grammatical elements like adpositions, articles, and auxiliaries) contribute their grammatical information as features on related content words.

While the content-function distinction is explicit in the training data, systems must identify this distinction themselves at test time from raw text. This identification determines which words participate in the dependency tree and which contribute features to other words. Additionally, the multi-label nature of features, where a content word can have multiple feature values for a given feature class,

[1] Our code and models are publicly available: https://github.com/DemianInostrozaAmestica/shared_task_UD_official

Proceedings of The UniDive 2025 Shared Task on Multilingual Morpho-Syntactic Parsing, pages 19–26
August 26, 2025 ©2025 Association for Computational Linguistics

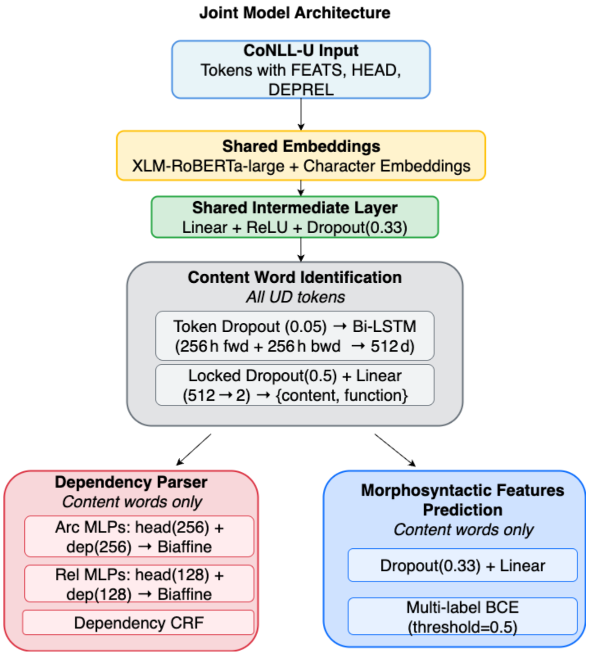
Joint Model Architecture
CoNLL-U Input
Tokens with FEATS, HEAD, DEPREL
Shared Embeddings
XLM-RoBERTa-large + Character Embeddings
Shared Intermediate Layer
Linear + ReLU + Dropout(0.33)
Content Word Identification
All UD tokens
Token Dropout (0.05) → Bi-LSTM (256 h fwd + 256 h bwd → 512 d)
Locked Dropout(0.5) + Linear (512 → 2) → {content, function}
Dependency Parser
Content words only
Arc MLPs: head(256) + dep(256) → Biaffine
Rel MLPs: head(128) + dep(128) → Biaffine
Dependency CRF
Morphosyntactic Features Prediction
Content words only
Dropout(0.33) + Linear
Multi-label BCE (threshold=0.5)

Figure 1: Joint model architecture for the shared task.

like `Case=Ine;Atr`,[2] requires models to learn intricate morphosyntactic patterns.

We present a joint multitask model (Figure 1) that explicitly addresses these challenges through three specialized decoders sharing a common XLM-RoBERTa encoder (Conneau et al., 2020), initialized from pre-trained multilingual representations. We design content word identification as an explicit task to be learned by the model rather than relying on intuition-driven heuristics. We participate in the multilingual track, training separate models for each of the nine languages, allowing us to tune hyperparameters specifically for each language's characteristics while still benefiting from multilingual pretrained representations. On the shared task's results, our system achieves the best overall performance with average scores of 78.7% MSLAS, 80.1% LAS, and 90.3% Feats F1 across all languages. Additionally, our model ranks first on each individual language, demonstrating the effectiveness of multitask learning for this task.

Our error analysis yields three main observations: (1) errors in tokenization and content word identification cascade through the pipeline, with gold annotations improving MSLAS by up to 12 points; (2) the majority of residual errors lie in nominal morphology—`Gender`, `Number`, and `Case`—with common `Nominative-Accusative` swaps; and (3) syntactic mislabels are concentrated in the `nmod` versus `obl` relation.

2 System Description

2.1 Model Overview

We propose a joint multitask model implemented using the Flair framework (Akbik et al., 2019) for morphosyntactic parsing, as shown in Figure 1. Although the evaluation metrics assess only dependency arcs and morphosyntactic features, producing these outputs requires distinguishing between content and function words. Because this classification is not given at test time, we treat it as an additional prediction task. Our system uses the large version of XLM-RoBERTa augmented with character embeddings (Akbik et al., 2018) as a shared encoder, both provided by the Flair framework. This encoder's output is then passed through a shared intermediate layer (linear transformation with ReLU and dropout) before being fed to three specialized decoders: content word identification, morphosyntactic feature prediction, and dependency parsing.

2.2 Decoders

Content word identification. The content word identification decoder accepts tokens as input. Each token's contextual embedding computed by the shared intermediate layer is passed through a bidirectional LSTM (256 hidden units in both directions). The LSTM output is then passed through a linear layer with 2 output units, each corresponds to "content" vs. "function" respectively. Training uses two forms of regularisation: token-level (word) dropout—zeroing the entire embedding of 5% of UD tokens—and locked dropout that masks 50% of the LSTM outputs with the same pattern across all timesteps. Class-weighted cross-entropy loss function is then used to compensate for the imbalance between the number of content and function tokens.

Morphosyntactic features. The morphosyntactic features decoder consists of a single linear layer that performs multi-label classification directly from the output of the shared intermediate layer. For each content word, it outputs probabilities for all possible feature-value pairs in the vocabulary (e.g., `Case=Gen`, `Number=Sing`, `Voice=Act`). Using sigmoid activation with a 0.5 threshold, the model can predict multiple features per token—for instance, a noun might simultaneously

[2] Ine=Inessive, "inside an enclosed area"; Atr="complement, attribute". Both definitions come from the official `Case` inventory supplied by the shared-task organisers.

have `Number=Plur` and `Case=Gen`. Complex features with multiple values (like `Case=Ine;Atr`) are handled by predicting each component separately, allowing the model to learn different value combinations. Function words bypass this decoder entirely and receive '_' as their feature value. At training time, we use gold content word (i.e. checking if its feature values exist). In contrast, we use the predicted content words by the content word identification at test time.

Dependency parser. The parsing decoder employs separate multilayer perceptrons (MLPs) for arc and relation prediction with biaffine attention mechanisms, following Dozat and Manning (2016). The arc MLPs have 256 hidden units while the relation MLPs use 128 units, both with layer normalization and ReLU activation. Operating exclusively on content words, we frame the parser as a conditional random field over projective dependency trees that we implement using TorchStruct (Rush, 2020). Similar to the morphosyntactic feature decoder, we use gold and predicted content word at training and test time respectively.

2.3 Data Handling and Inference

While the shared task data includes abstract nodes for representing implicit arguments, we initially attempted to handle them through sequence labeling by inserting mask tokens at potential abstract node positions. However, this approach introduced noise that degraded performance across all metrics, as incorrect abstract node predictions propagated errors to downstream decoders. Therefore, our final system filters out abstract nodes during data loading, simplifying the parsing task while improving overall performance.

During inference, raw text is first segmented into word tokens using Stanza (Qi et al., 2020). Since tokenization quality impacts downstream performance but is not the focus of this shared task, we choose to leverage Stanza's pre-trained models rather than training custom tokenizers. For each language, we evaluated different Stanza model variants on the development set and selected those that best matched the gold tokenization (e.g., HTB for Hebrew, IMST for Turkish). This selection was done manually by running the full pipeline with each available Stanza model variant and choosing the one that achieved the highest metrics on the official evaluation script.

We apply minimal post-processing to ensure valid output. For content word identification, tokens with confidence below 0.6 that appear between two tokens of the opposite type are relabeled to match their context (e.g., a low-confidence function word between two content words becomes content). As a fallback for extreme cases where content word identification predicts all tokens as function words (particularly in very short sentences of 2-3 tokens), we force the first token to be content with `deprel='root'` and `features='|'`. This ensures every sentence has at least one parseable token.

2.4 Training Objective and Optimization

The model is trained end-to-end using a weighted sum of the three decoders' losses: $\mathcal{L}_{\text{total}} = w_{\text{parser}}\mathcal{L}_{\text{parser}} + w_{\text{morph}}\mathcal{L}_{\text{morph}} + w_{\text{CWI}}\mathcal{L}_{\text{CWI}}$, where the weights are hyperparameters tuned for each language. The parser uses negative log-likelihood loss over projective trees, the morphosyntactic decoder uses binary cross-entropy for multi-label classification, and the content word identification uses class-weighted cross-entropy to handle class imbalance.

3 Experimental Setup

The shared task provided training and development sets for multiple languages. To simulate a realistic evaluation scenario, we split the official training data into 90% for training and 10% for development, using the official development set as our local test set. This allowed us to tune hyperparameters and select models before the official test release. The languages included in our experiments were English, Turkish, Hebrew, Czech, Polish, Portuguese, Italian, Serbian, and Swedish, with training sizes ranging from approximately 3,000 to 10,000 sentences depending on the language.

We develop a custom data loader to handle the modified CoNLL-U format used in the shared task. The loader automatically extracts content words by examining the FEATS column, where '_' indicates function words and any other value indicates content words. As mentioned before, we filter out abstract nodes during loading.

All models are trained using AdamW optimizer (Loshchilov and Hutter, 2019) with an initial learning rate of 2×10^{-5} and batch size of 16 for 25 epochs. We employ early stopping with patience of 1 epoch and learning rate reduction by factor 0.5 when validation loss plateaus. Training is performed on a NVIDIA A100 GPU with 32GB

RAM on a high-performance computing cluster, with each model taking approximately 1-5 hours to converge.

We perform grid search over task-specific loss weights on our development split. The optimal weights varied by language—for example, Turkish benefited from weighting parsing and morphological feature losses twice as much as content word identification (2.0:2.0:1.5), while English performed better with parsing weighted most heavily, followed by morphological features and content word identification (2.0:1.5:1.0).

For each language, we train three models with different random seeds using the same hyperparameter configuration to verify training stability and robustness. All three models are evaluated on our local test set (the official development set) using the shared task's official evaluation script.

Once hyperparameters are selected, we retrain a single model for each language using the complete official training and development data combined. These final models use the same hyperparameters determined during development. These models are used to generate predictions on the official covered test set, which contains only raw text without annotations. Evaluation is performed using the official script which computes three metrics: MSLAS (morphosyntactic features F1 only on correctly parsed tokens), LAS (labeled attachment score), and Feats F1 (morphosyntactic features F1).

4 Results

This section is divided into two parts: first, we present official test results from models trained on all available data (official train + dev combined) and evaluated on the covered test set; second, we report development results using our local data splits (90% train, 10% dev, official dev as test) to analyze design choices and hyperparameter impact.

4.1 Official Test Results

Table 2 presents the official test results from models trained on all available data. Our system achieved the highest performance among all submissions with an average MSLAS of 78.7%. The results show strong performance across most languages, with MSLAS scores exceeding 83% for seven of the nine languages. Portuguese (88.9%) and Czech (87.1%) achieved the highest scores, consistent with our development results. The morphologically complex languages continued to present challenges—Turkish (58.7%) and Hebrew (68.7%) showed the lowest performance.

Language	MSLAS	LAS	Feats
Czech	87.1	88.0	95.2
English	83.8	85.1	94.9
Hebrew	68.7	71.4	83.4
Italian	73.0	73.7	84.7
Polish	75.0	76.5	86.2
Portuguese	**88.9**	**89.5**	94.8
Serbian	86.6	88.3	95.6
Swedish	86.6	87.7	**95.7**
Turkish	58.7	60.9	82.1
Average	78.7	80.1	90.3

Table 2: Official test results on the covered test set. Our system achieved the highest average MSLAS score (78.7%) among all submissions.

System	MSLAS	LAS	Feats
Our model	**78.7**	**80.1**	**90.3**
baseline_multi	47.3	55.4	64.2
baseline_cross	36.7	51.2	50.6
baseline_finetune	33.0	36.1	52.3

Table 3: Comparison with baseline systems (average across all languages).

The baseline systems provide important context for understanding the task's difficulty (Table 3). The multilingual few-shot baseline achieved moderate performance (average MSLAS 47.3%), while the cross-lingual few-shot approach struggled significantly (36.7%), highlighting the importance of language-specific examples. The finetuned BERT baseline performed poorest (33.0%), suggesting that the reformulated parsing task with its content-function distinction and expanded feature inventory benefits from specialized modeling approaches. Our 31.4 point improvement over the best baseline (78.7% vs 47.3%) indicates that combining pretrained representations with task-specific architectural components can effectively address the challenges of unified morphosyntactic parsing.

4.2 Development Results

The ablations in Figure 2 show that most of the gain comes from using gold tokenization, with a smaller but consistent boost from explicit content/function labeling. Hebrew makes this clear: MSLAS goes from 75.2 (**Full**) → 84.5 (**GoldTok**, +9.3) → 85.7 (**GoldWT**, +1.2; +10.5 total). This motivates per-language tokenizer selection and modeling content word identification as a dedicated task.

Loss-weight tuning largely favored `parser=2.0`,

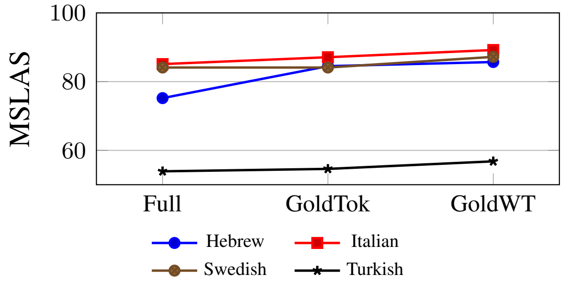

Figure 2: MSLAS across setups (Full, GoldTok, GoldWT) for four languages with the largest gains. **Full**: predicted tokenization and predicted content word identity. **GoldTok**: gold tokenization with predicted content word identity. **GoldWT**: gold tokenization plus gold content word identity.

Language	Parser	Morph	CWI
Czech	2.0	1.5	1.0
English	2.0	1.5	1.0
Hebrew	2.0	1.5	1.0
Italian	2.0	1.5	1.0
Polish	2.0	1.5	1.0
Portuguese	2.0	1.5	1.0
Serbian	2.0	1.5	1.0
Swedish	2.0	1.5	1.5
Turkish	2.0	2.0	1.5

Table 4: Optimal loss weight configurations by language. CWI =content word identification.

`morph=1.5`, `CWI=1.0`; Turkish and Swedish benefited from higher weights on `morph`/`CWI` (Table 4).

5 Error Analysis

We performed error analysis on the models trained with our local data splits (90% train, 10% dev, official dev as test). We analyzed only the first seed model for each language, as the low standard deviations indicate minimal variation across seeds. The analysis uses scripts that replicate the official evaluation logic to ensure our error categorization matches the scoring methodology.

5.1 Nominal Morphology Errors

The main feature prediction errors occur in nominal morphology, with Gender, Number, and Case showing the highest confusion rates. Since languages have different feature inventories (e.g., Czech includes Dual while others do not), creating a unified confusion matrix is not feasible. We selected Czech as a representative example because it has by far the most training data points, resulting in more stable model behavior.

Our analysis of Czech reveals strong overall performance, with 99.1% accuracy for Gender and 99.6% for Number predictions. For Gender, the model correctly classifies the vast majority of instances, with Feminine (12,405 correct), Masculine (14,914 correct), and Neuter (5,325 correct) all showing high diagonal values in the confusion matrix. The annotation scheme includes syncretic forms like "Fem,Masc" for grammatically ambiguous cases. The most common confusions occur between Masculine and Feminine (110 instances misclassified as Feminine when Masculine was correct), though these remain relatively rare. Similarly, for Number, Singular (24,653 correct) and Plural (9,587 correct) are accurately predicted, with minimal confusion between categories (only 44 Singular instances misclassified as Plural, and 85 Plural instances misclassified as Singular).

Since our model uses multi-label classification with sigmoid activation (threshold 0.5), it occasionally predicts semantically incompatible feature combinations—for instance, simultaneously predicting both a specific gender value (e.g., "Fem") and a syncretic form containing that value (e.g., "Fem,Masc"). While these semantically nonsensical predictions are rare (occurring in fewer than 100 instances out of over 30,000), they suggest that post-processing constraints based on linguistic compatibility rules could eliminate such predictions and further improve the performance.

For `Case` features, plotting a confusion matrix is impractical due to the >100 possible values in the expanded inventory. While there is some variation across languages, aggregating the most frequent errors reveals consistent patterns. Table 5 shows the 10 most common `Case` confusions averaged across all languages. The high frequency of `Nom-Acc` confusions (154 and 140 instances) reflects both the prevalence of these cases in the data and their potential ambiguity—distinguishing core arguments becomes particularly challenging in complex sentences with long-distance dependencies or multi-clause structures. This pattern holds across languages despite their individual variations, suggesting that even within the expanded `Case` system, these fundamental grammatical distinctions remain challenging when syntactic complexity increases. These systematic errors in core grammatical cases suggest a targeted improvement strategy: increasing loss weights for frequently confused cases (especially `Nom/Acc`) during training. Given our joint model architecture where all tasks share embed-

Count	Gold case	Predicted case
154	Acc	Nom
140	Nom	Acc
77	Nom	Conj;Nom
47	Nom	Gen
44	Conj;Nom	Nom
43	Gen	Nom
36	Acc	Gen
25	Gen	Acc
22	Gen	Conj;Gen
15	Dat	Ins

Table 5: Top 10 most frequent case prediction errors (average across all languages).

dings, better representation of these central arguments could benefit dependency parsing as well.

5.2 Spatial Case Results

We evaluate our model's performance on the fine-grained spatial `Case` values, a particularly challenging subset due to the numerous possible inflectional meanings that this domain contains.[3] The complete inventory of spatial cases includes over 40 fine-grained distinctions. Table 6 shows high performance across all languages (F1 scores 89.2-98.7%), demonstrating that our model successfully learned the unified `Case` system for spatial meanings. This annotation scheme directly names inflectional meanings regardless of the grammatical markers used - for instance, in Polish, when ablative meaning is expressed periphrastically through a clitic (an adposition)[4] plus an inflected form (a root with a genitive case affix), the system assigns the inflectional meaning (e.g., `Case=Abl`) instead of the genitive meaning conveyed by the suffix on its own. Our model's performance on these distinctions suggests it effectively captures the mapping between diverse surface forms and their underlying spatial semantics. This opens opportunities for injecting linguistic knowledge about spatial relations in downstream applications, leveraging the semantic transparency of the annotation scheme.

Language	Precision	Recall	F1
Czech	98.2	98.4	98.3
English	93.3	90.3	91.8
Hebrew	88.4	90.0	89.2
Italian	98.0	97.0	97.5
Polish	98.4	97.2	97.8
Portuguese	**98.5**	**99.0**	**98.7**
Serbian	96.4	93.7	95.1
Swedish	98.4	96.1	97.2
Turkish	94.7	96.4	95.6

Table 6: Spatial case performance (%) across languages using micro-averaged metrics.

Count	Gold label	Predicted label
67	obl	nmod
63	nmod	obl
11	obj	nsubj
11	advmod	_
10	nmod	flat
10	nmod	amod
9	nsubj	obj
9	iobj	obj
8	obj	obl
8	nsubj	root

Table 7: Top 10 most frequent deprel labeling errors (average across all languages).

5.3 Dependency Parsing Errors

For dependency relation errors, we analyze confusions across all languages since the label inventory is universal. Table 7 presents the 10 most frequent labeling errors aggregated across languages. The `nmod-obl` confusion dominates with 67 and 63 instances respectively, accounting for over 40% of the top errors. This pattern is linguistically expected as the boundary between nominal modifiers and oblique arguments could involve borderline cases.

Unlike other languages where errors concentrate on the `nmod/obl` distinction, Turkish shows a much more dispersed error pattern with confusions spread across many dependency relations. This suggests that our joint architecture may not be optimal for Turkish's non-projective structures and rich morphology. A dedicated non-projective parsing algorithm might better capture Turkish's complex dependency patterns.

Additionally, we analyze attachment distance patterns specifically for parsing errors (i.e., tokens with incorrect head assignments). Figure 3 shows the distribution of attachment distances for Czech parsing errors, comparing gold (blue) versus predicted (orange) distances for these misparsed tokens. The graph reveals that while most gold attachments occur at distances 1-3, the model's errors tend to predict longer distances (note the orange bars extending further right). This indicates

[3] These notions are understood as defined by Haspelmath (2025): "inflectional meaning" designates the specific meaning conveyed by an inflected form (for example, ablative), and "inflectional domain" denotes the broader class of related properties in which this meaning is categorized (for example, case).

[4] The classification of adpositions as clitics follows the definition proposed by Haspelmath (2023).

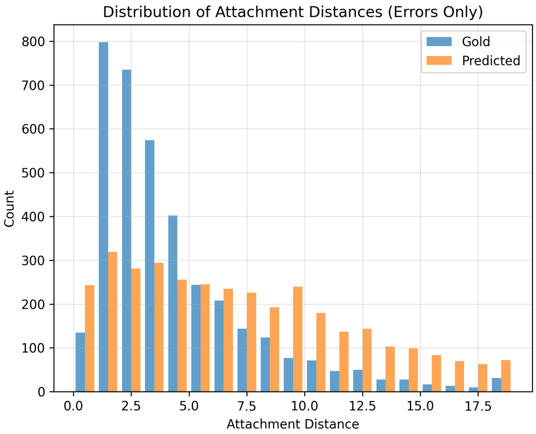

Figure 3: Distribution of attachment distances for parsing errors in Czech.

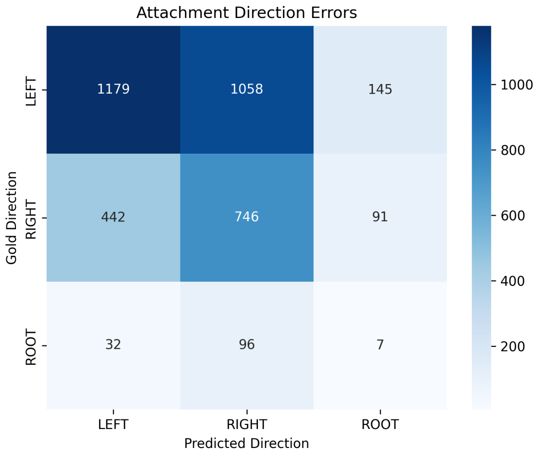

Figure 4: Attachment direction confusion matrix for Czech.

the parser frequently overlooks nearby heads in favor of more distant ones when making mistakes. Figure 4 presents a heatmap of misparsed tokens where rows represent gold attachment directions and columns show predicted directions. The strong diagonal (LEFT→LEFT: 1179, RIGHT→RIGHT: 1058) confirms the model correctly identifies attachment direction in most error cases. However, within each correct direction, the parser still selects the wrong head - for instance, when it correctly predicts a leftward attachment, it often chooses a head that is too far to the left.

6 Conclusions

We present a joint multitask architecture for unified morphosyntactic parsing that achieves first place in the UniDive 2025 shared task. Our key contribution is explicitly modeling content word identification as a classification task, creating a robust cascade where the identification determines parsing and feature assignment.

Our analysis reveals systematic error patterns pointing to specific improvement opportunities. `Case` confusions concentrate on core grammatical distinctions (`Nom-Acc`), while dependency errors reflect the expected challenges at the `nmod-obl` boundary. While these patterns are linguistically understandable, they suggest potential room for improvement through weighted training or specialized handling of frequently confused categories, though such optimizations may yield only incremental gains.

A more substantial enhancement to the annotation scheme could be making explicit which function words contribute features to which content words. Currently, function words are marked with '_' and their grammatical information is incorporated into "related" content words, but these relationships remain implicit. An indexing system could explicitly link each function word to its target content word. This would not only reduce ambiguity in feature assignment but also make the annotation more transparent for researchers unfamiliar with specific languages, as they could trace exactly how morphosyntactic information flows from function words to content words in the unified representation.

Finally, the 30-point performance gap between Portuguese and Turkish highlights fundamental challenges in handling typologically diverse languages within a unified framework. While the parser excels at the predominantly projective structures, Turkish's agglutinative morphology and flexible word order might be introducing some difficulties. The dispersed error patterns observed for Turkish—contrasting with the concentrated confusions in other languages—suggest that the current architecture may not be optimal for highly non-projective languages. Future work could explore specialized parsing algorithms designed for non-projective structures or alternative architectures that better handle long-distance dependencies and flexible word order. Despite these challenges, our results across nine languages demonstrate the viability of joint morphosyntactic modeling for the task.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative. Demian is funded by the Graduate Research Scholarship from the Faculty of Engineering and Information Technology, University of Melbourne. COST (European Cooperation in Science and Technology) kindly provided funding for travel to the event.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

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. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

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

Martin Haspelmath. 2023. Types of clitics in the world's languages. *Linguistic Typology at the Crossroads*, 3(2):1–59.

Martin Haspelmath. 2025. Grammatical markers and inflectional categories: Reconciling the two perspectives. Draft, Max Planck Institute for Evolutionary Anthropology, April 9, 2025.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

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.

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*.

Alexander M. Rush. 2020. Torch-struct: Deep structured prediction library. *Preprint*, arXiv:2002.00876.

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 ©2025 Association for Computational Linguistics

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:

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

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 nonlinearity (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:

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

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.

Author Index