COMBO: a new module for EUD parsing

Mateusz Klimaszewski[1,2] **Alina Wróblewska**[2]
[1]Warsaw University of Technology
[2]Institute of Computer Science, Polish Academy of Sciences
`m.klimaszewski@ii.pw.edu.pl`
`alina@ipipan.waw.pl`

Abstract

We introduce the COMBO-based approach for EUD parsing and its implementation, which took part in the IWPT 2021 EUD shared task. The goal of this task is to parse raw texts in 17 languages into Enhanced Universal Dependencies (EUD). The proposed approach uses COMBO to predict UD trees and EUD graphs. These structures are then merged into the final EUD graphs. Some EUD edge labels are extended with case information using a single language-independent expansion rule. In the official evaluation, the solution ranked fourth, achieving an average ELAS of 83.79%. The source code is available at `https://gitlab.clarin-pl.eu/syntactic-tools/combo`.

1 Introduction

Data-driven dependency parsers achieve high parsing performance for languages representing different language families. The state-of-the-art dependency parsers are trained with supervised learning methods on large correctly annotated treebanks, e.g. from Universal Dependencies (UD, Nivre et al., 2020). UD is an international initiative aimed at developing a cross-linguistically consistent annotation schema and at building a large multilingual collection of dependency treebanks annotated according to this schema. A relatively small subset of UD treebanks is annotated with higher-order syntactic-semantic representations that encode various linguistic phenomena and are called Enhanced Universal Dependencies (EUD).

Dependency treebanks, especially the uniformly annotated UD treebanks, are used for multilingual system development, e.g. within multiple shared tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007; Seddah et al., 2013, 2014; Zeman et al., 2017, 2018). In particular, the IWPT

2020 shared task on Parsing into Enhanced Universal Dependencies (Bouma et al., 2020) is worth mentioning, because it is the predecessor of the current IWPT 2021 shared task (Bouma et al., 2021). All shared tasks contributed to rapid advancement of language parsing technology, inter alia, the formulation of groundbreaking parsing algorithms and their publicly available implementations (e.g. Nivre et al., 2006; McDonald et al., 2006; Straka and Straková, 2017; Dozat et al., 2017; Rybak and Wróblewska, 2018; He and Choi, 2020).

Dependency parsing is an important issue in various sophisticated downstream tasks, including but not limited to sentiment analysis (Sun et al., 2019), relation extraction (Zhang et al., 2018; Vashishth et al., 2018; Guo et al., 2019), semantic role labelling (Wang et al., 2019), or question answering (Khashabi et al., 2018). On the other hand, even if EUD parsing aims at predicting semantically informed structures, which seem to be appropriate in advanced NLP tasks, it is not yet used in solving these tasks. An obstacle can be the availability of the state-of-the-art EUD parsers, e.g. two top systems at the IWPT 2020 EUD shared task (i.e. Kanerva et al., 2020; Heinecke, 2020) are not publicly available and therefore difficult to integrate into NLU systems without having to implement them from scratch. Meeting the potential expectations of NLU system architects, the source code of COMBO with the new EUD parsing module and the pre-trained models developed as part of our solution submitted to this shared task are publicly available.

The proposed solution to EUD parsing is based on (1) Stanza tokeniser (Qi et al., 2020), (2) COMBO (Klimaszewski and Wróblewska, 2021), a data-driven language-independent system for morphosyntactic prediction, i.e. part-of-speech tagging, morphological analysis, lemmatisation, dependency parsing, and EUD parsing (see Section

3.3), (3) an algorithm that merges predicted labelled dependency arcs and predicted EUD arcs, and builds the final EUD graphs (see Section 3.4), and (4) two linguistically motivated language-independent rules that improve the final EUD graphs (see Section 3.5). The first expansion rule adds case information sublabels to EUD modifiers, and the second one amends enhanced arcs coming into the function words. These two rules are integrated into the proposed EUD parsing system.

In the official evaluation, our EUD parser ranked 4th, obtaining an average ELAS of 83.79% and EULAS of 85.20%.[1] It is worth emphasising that COMBO predicts labelled dependency trees with an average LAS of 88.91%, only being slightly outperformed by the ROBERTNLP system.

2 Shared task description

The IWPT 2021 EUD shared task consists in evaluating systems for parsing raw texts into Enhanced Universal Dependencies. The systems are trained and evaluated on data supplied by the organisers.

Data The shared task dataset includes treebanks for 17 languages from 4 language families. The largest group in this collection is constituted by Indo-European languages, i.e. Bulgarian, Czech, Polish, Russian, Slovak, Ukrainian (Slavic), Dutch, English, Swedish (Germanic), French, Italian (Romance), and Latvian, Lithuanian (Baltic). There are also representatives of the Uralic (Finnic) languages, i.e. Estonian and Finnish, the Afro-Asiatic (Semitic) languages – Arabic, and the Southern Dravidian languages – Tamil. The datasets vary in size and type of enhancements.

Enhancement types Various linguistic phenomena are encoded in EUD graphs:

- propagation of conjuncts in coordination constructions (see Figure 1),

- null nodes encoding elided predicates in coordination constructions (see Figure 2),

- additional subject relations in control and raising constructions (see Figure 3),

- coreference relations in relative clause constructions (see Figure 4),

- detailed case information sublabels of the modifiers (see Figure 5).
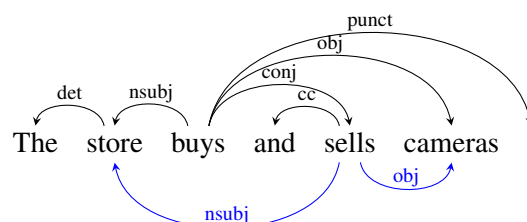

Figure 1: The EUD graph with the conjoined predicate; the conjoined verbs (*buys* and *sells*) share the subject (*the store*) and the object (*cameras*), and the propagated relations are indicated with the bottom blue enhanced edges.
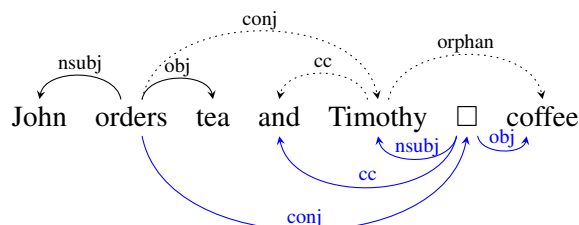

Figure 2: The EUD graph with an empty node □ and the bottom blue enhanced edges. The tree edges removed from the EUD graph are dotted.
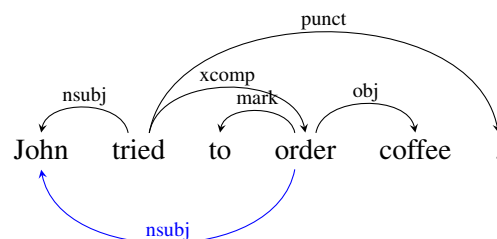

Figure 3: The EUD graph with the bottom blue enhanced edge encoding subject control with the control predicate *try*.

3 System overview

The EUD parsing system is built of the following components: a data encoder boosted with a contextual language model (see Section 3.1), morphosyntactic predictors (see Section 3.2), an EUD predictor (see Section 3.3), an algorithm merging predicted labelled dependency arcs and enhanced dependency arcs (see Section 3.4), and a post-processing module (see Section 3.5).
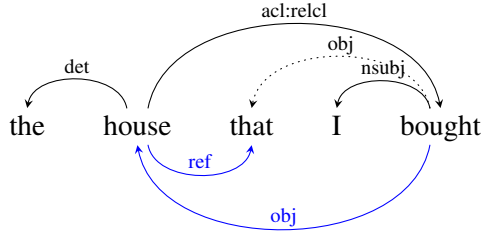
Figure 4: The EUD graph representing a relative clause modifying the noun *house*. The enhanced edges are marked with the bottom blue arcs and the tree edge removed from the EUD graph is dotted.

3.1 Data encoder

The encoder vectorises the tokenised input data. The input tokens are first represented as a concatenation of a character-based word embedding estimated during system training with a dilated convolutional neural network (Yu and Koltun, 2016), and a BERT-based embedding estimated as follows. BERT-based language models (LM, Devlin et al., 2019; Conneau et al., 2020) are not fine-tuned during system training. Instead, we apply the scalar mix technique based on Peters et al. (2018) to produce an embedding (h) for a word i as a weighted sum of embeddings from all layers:

$$h_i = \gamma \sum_{j=1}^{L} s_j h_{ij} \qquad (1)$$

Parameters γ and s_j are learnable weights, additionally s_j are softmax-normalised. L is the number of transformer layers. At the point of using LM, the data is already tokenised. If LM intra-tokeniser splits a word into multiple subwords, the embeddings h are estimated for these subwords and averaged. The vectors of words or averaged vectors of subwords are finally transformed with one fully connected (FC) layer.

The encoder with two BiLSTM layers (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) transforms the concatenations of the character-based word embeddings and the transformed BERT-based embeddings into token vectors. The BiLSTM-transformed token embeddings are used as input to morphosyntactic predictors and the EUD parsing module.

3.2 Morphosyntactic predictors

The proposed approach is based on various morphosyntactic predictions. Part-of-speech tags, morphological features, and lemmata are used in the post-processing step to extract case information expanding enhanced sublabels of modifiers (see Section 3.5). The merge algorithm (see Section 3.4), in turn, combines labelled dependency arcs with enhanced dependency arcs predicted by EUD parsing module.

3.3 EUD predictor

The EUD parsing module consists of an enhanced arc classifier and an enhanced label classifier. The arc classifier utilises two single FC layers that transform encoded token vectors into head and dependent embeddings. These embeddings are used to calculate an adjacency matrix (A) of an enhanced graph. A is a $n \times n$ matrix, where n is the number of tokens in a sentence (plus the ROOT node). The matrix element A_{ij} corresponds to the dot product of the i-th dependent embedding and the j-th head embedding. The dot product indicates the certainty of the edge between two tokens. The sigmoid function, applied to each element of A, allows the network to predict many heads for a given dependent, i.e. EUD graphs are built.

The enhanced label classifier also applies two fully connected layers to estimate head (e_i) and dependent (e_j) embeddings (they differ from embeddings estimated in the enhanced arc prediction). Enhanced dependency labels are predicted by a fully connected layer with the softmax activation function which is given the dependent embedding concatenated with the head embedding.

$$e_{head} = FC(e_i) \qquad (2)$$

$$e_{dep} = FC(e_j) \qquad (3)$$

$$label = argmax(FC(e_{head}, e_{dep})) \qquad (4)$$

The loss function is only propagated for those pairs (i, j) that belong to ground truth (i.e. arcs existing in the enhanced dependency graph).

3.4 Merge algorithm

The predicted enhanced graphs could be used without further processing. However, their quality could definitely be improved if they exploited information from the predicted dependency trees. Enhanced dependency graphs appear to be heavily tree-based (see the example EUD graphs in Section 2). The EUD graphs include some additional edges, empty nodes, and extended labels of modifiers (and conjuncts in some languages), or their

structure is slightly transformed. We therefore decided to merge the predicted trees and the predicted enhanced graphs.

Algorithm 1: The merge algorithm

Input : $T := (V, E_T)$: tree
$\quad\quad\quad G := (V, E_G)$: graph
Output : EUD : the final EUD graph
1   $E_{EUD} = \{\}$;
2   **for** e *in* E_T **do**
3   **if** $label(e) \neq acl{:}relcl$ **then**
4       $E_{EUD} := E_{EUD} + e$;
5   **for** e *in* E_G **do**
6   **if** $e \notin E_{EUD}$ **and**
      $has\_no\_cycle(E_{EUD} + e)$ **then**
7       $E_{EUD} := E_{EUD} + e$;
8   **for** e *in* E_T **do**
9   **if** $label(e) = acl{:}relcl$ **then**
10      $E_{EUD} := E_{EUD} + e$;
11   $EUD := (V, E_{EUD})$;

The merge algorithm (see Algorithm 1) successively adds the predicted tree and graph edges to the set of EUD edges, and then composes the final EUD graph of these edges. It starts by selecting all tree edges except for edges with the *acl:relcl* label. The EUD graphs representing relative clauses contain cycles (see Figure 4). Refraining from adding the *acl:relcl* relations in this step, we attempt to avoid the cycle problem thereafter. In the second step, consecutive graph edges are added to the EUD set as long as they do not form a cycle or there are no edges with the same or a different label in the EUD set (i.e. we eliminate duplicate edges). In the last step, the *acl:relcl* relations are added to the EUD set which is then used to compose a final EUD graph.

We are aware that UD relations selected in the first merging step do not contain case information, e.g. the *obl* relation is transferred to the EUD set, although this relation should be de facto labelled *obl:because_of*, *obl:for*, or *obl:outside*. However, our preliminary experiments indicated that the anticipated enhanced labels often had erroneous case extensions, which could not even come from a sentence. Correcting labels with accidental case extensions would require defining a large number of relabelling rules that would have to be adapted to a particular language. Extending the modifier labels rather than correcting them seems to be a more transparent and simple procedure. We thus define

one rule that derives case information from automatically predicted morphological features and lemmata (see Rule 1 in Section 3.5). The rule is utilised in the post-processing step, which is the last step of building the EUD graphs.

3.5 Post-processing

We define two rules that improve the automatically predicted EUD graphs.

Rule 1   The first rule specifies case information of the following modifiers: *nmod* (nominal modifier), *obl* (oblique nominal), *acl* (clasual modifier of nouns), *advcl* (adverbial clause modifier), and of conjuncts (*conj*). The case information (lemma) is derived from *case/mark* or *cc* dependents of a modifier or a conjunct, respectively, and from the modifier's morphological attribute *Case*. Figure 5 exemplifies extending UD labels with case information.[2]
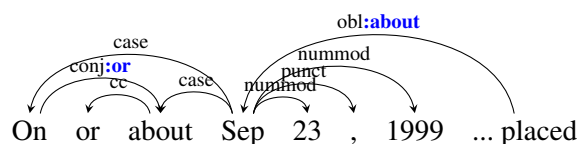

Figure 5: The EUD graph with blue, bolded sublabels representing case information. The text excerpt comes from the sentence "On or about September 23, 1999 a request for service was placed by the above referenced counterparty.".

The rule is language-independent and UD-based. However, as not all treebanks attribute case information to their modifiers or conjuncts, the rule applies only to predefined languages, e.g. the conjunct extension is only valid in English, Italian, Dutch, and Swedish.

Rule 2   The second rule corrects enhanced edges coming into the function words that are labelled *mark*, *punct*, *root*, *case*, *det*, *cc*, *cop*, *aux* and *ref*. They should not be assigned other dependency relation types in EUD graphs. If a token *and* is assigned the *cc* grammatical function in a dependency tree, and thus also in the corresponding EUD graph (the first merge step), it cannot be simultaneously a subject (*nsubj*), for example. If such an erroneous *nsubj* relation exists, it is removed from the EUD graph in line with the second rule.

---

[2]This sentence originates from the English dev set. As the case extension of the *obl* label is derived from the structure coordinating two prepositions (i.e. *on* and *about*), we wonder about correctness of selecting only *about* as the case extension.

Language	Model name	Reference
Arabic	bert-base-arabertv2	Antoun et al. (2020)
English	bert-base-cased	Devlin et al. (2019)
French	camembert-base	Martin et al. (2020)
Finnish	bert-base-finnish-cased-v1	Virtanen et al. (2019)
Polish	herbert-large-cased	Mroczkowski et al. (2021)
Others	xlm-roberta-large	Conneau et al. (2020)

Table 1: Language models used in the experiments. Names refer to Transformers library (Wolf et al., 2020).

4 Experimental setup

4.1 Segmentation and preprocessing

Stanza tokeniser (Qi et al., 2020) is used to split raw text into sentences, split sentences into tokens, and optionally to expand multi-words. We train a new segmentation model for each language on the training data provided in the shared task.[3] Whenever there are several UD treebanks for a language, we train the segmentation model on the concatenation of all training datasets available for that language. Multi-word expansion involves only two languages, i.e. Arabic and Tamil, because it does not cause substantial gains in parsing other languages.

In order to collapse empty nodes, training data are preprocessed with the official UD script.[4] Dependents of the collapsed empty nodes are assigned new labels, corresponding to the empty node label and the dependent label joined with the special symbol $>$. During prediction, the collapsed labels are expanded and empty nodes are added at the end of a sentence, following He and Choi (2020). This design decision is motivated by the fact that (1) it is difficult to find a proper position of elided tokens or phrases, especially in free word order languages, and (2) the evaluation procedure does not take an empty node position into account, i.e. appending an empty node at the end of a sentence does not downgrade the score. It is important to note that designing a heuristic that identifies proper positions of elided elements remains an open issue, and appending empty nodes at the end of a sentence is only a makeshift solution.

Input data are encoded using BERT-based language models. Depending on the language, either language-specific BERT (Devlin et al., 2019) or multilingual XLM-R (Conneau et al., 2020) is used (see Table 1).

4.2 Morphosyntactic prediction

COMBO system (Klimaszewski and Wróblewska, 2021) is used to predict part-of-speech tags, morphological features, lemmata, and dependency trees. For the purpose of this task, we also implement a new EUD parsing module (see Section 3.3) and integrate it with COMBO. Similarly to segmentation models, we train one COMBO model for a language on all treebanks provided for this language in the shared task data using the default training parameters (see Table 2).[5]

Hyperparameter	Value
Optimiser	Adam (Kingma and Ba, 2015)
Learning rate	0.002
β_1 and β_2	0.9
Number of epochs	400
BiLSTM layers	2
BiLSTM dropout rate	0.33
LSTM hidden size	512
Arc projection size	512
Label projection size	128

Table 2: COMBO training parameters (the upper entries) and model parameters (the bottom entries).

5 Results

The shared task submissions are evaluated with two evaluation metrics: ELAS – LAS[6] on enhanced de-

---

[3]It is not allowed to use versions of UD other than 2.7 in the IWPT 2021 shared task (see https://universaldependencies.org/iwpt21/task_and_evaluation.html). As the publicly available Stanza models are trained on UD 2.5, we have to train new models on UD 2.7.

[4]https://github.com/UniversalDependencies/tools/blob/master/enhanced_collapse_empty_nodes.pl

[5]All models are trained and tested on a single NVIDIA V100 card.

[6]LAS (labelled attachment score) is the proportion of tokens that are assigned the correct head and dependency label

pendencies, and EULAS – LAS on enhanced dependencies where labels are restricted to the UD relation types, i.e. sublabels are ignored. COMBO ranks 4th, achieving 84.71% ELAS in the qualitative evaluation (an average over treebanks), and 83.79% ELAS in the coarse evaluation (an average over languages). In terms of EULAS, it ranks 4th achieving 86.30% in the qualitative evaluation, and 5th achieving 85.20% in the coarse evaluation. In addition to ELAS and EULAS metrics, the systems are also compared in terms of quality of predicting labelled dependency trees measured with LAS (the secondary evaluation measure). In the LAS ranking, COMBO takes second place achieving 88.91% in the qualitative evaluation, and 87.84% in the coarse evaluation, being slightly overcome by the ROBERTNLP system (89.25% in the qualitative evaluation, and 89.18% in the coarse evaluation). Table 3 presents the official results of COMBO models per language.

Language	LAS	EULAS	ELAS
Arabic	81.04	78.35	76.39
Bulgarian	89.52	87.41	86.67
Czech	93.30	90.57	89.08
Dutch	90.93	88.90	87.07
English	87.22	85.27	84.09
Estonian	87.53	85.56	84.02
Finnish	92.28	88.79	87.28
French	89.29	88.10	87.32
Italian	93.27	91.16	90.40
Latvian	90.25	86.22	84.57
Lithuanian	84.75	81.28	79.75
Polish	92.75	90.22	87.65
Russian	94.29	91.76	90.73
Slovak	91.72	88.53	87.04
Swedish	87.82	85.26	83.20
Tamil	56.28	53.49	52.27
Ukrainian	90.96	87.60	86.92
Average	**87.84**	**85.20**	**83.79**

Table 3: The official evaluation results per language.

Post-processing impact We measure the impact of the post-processing step (i.e. extending graph labels with case information and correcting edges coming into the function words) on the development data per language (see Table 4). Following the training approach, we concatenate the datasets

according to the gold standard.

Language	Before		After	
	EULAS	ELAS	EULAS	ELAS
Arabic	77.46	57.32	77.89	76.40
Bulgarian	89.50	78.97	90.29	89.30
Czech	89.93	74.96	91.28	89.91
Dutch	87.96	76.22	88.94	87.64
English	85.13	74.40	85.49	84.30
Estonian	86.27	68.73	86.92	85.45
Finnish	86.98	72.08	87.92	86.44
French	90.48	89.99	91.10	90.62
Italian	89.84	75.47	91.10	90.31
Latvian	85.65	73.72	86.44	84.88
Lithuanian	82.37	63.56	83.41	82.32
Polish	90.08	77.97	90.64	87.64
Russian	90.43	75.93	91.03	90.10
Slovak	87.89	71.71	89.39	87.90
Swedish	85.62	73.59	86.09	84.07
Tamil	54.35	40.48	54.84	53.38
Ukrainian	88.30	73.51	89.13	88.52

Table 4: Impact of the post-processing step.

Language	Sentences		Tokens	
	TGIF	Stanza	TGIF	Stanza
Arabic	96.87	79.92	99.99	99.97
Dutch	94.32	83.82	99.90	99.89
Lithuanian	96.22	87.74	99.99	99.81
Swedish	99.03	93.64	99.86	99.44

Table 5: The quality of TGIF and Stanza segmentation in the selected languages.

if a language has multiple treebanks. The second rule modifies the graph structure. However, as the EULAS scores are almost negligible, using this rule seems questionable. The first rule, in turn, does not modify the structure of EUD graphs, but only their edge labels, and its impact on improving ELAS scores is significant.

Segmentation drawback The official evaluation results show significant discrepancies in the quality of tokenisation and sentence segmentation. The highest differences in sentence segmentation between TGIF, the winner of the shared task, and Stanza used in our approach are shown in Table 5. For example, there is a loss of more than 15 percentage points in sentence segmentation of the Arabic texts. We therefore decide to investigate the impact of the quality of sentence segmentation and tokenisation on the final results. For this purpose,

Language	LAS	EULAS	ELAS
Arabic	81.04 (+4.51)	78.35 (+4.3)	76.39 (+4.24)
Dutch	90.93 (+1.52)	88.90 (+1.61)	87.07 (+1.59)
Lithuanian	84.75 (+1.35)	81.28 (+1.34)	79.75 (+1.31)
Swedish	87.82 (+1.24)	85.26 (+1.21)	83.20 (+1.17)

Table 6: Performance gain in predicting UD trees and EUD graphs of gold-standard tokanised test sentences from the languages with the worst segmentation quality. The values in brackets show the improvement over the baseline (i.e. Stanza tokenisation).

we conduct an additional experiment consisting in predicting EUD graphs on the test data with gold-standard tokenisation and sentence segmentation. The results of this experiment show a gain of around 1.5 pp for all tested languages except Arabic with the gain over 4 pp (see Table 6).

6 Conclusion

We presented the COMBO-based solution to EUD parsing which took part in the IWPT 2021 EUD shared task. The proposed approach is hybrid, i.e. based on machine learning and rule-based algorithms. First, UD trees and EUD graphs (and also morphosyntactic features of tokens, i.e. parts of speech, morphological features, and lemmata) are automatically predicted with the data-driven COMBO system. Then, the predicted structures are combined into the EUD graphs using the developed rule-based merge algorithm. Finally, the labels of modifiers and conjuncts in the merged EUD graphs are extended with case information using an expansion rule. The proposed solution is simple and language-independent. We recognise that we could still improve the results, e.g. by defining language-specific correction rules. However, our objective was to build an easy-to-use system for predicting EUD graphs that is publicly available and can be efficiently use to solve sophisticated NLU tasks.

Acknowledgments

References

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Gosse Bouma, Djamé Seddah, and Daniel Zeman. 2020. Overview of the IWPT 2020 shared task on parsing into enhanced Universal Dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 151–161, Online. Association for Computational Linguistics.

Gosse Bouma, Djamé Seddah, and Daniel Zeman. 2021. From raw text to enhanced universal dependencies: The parsing shared task at iwpt 2021. In *Proceedings of the 17th International Conference on Parsing Technologies (IWPT 2021)*, pages 146–157, Bangkok, Thailand (online). Association for Computational Linguistics.

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. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency

parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5):602–610.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Han He and Jinho D. Choi. 2020. Adaptation of multilingual transformer encoder for robust enhanced Universal Dependency parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 181–191. Association for Computational Linguistics.

Johannes Heinecke. 2020. Hybrid enhanced Universal Dependencies parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 174–180, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. 2020. Turku enhanced parser pipeline: From raw text to enhanced graphs in the IWPT 2020 shared task. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 162–173, Online. Association for Computational Linguistics.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mateusz Klimaszewski and Alina Wróblewska. 2021. The COMBO package [online]. Available: https://gitlab.clarin-pl.eu/syntactic-tools/combo, Institute of Computer Science, PAS, Accessed on: June 2, 2021.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot.

2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 216–220. Association for Computational Linguistics.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine. Association for Computational Linguistics.

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.

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 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Piotr Rybak and Alina Wróblewska. 2018. Semi-supervised neural system for tagging, parsing and lemmatization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 45–54. Association for Computational Linguistics.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109, Dublin, Ireland. Dublin City University.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-level sentiment analysis via convolution over dependency tree. 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 5679–5688, Hong Kong, China. Association for Computational Linguistics.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266. Association for Computational Linguistics.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: BERT for finnish. *CoRR*, abs/1912.07076.

Yufei Wang, Mark Johnson, Stephen Wan, Yifang Sun, and Wei Wang. 2019. How to best use syntax in semantic role labelling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5338–5343, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. *CoRR*, abs/1511.07122.

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, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. 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.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215. Association for Computational Linguistics.