

COMBO: State-of-the-Art Morphosyntactic Analysis

Mateusz Klimaszewski^{1,2} Alina Wróblewska²

¹Warsaw University of Technology

²Institute of Computer Science, Polish Academy of Sciences

m.klimaszewski@ii.pw.edu.pl

alina@ipipan.waw.pl

Abstract

We introduce COMBO – a fully neural NLP system for accurate part-of-speech tagging, morphological analysis, lemmatisation, and (enhanced) dependency parsing. It predicts categorical morphosyntactic features whilst also exposes their vector representations, extracted from hidden layers. COMBO is an easy to install Python package with automatically downloadable pre-trained models for over 40 languages. It maintains a balance between efficiency and quality. As it is an end-to-end system and its modules are jointly trained, its training is competitively fast. As its models are optimised for accuracy, they achieve often better prediction quality than SOTA. The COMBO library is available at: <https://gitlab.clarin-pl.eu/syntactic-tools/combo>.

1 Introduction

Natural language processing (NLP) has long recognised morphosyntactic features as necessary for solving advanced natural language understanding (NLU) tasks. An enormous impact of contextual language models on presumably all NLP tasks has slightly weakened the importance of morphosyntactic analysis. As morphosyntactic features are encoded to some extent in contextual word embeddings (e.g. Tenney et al., 2019; Lin et al., 2019), doubts arise as to whether explicit morphosyntactic knowledge is still needed. For example, Glavaš and Vulić (2021) have recently investigated an intermediate fine-tuning contextual language models on the dependency parsing task and suggested that this step does not significantly contribute to advance NLU models. Conversely, Warstadt et al. (2019) reveal the powerlessness of contextual language models in encoding linguistic phenomena like negation. This is in line with our intuition about representing negation in Polish sentences (see Figure 1). It does not seem trivial to differentiate between the con-

tradicting meanings of these sentences using contextual language models, as the word context is similar. The morphosyntactic features, e.g. parts of speech PART vs. INTJ, and dependency labels *advmod:neg* vs. *discourse:intj*, could be beneficial in determining correct reading.

In order to verify the influence of explicit morphosyntactic knowledge on NLU tasks, it is necessary to design a technique for injecting this knowledge into models or to build morphosyntax-aware representations. The first research direction was initiated by Glavaš and Vulić (2021). Our objective is to provide a tool for predicting high-quality morphosyntactic features and exposing their embeddings. These vectors can be directly combined with contextual word embeddings to build morphosyntactically informed word representations.

The emergence of publicly available NLP datasets, e.g. Universal Dependencies (Zeman et al., 2019), stimulates the development of NLP systems. Some of them are optimised for efficiency, e.g. spaCy (Honnibal et al., 2020), and other for accuracy, e.g. UDPipe (Straka, 2018), the Stanford system (Dozat and Manning, 2018), Stanza (Qi et al., 2020). In this paper, we introduce COMBO, an open-source fully neural NLP system which is optimised for both training efficiency and prediction quality. Due to its end-to-end architecture, which is an innovation within morphosyntactic analysers, COMBO is faster in training than the SOTA pipeline-based systems, e.g. Stanza. As a result of applying modern NLP solutions (e.g. contextualised word embeddings), it qualitatively outperforms other systems.

COMBO analyses tokenised sentences and predicts morphosyntactic features of tokens (i.e. parts of speech, morphological features, and lemmata) and syntactic structures of sentences (i.e. dependency trees and enhanced dependency graphs). At the same time, its module, COMBO-vectoriser, extracts vector representations of the predicted fea-

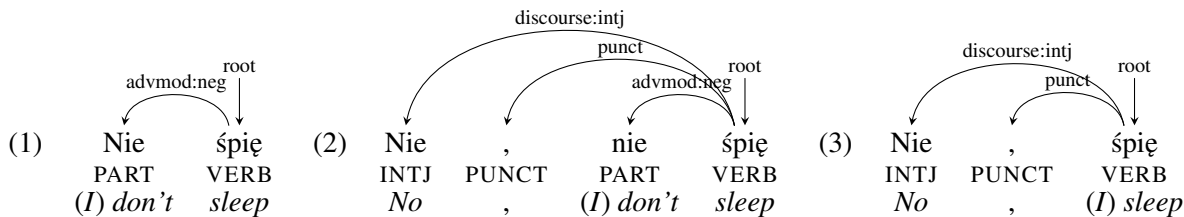


Figure 1: UD trees of Polish sentences: (1) and (2) mean a *non-sleeping* situation and (3) means *sleeping*.

tures from hidden layers of individual predictors. COMBO user guide is in §4 and a live demo is available on the website <http://combo-demo.nlp.ipipan.waw.pl>.

Contributions 1) We implement COMBO (§2), a fully neural NLP system for part-of-speech tagging, morphological analysis, lemmatisation, and (enhanced) dependency parsing, together with COMBO-vectoriser for revealing vector representations of predicted categorical features. COMBO is implemented as a Python package which is easy to install and to integrate into a Python code. 2) We pre-train models for over 40 languages that can be automatically downloaded and directly used to process new texts. 3) We evaluate COMBO and compare its performance with two state-of-the-art systems, spaCy and Stanza (§3).

2 COMBO Architecture

COMBO’s architecture (see Figure 2) is based on the forerunner (Rybak and Wróblewska, 2018) implemented in the Keras framework. Apart from a new implementation in the PyTorch library (Paszke et al., 2019), the novelties are the BERT-based encoder, the EUD prediction module, and COMBO-vectoriser extracting embeddings of UPOS and DEPREL from the last hidden layers of COMBO’s tagging and dependency parsing module, respectively. This section provides an overview of COMBO’s modules. Implementation details are in Appendix A.

Local Feature Extractors Local feature extractors (see Figure 2) encode categorical features (i.e. words, parts of speech, morphological features, lemmata) into vectors. The feature bundle is configurable and limited by the requirements set for COMBO. For instance, if we train only a dependency parser, the following features can be input to COMBO: internal character-based word embeddings (CHAR), pre-trained word embeddings (WORD), and embeddings of lemmata (LEMMA),

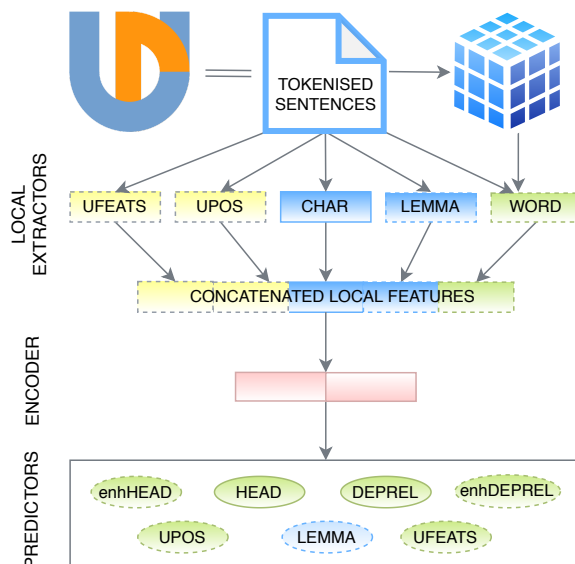
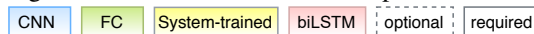


Figure 2: COMBO architecture. Explanations:



parts of speech (UPOS) and morphological features (UFEATS). If we train a morphosyntactic analyser (i.e. tagger, lemmatiser and parser), internal word embeddings (CHAR) and pre-trained word embeddings (WORD), if available, are input to COMBO.

Words and lemmata are always encoded using character-based word embeddings (CHAR and LEMMA) estimated during system training with a dilated convolutional neural network (CNN) encoder (Yu and Koltun, 2016; Strubell et al., 2017).

Additionally, words can be represented using pre-trained word embeddings (WORD), e.g. fastText (Grave et al., 2018), or BERT (Devlin et al., 2019). The use of pre-trained embeddings is an optional functionality of the system configuration. COMBO freezes pre-trained embeddings (i.e. no fine-tuning) and uses their transformations, i.e. embeddings are transformed by a single fully connected (FC) layer.

Part-of-speech and morphological embeddings (UPOS and UFEATS) are estimated during system training. Since more than one morphological feature can attribute a word, embeddings of all pos-

sible features are estimated and averaged to build a final morphological representation.

Global Feature Encoder The encoder uses concatenations of local feature embeddings. A sequence of these vectors representing all the words in a sentence is processed by a bidirectional LSTM (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005). The network learns the context of each word and encodes its global (contextualised) features (see Figure 3). Global feature embeddings are input to the prediction modules.

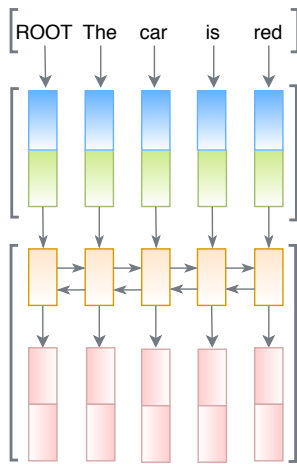


Figure 3: Estimation of global feature vectors.

biLSTM GLOBAL

Tagging Module The tagger takes global feature vectors as input and predicts a universal part of speech (UPOS), a language-specific tag (XPOS), and morphological features (UFEATS) for each word. The tagger consists of two linear layers followed by a softmax. Morphological features build a disordered set of category-value pairs (e.g. Number=Plur). Morphological feature prediction is thus implemented as several classification problems. The value of each morphological category is predicted with a FC network. Different parts of speech are assigned different sets of morphological categories (e.g. a noun can be attributed with grammatical gender, but not with grammatical tense). The set of possible values is thus extended with the NA (not applicable) symbol. It allows the model to learn that a particular category is not a property of a word.

Lemmatisation Module The lemmatiser uses an approach similar to character-based word embedding estimation. A character embedding is concatenated with the global feature vector and transformed by a linear layer. The lemmatiser takes a sequence of such character representations and trans-

forms it using a dilated CNN. The softmax function over the result produces the sequence of probabilities over a character vocabulary to form a lemma.

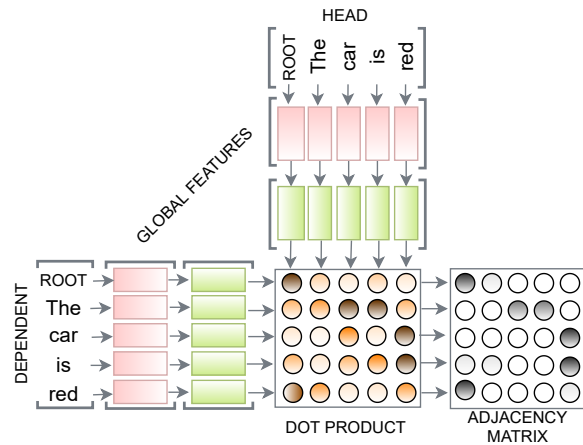


Figure 4: Prediction of dependency arcs.

Parsing Module Two single FC layers transform global feature vectors into head and dependent embeddings (see Figure 4). Based on these representations, a dependency graph is defined as an adjacency matrix with columns and rows corresponding to heads and dependents, respectively. The adjacency matrix elements are dot products of all pairs of the head and dependent embeddings (the dot product determines the certainty of the edge between two words). The softmax function applied to each row of the matrix predicts the adjacent head-dependent pairs. This approach, however, does not guarantee that the resulting adjacency matrix is a properly built dependency tree. The Chu-Liu-Edmonds algorithm (Chu and Liu, 1965; Edmonds, 1967) is thus applied in the last prediction step.

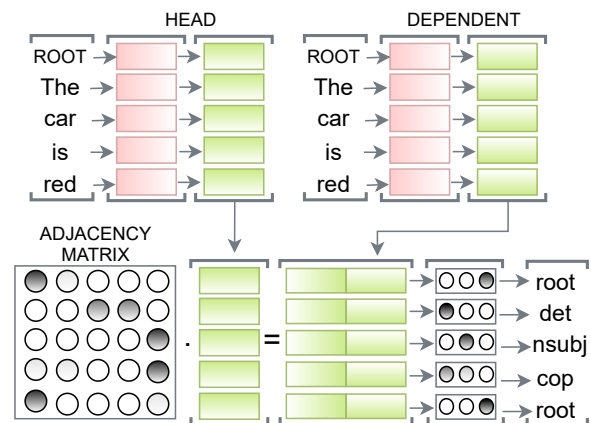


Figure 5: Prediction of grammatical functions.

The procedure of predicting words’ grammatical functions (aka dependency labels) is shown in Figure 5. A dependent and its head are represented as vectors by two single FC layers. The dependent embedding is concatenated with the weighted average of (hypothetical) head embeddings. The weights are the values from the corresponding row of the adjacency matrix, estimated by the arc prediction module. Concatenated vector representations are then fed to a FC layer with the softmax activation function to predict dependency labels.

EUD Parsing Module Enhanced Universal Dependencies (EUD) are predicted similarly to dependency trees. The EUD parsing module is described in details in Klimaszewski and Wróblewska (2021).

3 COMBO Performance

Data COMBO is evaluated on treebanks from the Universal Dependencies repository (Zeman et al., 2019), preserving the original splits into training, validation, and test sets. The treebanks representing distinctive language types are summarised in Table 4 in Appendix B.

By default, pre-trained 300-dimensional fastText embeddings (Grave et al., 2018) are used. We also test encoding data with pre-trained contextual word embeddings (the tested BERT models are listed in Table 5 in Appendix B). The UD datasets provide gold-standard tokenisation. If BERT intra-tokeniser splits a word into sub-words, the last layer embeddings are averaged to obtain a single vector representation of this word.

Qualitative Evaluation Table 1 shows COMBO results of processing the selected UD treebanks.¹ COMBO is compared with Stanza (Qi et al., 2020) and spaCy.² The systems are evaluated with the standard metrics (Zeman et al., 2018): F1, UAS (unlabelled attachment score), LAS (labelled attachment score), MLAS (morphology-aware LAS) and BLEX (bi-lexical dependency score).³

COMBO and Stanza undeniably outrun spaCy models. COMBO using non-contextualised word

embeddings is outperformed by Stanza in many language scenarios. However, COMBO supported with BERT-like word embeddings beats all other solutions and is currently the SOTA system for morphosyntactic analysis.

Regarding lemmatisation, Stanza has an advantage over COMBO in most tested languages. This is probably due to the fact that Stanza lemmatiser is enhanced with a key-value dictionary, whilst COMBO lemmatiser is fully neural. It is not surprising that a dictionary helps in lemmatisation of isolating languages (English). However, the dictionary approach is also helpful for agglutinative languages (Finnish, Korean, Basque) and for Arabic, but not for Polish (fusional languages). Comparing COMBO models estimated with and without BERT embeddings, we note that BERT boost only slightly increases the quality of lemma prediction in the tested fusional and agglutinative languages.

For a complete insight into the prediction quality, we evaluate individual UPOS and UDEPREL predictions in English (the isolating language), Korean (agglutinative) and Polish (fusional). Result visualisations are in Appendix C.

COMBO took part in IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (Bouma et al., 2021), where it ranked 4th.⁴ In addition to ELAS and EULAS metrics, the third evaluation metric was LAS. COMBO ranked 2nd, achieving the average LAS of 87.84%. The score is even higher than the average LAS of 86.64% in Table 1, which is a kind of confirmation that our evaluation is representative, reliable, and fair.

Downstream Evaluation According to the results in Table 1, COMBO predicts high-quality dependency trees and parts of speech. We therefore conduct a preliminary evaluation of morphosyntactically informed word embeddings in the textual entailment task (aka natural language inference, NLI) in English (Bentivogli et al., 2016) and Polish (Wróblewska and Krasnowska-Kieraś, 2017). We compare the quality of entailment classifiers with two FC layers trained on max/mean-pooled BERT embeddings and sentence representations estimated by a network with two transformer layers which is given morphosyntactically informed word embeddings (i.e. BERT-based word embeddings concatenated with UPOS embeddings, DEPREL embeddings, and BERT-based embeddings of the head

¹Check the prediction quality for other languages at: <https://gitlab.clarin-pl.eu/syntactic-tools/combo/-/blob/master/docs/performance.md>.

²<https://spacy.io> We use the project template https://github.com/explosion/projects/tree/v3/pipelines/tagger_parser_ud. The lemmatiser is implemented as a standalone pipeline component in spaCy v3 and we do not test it.

³http://universaldependencies.org/conll118/conll118_ud_eval.py (CoNLL 2018 evaluation script).

⁴<https://universaldependencies.org/iwpt21/results.html>

System	UPOS	XPOS	UFeat	Lemma	UAS	LAS	CLAS	MLAS	BLEX
English EWT (isolating)									
spaCy	93.79	93.10	94.89	NA	83.38	79.76	75.74	68.91	NA
Stanza	96.36	96.15	97.01	98.18	89.64	86.89	83.84	79.44	82.03
COMBO	95.60	95.21	96.60	97.43	88.56	85.58	82.35	76.56	79.78
COMBO _{BERT}	96.57	96.44	97.24	97.86	91.76	89.28	86.83	81.71	84.38
Arabic PADT (fusional)									
spaCy	90.27	82.15	82.70	NA	74.24	67.28	63.28	50.48	NA
Stanza	96.98	93.97	94.08	95.26	87.96	83.74	80.57	74.96	76.80
COMBO	96.71	93.72	93.83	93.54	87.06	82.70	79.46	73.25	73.64
COMBO _{BERT}	97.04	94.83	95.05	93.95	89.21	85.09	82.36	76.82	76.67
Polish PDB (fusional)									
spaCy	96.14	86.94	87.41	NA	86.73	82.06	79.00	65.42	NA
Stanza	98.47	94.20	94.42	97.43	93.15	90.84	88.73	81.98	85.75
COMBO	98.24	94.26	94.53	97.47	92.87	90.45	88.07	81.31	85.53
COMBO _{BERT}	98.97	96.54	96.80	98.06	95.60	93.93	92.34	87.59	89.91
Finnish TDT (agglutinative)									
spaCy	92.15	93.34	87.89	NA	80.06	74.75	71.52	61.95	NA
Stanza	97.24	97.96	95.58	95.24	89.57	87.14	85.52	80.52	81.05
COMBO	96.72	98.02	94.04	88.73	89.73	86.70	84.56	77.63	72.42
COMBO _{BERT}	98.29	99.00	97.30	89.48	94.11	92.52	91.34	87.18	77.84
Korean Kaist (agglutinative)									
spaCy	85.21	72.33	NA	NA	76.15	68.13	61.98	57.52	NA
Stanza	95.45	86.31	NA	93.02	88.42	86.39	83.97	80.64	77.59
COMBO	94.46	81.66	NA	89.16	87.31	85.12	82.70	78.38	72.79
COMBO _{BERT}	95.89	85.16	NA	89.95	89.77	87.83	85.96	82.66	75.89
Turkish IMST (agglutinative)									
spaCy	87.66	86.18	82.26	NA	60.43	51.32	47.74	37.28	NA
Stanza	95.98	95.18	93.77	96.73	74.14	67.52	64.03	58.13	61.91
COMBO	93.60	92.36	88.88	96.47	72.00	64.48	60.48	49.88	58.75
COMBO _{BERT}	95.14	94.27	93.56	97.54	78.53	72.03	68.88	60.55	67.13
Basque BDT (agglutinative with fusional verb morphology)									
spaCy	91.96	NA	86.67	NA	76.11	70.28	66.96	54.46	NA
Stanza	96.23	NA	93.09	96.52	86.19	82.76	81.30	73.56	78.27
COMBO	94.28	NA	90.44	95.47	84.64	80.44	78.82	67.33	74.95
COMBO _{BERT}	96.26	NA	93.84	96.38	88.73	85.80	84.93	75.96	81.25
Average scores									
spaCy	91.03	85.67	86.97	NA	76.73	70.51	66.60	56.57	NA
Stanza	96.67	93.96	94.66	96.05	87.01	83.61	81.14	75.60	77.63
COMBO	95.66	92.54	93.05	94.04	86.02	82.21	79.49	72.05	73.98
COMBO _{BERT}	96.88	94.37	95.63	94.75	89.67	86.64	84.66	78.92	79.01

Table 1: Processing quality (F_1 scores) of spaCy, Stanza and COMBO on the selected UD treebanks (the language types are given in parentheses). The highest scores are marked in bold.

Treebank	spaCy	Stanza				COMBO	
		Tagger	Lemmatiser	Parser	Total	fastText	BERT
English EWT	00:22:34	02:08:51	02:12:17	02:29:13	06:50:21	01:26:55	1:54:11
Polish PDB	01:07:55	04:36:51	03:19:04	05:08:41	13:04:36	02:39:44	3:31:41

Table 2: Training time of spaCy, Stanza and COMBO.

word). The morphosyntactically informed English NLI classifier achieves an accuracy of 78.84% and outperforms the max/mean-pooled classifiers by

20.77 pp and 5.44 pp, respectively. The Polish syntax-aware NLI classifier achieves an accuracy of 91.60% and outperforms the max/mean-pooled

classifiers by 17.2 pp and 7.7 pp, respectively.

Efficiency Evaluation We also compare spaCy, Stanza and COMBO in terms of their efficiency, i.e. training and prediction speed.⁵ According to the results (see Tables 2 and 3), spaCy is the SOTA system, and the other two are not even close to its processing time. Considering COMBO and Stanza, whose prediction quality is significantly better than spaCy, COMBO is 1.5 times slower (2 times slower with BERT) than Stanza in predicting, but it is definitely faster in training. The reason for large discrepancies in training times is the different architecture of these two systems. Stanza is a pipeline-based system, i.e. its modules are trained one after the other. COMBO is an end-to-end system, i.e. its modules are jointly trained and the training process is therefore faster.

Treebank	Stanza	COMBO	COMBO _{BERT}
English EWT	4.7×	6.8×	10.8×
Polish PDB	4.1×	5.8×	10.6×

Table 3: Prediction time of Stanza and COMBO relative to spaCy (1×) on English and Polish test data.

4 Getting Started with COMBO

Prediction COMBO provides two main prediction modes: a Python library and a command-line interface (CLI). The Python package mode supports automated model download. The code snippet demonstrates downloading a pre-trained Polish model and processing a sentence:

```
from combo.predict import COMBO

nlp = COMBO.from_pretrained("polish")
sentence = nlp("Ala ma kota.")
print(sentence.tokens)
```

To download a model for another language, select its name from the list of pre-trained models.⁶ The Python mode also supports acquisition of DEPREL or UPOS embeddings, for example:

```
sentence = nlp("Ala ma kota.")
chosen_token = sentence.tokens[1]
print(chosen_token.embeddings["upostag"])
```

⁵A single NVIDIA V100 card is used in all tests.

⁶The list of the pretrained COMBO models: <https://gitlab.clarin-pl.eu/syntactic-tools/combo/-/blob/master/docs/models.md#pre-trained-models>

In CLI mode, COMBO processes sentences using either a downloaded model or a model trained by yourself. CLI works on raw texts and on the CoNLL-U files (i.e. with tokenised sentences and even morphologically annotated tokens):

```
combo --mode predict \
      --model_path model.tar.gz \
      --input_file input.conllu \
      --output_file output.conllu
```

Model Training COMBO CLI allows to train new models for any language. The only requirement is a training dataset in the CoNLL-U/CoNLL-X format. In the default setup, tokenised sentences are input and all possible predictors are trained:

```
combo --mode train \
      --training_data training.conllu \
      --validation_data valid.conllu
```

If we only train a dependency parser, the default setup should be changed with configuration flags: `--features` with a list of input features and `--targets` with a list of prediction targets.

5 Conclusion

We have presented COMBO, the SOTA system for morphosyntactic analysis, i.e. part-of-speech tagging, morphological analysis, lemmatisation, and (enhanced) dependency parsing. COMBO is a language-agnostic and format-independent system (i.e. it supports the CoNLL-U and CoNLL-X formats). Its implementation as a Python package allows effortless installation, and incorporation into any Python code or usage in the CLI mode. In the Python mode, COMBO supports automated download of pre-trained models for multiple languages and outputs not only categorical morphosyntactic features, but also their embeddings. In the CLI mode, pre-trained models can be manually downloaded or trained from scratch. The system training is fully configurable in respect of the range of input features and output predictions, and the method of encoding input data.

Last but not least, COMBO maintains a balance between efficiency and quality. Admittedly, it is not as fast as spaCy, but it is much more efficient than Stanza considering the training time. Tested on the selected UD treebanks, COMBO morphosyntactic models enhanced with BERT embeddings outperform spaCy and Stanza models.

Acknowledgments

The authors would like to thank Piotr Rybak for his design and explanations of the architecture of COMBO's forerunner. The research presented in this paper was funded by SONATA 8 grant no 2014/15/D/HS2/03486 from the National Science Centre Poland and the European Regional Development Fund as a part of the 2014-2020 Smart Growth Operational Programme, CLARIN – Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19. The computing was performed at Poznań Supercomputing and Networking Center.

References

- Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. [Give your text representation models some love: the case for Basque](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France. European Language Resources Association.
- 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.
- Maria Jesus Aranzabe, Aitziber Atutxa, Kepa Bengoetxea, Arantza Diaz de Ilarraza, Iakes Goenaga, Koldo Gojenola, and Larraitz Uria. 2015. Automatic Conversion of the Basque Dependency Treebank to Universal Dependencies. In *Proceedings of the 14th International Workshop on Treebanks and Linguistic Theories (TLT14)*, pages 233–241.
- Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2016. [SICK through the SemEval Glasses. Lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment](#). *Journal of Language Resources and Evaluation*, 50:95–124.
- 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 and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 146–157, Online. Association for Computational Linguistics.
- Y. J. Chu and T. H. Liu. 1965. On the Shortest Arborescence of a Directed Graph. *Science Sinica*, 14:1396–1400.
- Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. 2018. [Building Universal Dependency treebanks in Korean](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- 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 and Christopher D. Manning. 2018. [Simpler but more accurate semantic dependency parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.
- Jack Edmonds. 1967. Optimum Branchings. *Journal of Research of the National Bureau of Standards*, 71B(4):233–240.
- Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16*, page 1027–1035, Red Hook, NY, USA. Curran Associates Inc.
- 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.
- Goran Glavaš and Ivan Vulić. 2021. [Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online. Association for Computational Linguistics.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Alex Graves and Jürgen Schmidhuber. 2005. [Frame-wise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures](#). *Neural Networks*, 18(5):602–610.

- Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel Beška, Jakub Kracmar, and Kamila Hassanová. 2009. [Prague arabic dependency treebank 1.0](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. [Building the essential resources for finnish: The turku dependency treebank](#). *Lang. Resour. Eval.*, 48(3):493–531.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Computation*, 9(8):1735–1780.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spaCy: Industrial-strength Natural Language Processing in Python](#).
- Kiyoung Kim. 2020. Pretrained Language Models For Korean. <https://github.com/kiyoungkim1/LMkor>.
- 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. [COMBO: A new module for EUD parsing](#). In *Proceedings of the 17th International Conference on Parsing Technologies and the IWPT 2021 Shared Task on Parsing into Enhanced Universal Dependencies (IWPT 2021)*, pages 158–166, Online. Association for Computational Linguistics.
- Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. [Open sesame: Getting inside BERT’s linguistic knowledge](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. 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.
- Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML’10*, page 807–814, Madison, WI, USA. Omnipress.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [PyTorch: An Imperative Style, High-Performance Deep Learning Library](#). In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- 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 Lematization](#). In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 45–54.
- Stefan Schweter. 2020. [BERTurk - BERT models for Turkish](#).
- Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. [A gold standard dependency corpus for English](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: A Simple Way to Prevent Neural Networks from Overfitting](#). *Journal of Machine Learning Research*, 15:1929–1958.
- 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.
- Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. [Fast and Accurate Entity Recognition with Iterated Dilated Convolutions](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.
- Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. 2016. [Universal Dependencies for Turkish](#). In *Proceedings of COLING 2016, the 26th International*

- Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan. The COLING 2016 Organizing Committee.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. [What do you learn from context? probing for sentence structure in contextualized word representations](#). In *International Conference on Learning Representations*.
- 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](#).
- Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohanane, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019. [Investigating BERT’s knowledge of language: Five analysis methods with NPIs](#). 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 2877–2887, Hong Kong, China. 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.
- Alina Wróblewska and Katarzyna Krasnowska-Kieras. 2017. [Polish evaluation dataset for compositional distributional semantics models](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 784–792, Vancouver, Canada. 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, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielé Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mițitelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Ben-goetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čěplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droганova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyan Ephrem, Olga Erina, Tomáš Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Groni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mý, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Olájidé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Mackentanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Mieška, Margarita Misirpashayeva, Anna Missilä, Cătălin Mițitelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert

Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horňiáček, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayò Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cene-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibus-sirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Lisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. [Universal dependencies 2.5](#). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

A COMBO Implementation

COMBO is a Python package that uses the PyTorch (Paszke et al., 2019) and AllenNLP (Gardner et al., 2018) libraries. The COMBO models used in the evaluation presented in Section 3 are trained with the empirically set default parameters specified below. The training parameters can be easily configured and adjusted to the specifics of an individual model.

A.1 Network Hyperparameters

Embeddings An internal character-based word embedding is calculated with three convolutional layers with 512, 256 and 64 filters with dilation rates equal to 1, 2 and 4. All filters have the kernel size of 3. The internal word embedding has a size of 64 dimensions. All external word embeddings are reduced to 100-dimensional vectors by a single FC layer. As only words are used as input features in the system evaluation, the local feature embedding is a concatenation of the 64-dimensional internal and 100-dimensional external word embedding. The global feature vectors are computed by two biLSTM layers with 512 hidden units.

Prediction modules The tagger uses a FC network with a hidden layer of the size 64 to predict UPOS and FC networks with 128-dimensional hidden layers to predict XPOS and UFEATS.

The lemmatiser uses three convolutional layers with 256 filters and dilation rates equal to 1, 2 and 4. All filters have the kernel size of 3. The fourth convolutional layer with the number of filters equal to the number of character instances in training data is used to predict the probability of each character. The final layer filters have the kernel size of 1. The 256-dimensional embeddings of input characters are concatenated with the global feature vectors reduced to 32 dimensions with a single FC layer.

The arc prediction module uses 512-dimensional head, and dependent embeddings and the labelling module uses 128-dimensional vectors.

COMBO-vectoriser currently outputs 64-dimensional UPOS and 128-dimensional DEPREL embeddings.

Activation function FC and CNN layers use hyperbolic tangent and rectified linear unit (Nair and Hinton, 2010) activation functions, respectively.

A.2 Regularisation

Dropout technique for Variational RNNs (Gal and Ghahramani, 2016) with 0.33 rate is applied to the local feature embeddings and on top of the stacked biLSTM estimating global feature embeddings. The same dropout, for output and recurrent values, is used in the context of each biLSTM layer. The FC layers use the standard dropout (Srivastava et al., 2014) with 0.25 rate. Moreover, the biLSTM and convolutional layers use L2 regularisation with the rate of 1×10^{-6} , and the trainable embeddings use L2 with the rate of 1×10^{-5} .

A.3 Training

The cross-entropy loss is used for all parts of the system. The final loss is the weighted sum of losses with the following weights for each task:

- 0.05 for predicting UPOS and LEMMA,
- 0.2 for predicting UFEATS and (enh)HEAD,
- 0.8 for predicting (enh)DEPREL.

The whole system is optimised with ADAM (Kingma and Ba, 2015) with the learning rate of 0.002 and $\beta_1 = \beta_2 = 0.9$. The model is trained for a maximum of 400 epochs, and the learning rate is reduced twice by the factor of two when the validation score reaches a plateau.

B External Data Summary

Tables 4 and 5 list the UD dependency treebanks and BERT models used in the evaluation experiments presented in Section 3.

C Evaluation of UPOS and UDEPREL

The comparison of the universal parts of speech predicted by the tested systems in English, Korean and Polish data is shown in the charts in Figures 6, 7 and 8, respectively. The comparison of the quality of the predicted universal dependency types in English, Korean and Polish data is presented in Figures 9, 10 and 11, respectively.

Language	Language Type	UD Treebank	#Words	#Trees	Reference
English	isolating	English-EWT	254,856	16,622	Silveira et al. (2014)
Arabic	fusional	Arabic-PADT	282,384	7,664	Hajič et al. (2009)
Polish	fusional	Polish-PDB	350,036	22,152	Wróblewska (2018)
Finnish	agglutinative	Finnish-TDT	202,453	15,136	Haverinen et al. (2014)
Korean	agglutinative	Korean-Kaist	350,090	27,363	Chun et al. (2018)
Turkish	agglutinative	Turkish-IMST	57,859	5,635	Sulubacak et al. (2016)
Basque	agglutinative (fusional verb morphology)	Basque-BDT	121,443	8,993	Aranzabe et al. (2015)

Table 4: The UD treebanks used in the evaluation experiments.

Language	BERT model	Reference
Arabic	bert-base-arabertv2	Antoun et al. (2020)
Basque	bert-base-cased	Agerri et al. (2020)
English	bert-base-cased	Devlin et al. (2019)
Finnish	bert-base-finnish-cased-v1	Virtanen et al. (2019)
Korean	bert-kor-base	Kim (2020)
Polish	herbert-base-cased	Mroczkowski et al. (2021)
Turkish	bert-base-turkish-cased	Schweter (2020)

Table 5: The BERT models used in the evaluation experiments.

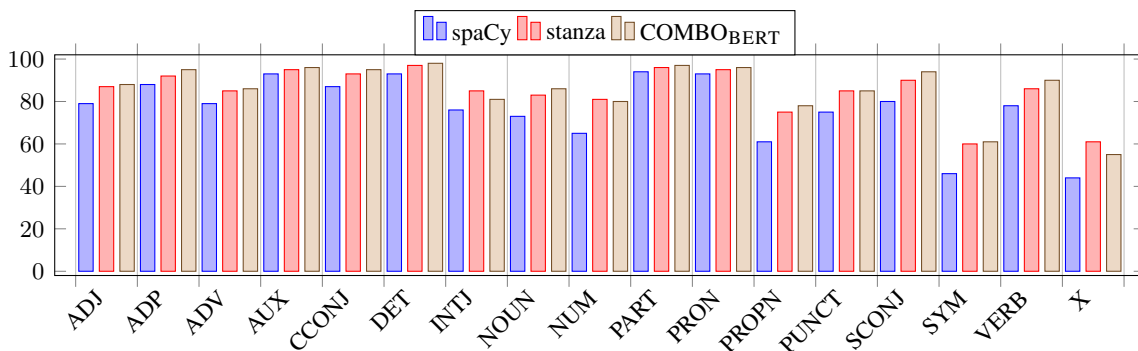


Figure 6: Evaluation of predicted universal parts of speech (UPOS) in the English test set (F_1 -scores).

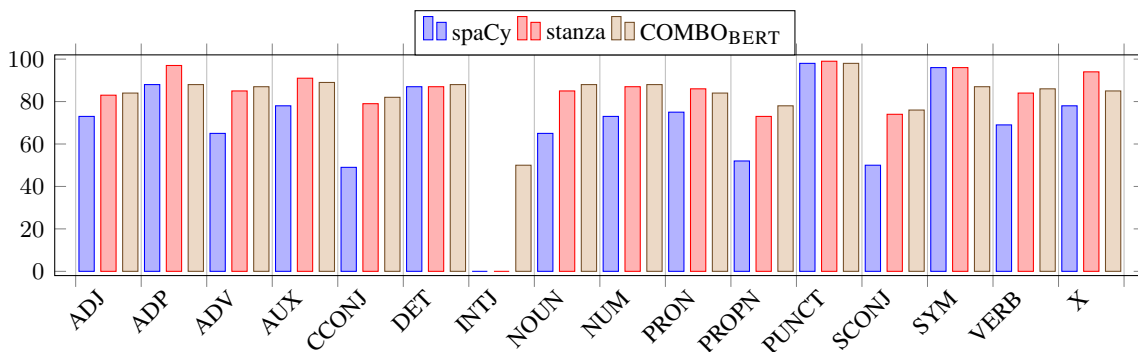


Figure 7: Evaluation of predicted universal parts of speech (UPOS) in the Korean test set (F_1 -scores).

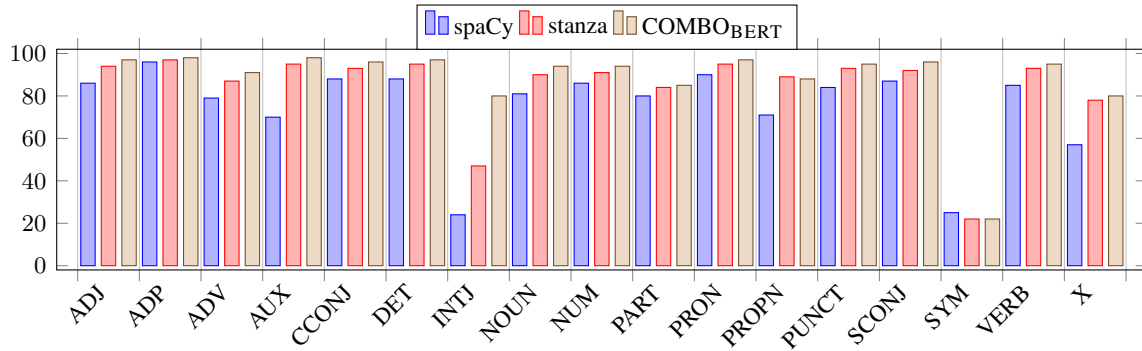


Figure 8: Evaluation of predicted universal parts of speech (UPOS) in the Polish test set (F_1 -scores).

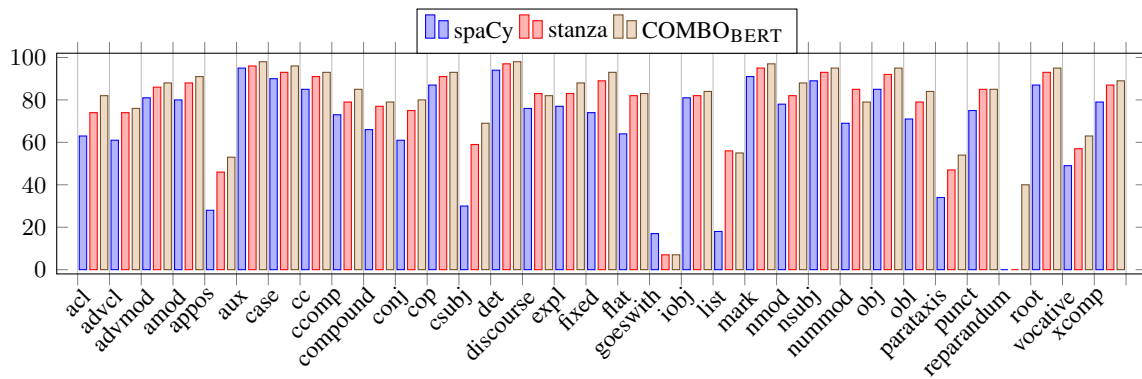


Figure 9: Evaluation of predicted grammatical functions (UDEPREL) in the English test set (F_1 -scores).

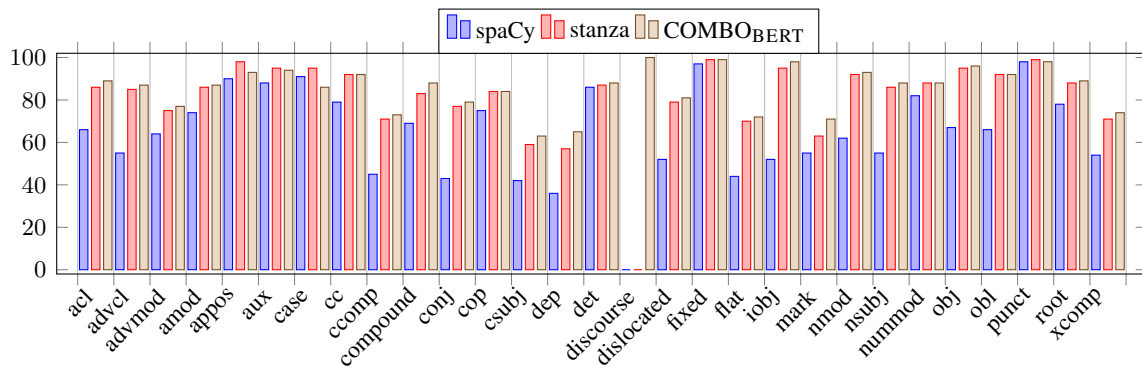


Figure 10: Evaluation of predicted grammatical functions (UDEPREL) in the Korean test set (F_1 -scores).

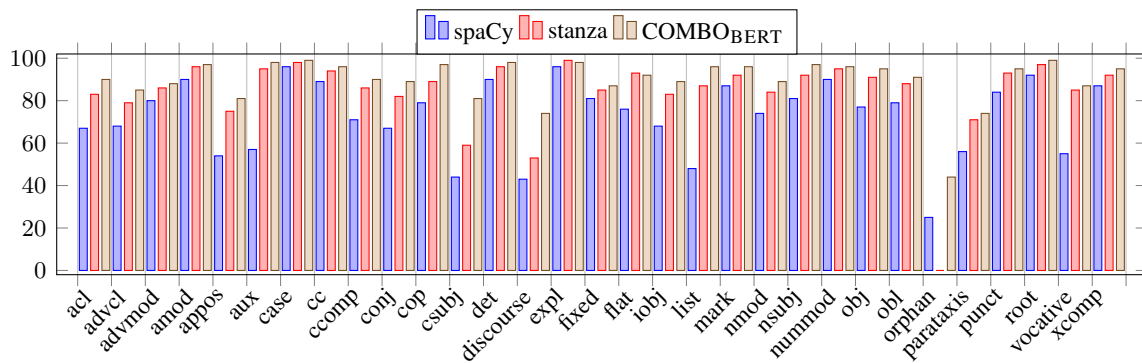


Figure 11: Evaluation of predicted grammatical functions (UDEPREL) in the Polish test set (F_1 -scores).