Turku Enhanced Parser Pipeline: From Raw Text to Enhanced Graphs in the IWPT 2020 Shared Task

Jenna Kanerva* **Filip Ginter** **Sampo Pyysalo**
TurkuNLP group, Department of Future Technologies
University of Turku, Finland
first.last@utu.fi

Abstract

We present the approach of the TurkuNLP group to the IWPT 2020 shared task on Multilingual Parsing into Enhanced Universal Dependencies. The task involves 28 treebanks in 17 different languages and requires parsers to generate graph structures extending on the basic dependency trees. Our approach combines language-specific BERT models, the UDify parser, neural sequence-to-sequence lemmatization and a graph transformation approach encoding the enhanced structure into a dependency tree. Our submission averaged 84.5% ELAS, ranking first in the shared task. We make all methods and resources developed for this study freely available under open licenses from https://turkunlp.org.

1 Introduction

The Universal Dependencies[1] (UD) effort (Nivre et al., 2016, 2020) seeks to create cross-linguistically consistent dependency annotation and has to date produced more than 150 treebanks in 90 languages. UD is a broad and open community effort with more than 300 contributors (Zeman et al., 2019), and the resources they have created have been instrumental in driving progress in dependency parsing in recent years, also serving as the basis of widely attended CoNLL shared tasks on multilingual parsing in 2017 and 2018 (Zeman et al., 2017, 2018). While UD resources, the CoNLL shared tasks, and recent advances in deep learning-based parsing technology (Dozat et al., 2017; Kanerva et al., 2018; Kondratyuk and Straka, 2019) have contributed substantially to accurate dependency parsing using a consistent syntactic representation for a wide range of human languages, these efforts have focused almost exclusively on the *basic* UD dependency trees. UD defines also an *enhanced* graph representation, which allows more detailed representation of the sentence. Common types of enhancements include null nodes for elided predicates, propagation of conjuncts for making connections between words more explicit, and augmentation of modifier labels with prepositional or case-marking information. The ability to produce enhanced UD graphs from raw text, previously explored by e.g. Schuster and Manning (2016), Nivre et al. (2018), and Schuster et al. (2018), would represent a further advance over existing tools.

The IWPT 2020 Shared Task on Multilingual Parsing into Enhanced Universal Dependencies[2] (Bouma et al., 2020) is the first shared task evaluation targeting the enhanced UD graph. The task was organized using data from 28 UD treebanks covering 17 languages, representing Baltic, Finnic, Germanic, Romance, Semitic, Slavic, and Southern Dravidian languages. We participated in the IWPT shared task with our parsing pipeline consisting of components for segmentation, part-of-speech and morphological tagging, lemmatization, dependency parsing, and enhanced dependency graph analysis. Our approach builds on custom pre-trained deep language models (Devlin et al., 2018), a deep neural network-based parser (Kondratyuk and Straka, 2019), a character-level sequence-to-sequence lemmatizer (Kanerva et al., 2020), and a custom graph transformation approach encoding an enhanced dependency graph in a labeled tree structure. The parsing pipeline is fully language agnostic, and therefore trainable with any UD treebank. Our submission to IWPT achieved an average enhanced labeled attachment score (ELAS) of 84.5%, the best performance among the 35 evaluated submissions from ten participating groups with an approximately 2% point margin to the second-best submission.

*Equal contribution by all three authors

[1] https://universaldependencies.org/

[2] https://universaldependencies.org/iwpt20/

Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task, pages 162–173

2 Shared Task Data

The shared task data invoves 28 UD treebanks for 17 languages, representing the subset of treebanks for which enhanced dependencies are available. The enhanced dependencies fall into five types: gapping, propagation of conjuncts, controlled and raised subjects, relative clause antecedents, and case information. However, not all treebanks have all of these types. While the training data is divided according to individual treebanks, test data is divided on language level through pooling of the individual treebank test sets, without any direct possibility to identify which test set sentence originates from which source treebank. We note that this is a departure from previous UD parsing shared tasks, where the treebank distinction was preserved also in the test data. The training and development data range from less than 10,000 words for Tamil to over a million for Czech. Table 1 gathers statistics of the enhanced dependencies, compared to the base parse trees. We can see that the number of unique relation types increases by an order of magnitude, yet roughly 70-80% of the enhanced dependencies are copied unmodified from the base tree, and roughly 90-95% are a base dependency with its relation type modified.

Treebank	Base	Enh	R%	UR%
Arabic-PADT	36	1074	66.1	92.9
Bulgarian-BTB	36	173	84.7	96.1
Czech-CAC	43	639	72.4	89.3
Czech-FicTree	42	295	78.7	90.5
Czech-PDT	43	759	75.6	91.8
Dutch-Alpino	35	416	83.3	95.7
Dutch-LassyS.	35	293	82.2	95.3
English-EWT	49	375	82.3	94.7
Estonian-EDT	38	560	76.1	98.3
Estonian-EWT	39	178	74.1	92.6
Finnish-TDT	45	418	74.1	91.1
French-Sequoia	46	71	93.9	95.3
Italian-ISDT	44	348	78.6	94.8
Latvian-LVTB	40	133	75.9	90.6
Lithuanian-A.	35	194	66.9	88.8
Polish-LFG	40	178	88.8	97.1
Polish-PDB	67	859	77.2	91.8
Russian-SynTag.	40	635	77.5	93.9
Slovak-SNK	41	268	81.0	94.3
Swedish-Talbank.	40	302	79.1	93.2
Tamil-TTB	28	116	69.3	97.3
Ukrainian-IU	57	351	77.5	91.6

Table 1: Statistics of base and enhanced relations from the training sections of the treebanks: *Base* is the number of unique relations in the base tree, *Enh* is the number of unique relations in the enhanced graph, *R%* is the proportion of enhanced dependencies also present in the base tree, and *UR%* is the proportion of unlabelled enhanced dependencies also present in the base tree. The letter R refers to recall.

3 System Overview

We next introduce our system and our approach to predicting enhanced dependencies.

3.1 Segmentation

For tokenization, multiword token expansion and sentence splitting we apply the Stanza toolkit by Qi et al. (2020) and its downloadable models trained on UD version 2.5 treebanks. Stanza implements a neural model that treats segmentation as a tagging problem over sequences of characters, where for a given character the model predicts whether it is the end of a token, the end of a sentence, or the end of a multiword token. Predicted multiword tokens are then expanded using a combination of a dictionary compiled from the training data and a sequence-to-sequence generation model.

3.2 Base Parser

We use the UDify dependency parser introduced by Kondratyuk and Straka (2019). UDify is a multi-task model for part-of-speech and morphological tagging, lemmatization and dependency parsing supporting fine-tuning of pre-trained BERT models on UD treebanks. UDify implements a multi-task network where a separate prediction layer for each task is added on top of the pre-trained BERT encoder. Additionally, instead of using only the top encoder layer representation in prediction, UDify adds attention vertically over the 12 layers of BERT, calculating a weighted sum of all intermediate representations of BERT layers for each token. All prediction layers as well as layer-wise attention are trained simultaneously, while also fine-tuning the pre-trained BERT weights.

In our shared task system we use UDify for part-of-speech tagging (`UPOS`), predicting morphological features (`FEATS`) as well as for dependency parsing. By contrast to the original UDify work, we train separate language-specific models rather than one model covering all languages.

3.3 Lemmatizer

For lemmatization we use the Universal Lemmatizer by Kanerva et al. (2020) trained on the shared task training data. The lemmatizer casts the task as a sequence-to-sequence rewrite problem where the input token is represented as a sequence of characters followed by a sequence of its part-of-speech

and morphological tags, and the desired lemma is then generated a character at time from the input. Following this approach, the contextual information needed for disambiguating between possible lemmas for ambiguous words is obtained directly from the predicted morphological tags, thus creating a compact context representation which generalizes well. In order to obtain predicted tags for lemmatization, we apply the lemmatizer as the final component in our pipeline.

3.4 Enhanced Representation

Since our base parser is only capable of reproducing trees, the enhanced representation needs to either be encoded into the base trees by enriching the set of dependency types, or alternatively introduced in a separate step after base parsing. In our system submission, we chose the former, but have also experimented with the latter approach. The overall approach of encoding the graph into a tree is well-known and has been applied previously, e.g. by a number of teams in the SemEval tasks on semantic dependency parsing (Oepen et al., 2014, 2015).

Our choices adhered to the following principles: (a) the LAS of the base parser must not be compromised, (b) the encoding must be language-independent and applicable to any treebank, and (c) the method must be sufficiently simple to be included in a production-grade parsing pipeline.

3.4.1 Encoding into Base Tree

In order to encode enhanced dependencies into the base tree, we focused on a just four structures, which nevertheless cover the vast majority of the edges in the enhanced representation (see Table 2 below). The four structures and their encoding are shown in Figure 1. In the encoding, the base tree structure does not change; the enhanced relations are encoded into the base tree relations, also recording whether the enhanced dependency goes from or to the head in the base tree, or from or to the head of the head in the base tree. This encoding makes the decoding process straightforward and deterministic, because there can be at most one head and at most one head of head in the parse tree. The downside of this approach is that the number of unique relation types which the parser needs to predict increases substantially. Note that this encoding applies straightforwardly to cases where a token is the head or dependent in several enhanced relations; their encoding is simply concatenated.

The main reason for the increase in the number of unique relation types is the lexicalized relations which encode the lemma of a functional word (e.g. the *case* dependent) into the enhanced relation. To address this issue in a language-independent manner, we scan the enhanced relations for occurrences of a lemma of a dependent of the head or the dependent in the enhanced relation. If one is found, it is replaced with a placeholder encoding which position the lemma occurred at. For instance $\{$*lemma-d-case*$\}$ indicates that this placeholder is to be replaced with the lemma of a case dependent of the dependent in this enhanced relation. Similarly, $\{$*lemma-h-case*$\}$ indicates that this placeholder is to be replaced with the lemma of a case dependent of the head in this enhanced relation. Such delexicalization is once again straightforward to reverse and in practice deterministic, although not so in theory, since a word can have several dependents of the same type.

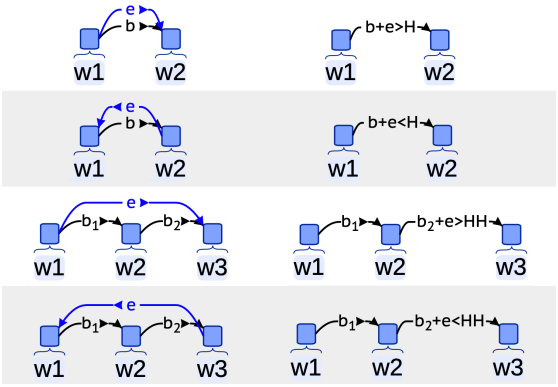

Figure 1: The four enhanced dependency structures currently captured in our encoding. The base (b) and enhanced (e) relations in the left column are encoded in a tree structure as in the right column. In the encoding, the symbol $>$ stands for "relation from", $<$ stands for "relation to", *H* is the head in the base tree, and *HH* is the head of the head in the base tree.

The final feature of the enhanced representation that we address is the empty nodes occuring in elliptic constructions. Here, we once again rely on encoding of information into the base tree. The shared task evaluation procedure includes a step whereby empty nodes are removed and encoded in the form of enhanced relations that every two relations (h, e, r_1), (e, d, r_2) produce a new enhanced relation $(h, d, r_1{>}r_2)$ which encodes the presence of an empty node. Once all relations of the empty node are encoded in this manner, the empty node is removed. This representation is easy to reverse, and in practice allows one to reconstruct the empty

nodes in the enhanced representation except for their position in the sentence, which is not particularly relevant nor evaluated in the shared task. Only cases where a word has several empty node dependents with the same relation type cannot be reconstructed correctly.

The overall procedure for encoding the enhanced representation is:

1. Encode empty nodes as enhanced relations, remove from the graph
2. Replace all recognized function word lemmas with their corresponding placeholders
3. Encode all enhanced relations of the four types using the encoding in Figure 1, discard any other enhanced relations

This sequence of steps produces a tree representation that a standard dependency parser can be trained on. The output of the parser is decoded in the reverse order of the encoding steps, producing the enhanced representation. The decoding must take into account any errors the parser produced which might impair the decoding of the encoded representation, or produce an enhanced graph which does not validate as Universal Dependencies. In particular:

- Any relation headed by the root is given the type *root* regardless of the parser's prediction.
- If a lemma placeholder cannot be reversed (e.g. when a parser predicts a placeholder {*lemma-d-case*} but there is no such dependent in the tree, the enhanced relation is discarded. Note that leads to unconnected words in the enhanced graph.
- Any word that remains unconnected in the enhanced graph is made the dependent of the same head, with the same relation, as in the base tree.
- For any (undirected) connected component that does not include the root node, we identify a word that all other words of the component can be reached from in the directed graph, and make this word a dependent of the root node. If no such word can be found, then the set of words with no incoming edge in the component are made dependents of the root node. This latter condition did not trigger in practice.

The encode-decode procedure can be evaluated by first encoding the enhanced training graphs into trees, decoding back, and measuring the ELAS of the decoded data against the original. A lossless representation would result in ELAS of 100%. As shown in Table 2, this value is in the 97.9–99.9% range across all treebanks, meaning the encoding is not far from lossless, and only little gain can be expected from encoding more complex structures. Note, however, that this reflects the comparative structural simplicity of the enhanced relations present in the UD data, rather than the generality of our encoding. Table 2 also reports on the number of unique dependency relations in the training section of each treebank, showing an order of magnitude increase compared to the base tree.

Treebank	Rels	ELAS
Arabic-PADT	1,108	99.28
Bulgarian-BTB	152	99.22
Czech-CAC	939	98.13
Czech-FicTree	355	98.38
Czech-PDT	1,079	98.75
Dutch-Alpino	569	99.16
Dutch-LassySmall	420	99.23
English-EWT	611	98.89
Estonian-EDT	359	99.88
Estonian-EWT	202	99.74
Finnish-TDT	451	97.96
French-Sequoia	79	99.09
Italian-ISDT	561	99.53
Latvian-LVTB	405	97.94
Lithuanian-ALKSNIS	267	98.12
Polish-LFG	146	99.21
Polish-PDB	845	98.34
Russian-SynTagRus	1,119	99.57
Slovak-SNK	281	99.44
Swedish-Talbanken	494	99.16
Tamil-TTB	78	99.79
Ukrainian-IU	363	98.88

Table 2: Number of unique dependency relations after the encoding procedure, and the ELAS value after an encode-decode cycle. The latter number reflects to what extent the original enhanced graphs can be reconstructed after the encoding. The numbers are reported on the training portions of the treebanks.

3.4.2 Enhanced Relations as Tagging

The encoding of the enhanced relations into the base tree can also be seen as a tagging task, since every word has exactly one base relation, and therefore also exactly one relation in the encoded tree. It is therefore possible to first parse the sentence with a parser that predicts the base tree, and then subsequently tag the words with tags corresponding to the encoding of the enhanced relations, as introduced earlier, with the base parse tree serving as a source of features. The main advantage of such an approach would be guaranteeing that the

Model	Languages	References
Arabic-BERT	Arabic	https://github.com/alisafaya/Arabic-BERT
BERTje	Dutch	https://github.com/wietsedv/bertje; (de Vries et al., 2019)
BERT (original)	English	https://github.com/google-research/bert; (Devlin et al., 2018)
FinBERT	Finnish	https://turkunlp.org/FinBERT/; (Virtanen et al., 2019)
CamemBERT	French	https://camembert-model.fr/; (Martin et al., 2020)
Italian BERT	Italian	https://github.com/dbmdz/berts
RuBERT	Russian	https://github.com/deepmipt/deeppavlov/; (Kuratov and Arkhipov, 2019)
Slavic-BERT	Slavic[1]	https://github.com/deepmipt/Slavic-BERT-NER; (Arkhipov et al., 2019)
Swedish BERT	Swedish	https://github.com/Kungbib/swedish-bert-models
mBERT	104 lang.	https://github.com/google-research/bert

Table 3: Previously released BERT models for shared task languages. [1]Slavic-BERT is trained on Bulgarian, Czech, Polish, and Russian.

base LAS of the parser does not change, while the main disadvantage is the added complexity of an additional step and the possibility of error chaining.

We pursued this alternative approach in parallel to the main line of work. As the results presented in Section 5 show, however, the encoding of the enhanced dependencies does not negatively affect the base LAS, undermining the motivation for a separate tagging approach with its added software complexity. In our preliminary experiments on the development data, the tagging approach resulted in a minimally worse performance than the primary approach, and was therefore not pursued further.

4 Language Models

We apply transfer learning using pre-trained BERT models, using multilingual BERT[3] (mBERT) as a starting point. Based on recent studies introducing language-specific BERT models (Arkhipov et al., 2019; Virtanen et al., 2019; de Vries et al., 2019; Martin et al., 2020), we anticipated that parsing performance could be substantially improved by replacing the multilingual model with dedicated language-specific ones. To identify or create a model that would improve on performance with mBERT for every treebank in the shared task, we adopted a three-stage approach: 1) use previously released models, 2) pre-train a new model on Wikipedia data, and 3) continue pre-training on texts from a web crawl.

4.1 Previously Released Models

We considered the previously released models summarized in Table 3. Based on preliminary experiments, we focused on cased models in cases where both cased and uncased variants are available. We evaluated mBERT for all shared task treebanks, Slavic-BERT for Bulgarian, Czech, Polish, and Russian, and the other models for treebanks for the individual languages that those models target.

4.2 Unannotated Texts

Our primary source of unannotated texts in various languages is Wikipedia. To extract plain text, we processed the full 2020/01/20 Wikipedia database backup dumps[4] for the various languages with WikiExtractor[5]. The basic statistics of extracted Wikipedia texts for the IWPT languages are summarized in Table 9 in the Appendix. We note that the sizes of these unnanotated texts vary greatly between languages, ranging just over 20 million tokens for Latvian to nearly 3 billion for English. In many cases, languages with large Wikipedias also have large annotated treebanks, and vice versa; the language with the smallest amount of annotated training data in the shared task, Tamil, also ranks second from bottom in terms of the available unannotated Wikipedia data. We augmented the collection of unannotated texts for selected languages with texts drawn from OSCAR[6] (Ortiz Suárez et al., 2019), using unshuffled versions provided by the creators of the corpus (see Table 8 in the Appendix). The unshuffled version of the corpus is used since BERT training is carried out on text segments of up to 512 sub-words, far longer than most individual sentences. To reduce the level of noise in the web-crawled texts, we filtered the OSCAR source using 5-gram perplexity with a KenLM[7] language model estimated on Wikipedia data. In brief, we measured the average sentence-level perplexity t and filtered out any document where the average perplexity was greater than t. In terms of tokens, this procedure

[3] https://github.com/google-research/bert/blob/master/multilingual.md

[4] https://dumps.wikimedia.org/

[5] https://github.com/attardi/wikiextractor

[6] https://traces1.inria.fr/oscar/

[7] https://github.com/kpu/kenlm

Treebank	Model mBERT	Language-specific	
Arabic PADT	**83.62**	82.76	(Arabic-BERT)
Bulgarian BTB	90.75	**91.83**	(Slavic-BERT)
Czech CAC	91.80	**92.99**	(Slavic-BERT)
Czech FicTree	92.31	**93.27**	(Slavic-BERT)
Czech PDT	92.58	**93.44**	(Slavic-BERT)
Dutch Alpino	92.58	**93.36**	(BERTje)
Dutch LassySmall	**88.30**	87.69	(BERTje)
English EWT	90.08	**91.82**	(BERT-large)
Estonian EWT	71.27	**73.08**	(WikiBERT-et)
Finnish TDT	87.83	**92.89**	(FinBERT)
French Sequoia	**93.12**	92.99	(CamemBERT)
Italian ISDT	92.75	**93.44**	(Italian BERT)
Latvian LVTB	**86.71**	85.96	(WikiBERT-lv)
Lithuanian ALKSNIS	83.02	**85.26**	(WikiBERT-lt)
Polish LFG	95.34	**96.22**	(Slavic-BERT)
Polish PDB	91.90	**93.37**	(Slavic-BERT)
Russian SynTagRus	92.06	**93.34**	(RuBERT)
Slovak SNK	**92.52**	91.89	(WikiBERT-sk)
Swedish Talbanken	86.96	**90.56**	(Swedish BERT)
Tamil TTB	**69.12**	67.38	(WikiBERT-ta)
Ukrainian IU	89.60	**91.25**	(WikiBERT-uk)
Average	88.30	**89.28**	

Table 4: UDify development set LAS performance with mBERT compared to language-specific BERTs

filtered out approx. 10% of the OSCAR data for Latvian and Slovak and 24% for Tamil.

4.3 Pre-training

For pre-training new BERT models, we largely follow the approach used to create the original BERT-base English model by Devlin et al. (2018). Specifically, we adapt the preprocessing pipeline and pre-training process introduced by Virtanen et al. (2019) for creating the Finnish BERT model. In brief, we train BERT-base models for 1M steps, the initial 900K with a maximum sequence length of 128 and the last 100K with 512, using the original BERT software[8] and the same optimizer parameters as Devlin et al. (2018) with the exception of batch size. Due to memory limitations, a batch size of 140 was used with 4 GPUs for the first 900K steps and a batch size of 20 with 8 GPUs for the last 100K steps. Nvidia V100 GPUs with 32 GB memory were used for pre-training. For comprehensive details of the preprocessing and pre-training process, we refer to the documentation of our pipeline.[9]

4.4 Language Model Evaluation

For evaluating pre-trained language models, we trained UDify with the shared task training data for each language and evaluated on the corresponding development dataset using gold standard tokenization. The standard LAS metric was used to assess model performance.

[8] https://github.com/google-research/bert

[9] https://github.com/TurkuNLP/wikibert

Treebank	Model mBERT	Language-specific	
Arabic PADT	83.62	**84.79**	(WikiBERT-ar)
Dutch Alpino	92.58	**93.47**	(WikiBERT-nl)
Dutch LassySmall	88.30	**89.23**	(WikiBERT-nl)
French Sequoia	93.12	**93.21**	(WikiBERT-fr)
Average	89.41	**90.18**	

Table 5: UDify development set LAS performance with mBERT compared to additional WikiBERTs

Treebank	Model mBERT	Language-specific	
Latvian LVTB	86.71	**88.47**	(Wiki+OSCAR-BERT-lv)
Slovak SNK	**92.52**	**92.52**	(Wiki+OSCAR-BERT-sk)
Tamil TTB	69.12	**71.02**	(Wiki+OSCAR-BERT-ta)
Average	82.78	**84.00**	

Table 6: UDify development set LAS performance with mBERT compared to Wiki+OSCAR-BERTs

Table 4 summarizes evaluation results comparing parsing performance with mBERT and language-specific models. As expected, we find that language-specific models outperform the multilingual model in most cases, averaging approximately 1% point higher LAS (∼8% reduction in error). There are nevertheless a number of cases where UDify with mBERT outperforms the language-specific model. To address these cases, we introduced additional WikiBERT models for Arabic, Dutch, and French. Results comparing the performance of these models with mBERT are summarized in Table 5. We find that in each case using the WikiBERT model improves on results with mBERT, with absolute differences around 1% point for the Arabic and Dutch treebanks but very limited (∼0.1% point) difference for French, averaging 0.8% point higher LAS than mBERT (∼7% reduction in error).

Finally, there are three languages for which no previously released language-specific model was available and the WikiBERT failed to improve on performance with mBERT: Latvian, Slovak, and Tamil. For these languages, we continued pre-training with texts from OSCAR for an additional 300,000 steps. Table 6 summarizes performance with these models. For Slovak, the new model improves over the WikiBERT model performance but merely matches the performance with mBERT, while the Latvian and Tamil models outperform

	Team									
Language	**adapt**	**clasp**	**emory**	**fastparse**	**koebsala**	**orange**	**robert**	**shanghai**	**turku**	**unipi**
Arabic	57.19	51.26	67.26	66.92	60.84	70.96	0.0	63.41	**77.82**	57.79
Bulgarian	77.29	84.90	88.19	84.86	68.88	89.42	0.0	78.67	**90.73**	84.93
Czech	66.41	67.13	85.51	77.21	61.11	86.95	0.0	75.43	**87.51**	75.99
Dutch	67.67	78.93	80.72	77.37	62.93	**85.14**	0.0	70.94	84.73	77.62
English	70.44	82.87	85.30	78.45	65.37	85.21	**88.94**	72.34	87.15	83.95
Estonian	61.12	60.44	81.36	74.09	59.07	81.03	0.0	74.91	**84.54**	57.24
Finnish	72.37	65.96	82.96	75.73	67.54	86.24	0.0	75.99	**89.49**	72.13
French	74.74	72.76	**86.23**	77.77	67.93	83.63	0.0	76.99	85.90	78.85
Italian	71.98	87.14	88.52	84.77	69.08	90.83	0.0	73.08	**91.54**	89.14
Latvian	72.41	66.01	79.19	75.57	64.75	82.11	0.0	77.77	**84.94**	68.23
Lithuanian	58.36	52.56	66.12	61.41	56.28	75.89	0.0	66.85	**77.64**	61.06
Polish	65.86	71.22	82.39	74.54	61.34	80.39	0.0	71.01	**84.64**	70.61
Russian	75.27	70.37	88.60	80.35	64.23	89.84	0.0	78.26	**90.69**	76.90
Slovak	68.43	65.16	82.72	73.46	64.08	84.36	0.0	73.14	**88.56**	81.40
Swedish	68.39	71.35	78.19	75.24	64.50	83.27	0.0	69.60	**85.64**	78.73
Tamil	48.47	42.15	54.26	46.99	47.44	**64.23**	0.0	48.20	57.83	48.50
Ukrainian	66.43	63.24	79.69	74.02	64.17	84.64	0.0	72.98	**87.22**	73.90
Average	67.23	67.85	79.84	74.04	62.91	82.60	5.23	71.74	**84.50**	72.76

Table 7: ELAS results for submissions to IWPT 2020 shared task. Team names abbreviated for space: emory = emorynlp, orange = orange_deskin, robert = robertnlp, shanghai = shanghaitech_alibaba, turku = turkunlp.

mBERT with a nearly 2% point absolute difference in LAS. On average, the new models improve on mBERT by 1.2% points, again an approx. 7% reduction in error.

5 Results

For our final submission, we trained a model for each language using the largest treebank (in terms of token count) for the language in the shared task data release. All segmentation, tagging, parsing, and lemmatization models are thus monolingual and trained using only a single treebank. Each UDify model is fine-tuned for 160 epochs using a number of warm-up steps[10] roughly equal to a single pass over the training dataset. For each language the fine-tuning is based on a custom pre-trained BERT model selected as detailed in Section 4.4. Lemmatization models do not require any external resources, and all hyperparameters follow the values used in Kanerva et al. (2020).

The primary evaluation metric in the shared task is ELAS (Labeled Attachment Score on Enhanced dependencies), which calculates F-score over the set of enhanced dependencies in the system output and gold standard.[11] Table 7 summarizes the ELAS results for all ten teams participating the shared task. We note that in addition to achieving the best average ELAS performance, our system also outperforms all other submissions for 13 out of the 17 individual languages included in the task. For these 13 languages, the largest absolute differences for the second-best result are for Arabic (∼6.9% points), Slovak (∼4.2% points), Estonian, and Finnish (both slightly above 3% points).

For the four languages where our system did not achieve the highest ELAS results, the differences to the highest-performing submission are small (0.3-0.4% points) for Dutch and French, and 1.8% points for English. However, there is a more than 6% point difference to the top result for Tamil, the language with the smallest treebank in the shared task. This difference indicates a tradeoff of our approach in training monolingual models: languages with particularly limited resources do not gain support from annotations in other languages as they would in multilingual training.

Table 10 in the Appendix shows average results for all metrics excepting for XPOS, which due time limitations we decided not to predict, and AllTags, which is not meaningfully defined when not predicting XPOS. We note that our system achieves the best performance for all but two metrics, outperforming other systems in segmentation (Tokens, Words, Sentences), part-of-speech tagging (UPOS), lemmatization (Lemmas) as well as for all but one of the seven dependency attachment score (*AS) metrics. Our system falls behind the best-performing submission (orange_deskin) for the UFeats and MLAS metrics. As MLAS (Morphology-Aware Labeled Attachment Score)

[10] During warm-up, the learning rate is gradually increased from zero to its initial value, so as to avoid large changes at the very beginning of the training.

[11] Note that in UD many of the base layer relations are repeated in the enhanced graph, and therefore the ELAS metric evaluates a combination of basic dependencies and enhancements as seen in statistics presented in Table 1.

requires selected features to match, the results for these two metrics likely both reflect performance for morphological features. The absolute difference of our system to the top result for UFeats is 1.2% points, reflecting a 20% relative increase in error and indicating a clear remaining point for improvement in our system.

6 Discussion

Cross-lingual compatibility is a major goal of the UD effort and the ability to train multilingual models where lower-resourced languages can benefit from data in higher-resourced languages a clearly desirable aim in language modeling. While our approach – which trains monolingual models and uses language-specific pre-trained models – can be seen as running counter to these goals, we do nevertheless share them. Our choice to train separate models for each language for the shared task is based in part in awareness of remaining compatibility issues in UD treebanks, even within languages. We hope contrasting results for joint and language-specific models for this shared task will help identify and resolve some of these challenges. Regarding multilingual language models, we note that in aiming to cover more than 100 languages without a corresponding increase in model and vocabulary size, mBERT faces multiple challenges in its capacity, and the model training does not fully balance lower- and higher-resourced languages. While we here found language-specific models to outperform a specific mBERT model, highly multilingual models addressing these challenges might well be competitive with language-specific ones, and the creation of such models would greatly benefit practical parsing efforts targeting a large number of languages.

To study the impact of the language-specific language models in our shared task results, we reproduce our pipeline using exactly same configurations except for replacing all language-specific BERT models with the multilingual mBERT. In this experiment, all languages are using the same multilingual language model as a starting point, later individually fine-tuned for each language while training the language-specific parsing models. When comparing these models to the official submissions of all 10 teams, the average ELAS is approximately 1.7% points below our own primary submission ($\sim$11% increase in error), but still slighty above the second best submission by approximately 0.2% points. This means that, our pipeline would have reached the highest average ELAS score among the official submissions also without the language-specific BERT models, but only with a very thin margin to the next best team.

7 Conclusions

We have presented the approach of the TurkuNLP group to the IWPT 2020 shared task on Multilingual Parsing into Enhanced Universal Dependencies. Our approach is based on deep transfer learning with language-specific models, the state-of-the-art UDify neural parsing pipeline, sequence-to-sequence lemmatization, and a graph transformation approach to predicting enhanced dependency graphs. Our submission to the shared task achieved the highest performance for the primary evaluation metric (ELAS) both on average as well as for 13 out of the 17 languages involved in the task, also achieving the highest average performance for most other evaluation metrics.

All of the methods and resources developed for this study are made freely available under open licenses from `https://turkunlp.org`.

Acknowledgments

We gratefully acknowledge the support of the Academy of Finland, and CSC — the Finnish IT Center for Science for providing computational resources. We also thank the creators of the OSCAR corpus for making unshuffled versions of their corpus available for this work.

References

Mikhail Arkhipov, Maria Trofimova, Yurii Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93.

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*, Seattle, US. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the CoNLL 2018 Shared Task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 133–142.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2020. Universal Lemmatizer: A sequence to sequence model for lemmatizing Universal Dependencies treebanks. *Natural Language Engineering*. To appear.

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.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. *arXiv preprint arXiv:1905.07213*.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte 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*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

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 4027–4036. European Language Resources Association.

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing Universal Dependency Treebanks: A Case Study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 Task 18: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 Task 8: Broad-Coverage Semantic Dependency Parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. In *7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7)*. Leibniz-Institut für Deutsche Sprache.

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 (ACL), System Demonstrations*.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378. European Language Resources Association (ELRA).

Sebastian Schuster, Joakim Nivre, and Christopher D. Manning. 2018. Sentences with Gapping: Parsing and Reconstructing Elided Predicates. 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 1156–1168, New Orleans, Louisiana. 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. *arXiv preprint arXiv:1912.07076*.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model. *arXiv preprint arXiv:1912.09582*.

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.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė 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 Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, 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 Čéplö, 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 Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž 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 Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, 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, Ọlájídé 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, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Logi-nova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, 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 Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, 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 Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Luong Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Mai Olúòkun, Adédayọand Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-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 Shohibussirri, 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, Liisi 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.

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 Cinkova, Jan Hajič jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, 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 dePaiva, 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 Mendonca, 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. Association for Computational Linguistics.

A Appendix

Table 8 shows the same statistics for the OSCAR corpora of selected languages, and Table 9 summarizes the basic statistics of extracted Wikipedia texts for the IWPT languages. Table 10 shows average results for various metrics for all submissions to IWPT 2020 shared task.

Language	Docs	Sents	Tokens	Chars
Latvian	1.6M	34M	628M	4.0B
Slovak	5.5M	99M	1.5B	9.1B
Tamil	1.3M	39M	528M	3.8B

Table 8: OSCAR source statistics for selected IWPT 2020 shared task languages

Language	Docs	Sents	Tokens	Chars
Arabic	1.0M	8.0M	184M	889M
Bulgarian	259K	4.1M	71M	397M
Czech	444K	7.9M	143M	804M
Dutch	2.0M	19M	300M	1.7B
English	5.9M	124M	2.7B	14B
Estonian	205K	2.7M	38M	252M
Finnish	477K	7.4M	97M	731M
French	2.2M	34M	858M	4.5B
Italian	1.6M	22M	579M	3.0B
Latvian	99K	1.3M	21M	126M
Lithuanian	196K	2.3M	34M	207M
Polish	1.4M	16M	282M	1.7B
Russian	1.6M	31M	565M	3.5B
Slovak	232K	2.8M	39M	229M
Swedish	3.7M	30M	364M	2.1B
Tamil	132K	1.9M	26M	195M
Ukrainian	979K	15M	260M	1.5B

Table 9: Wikipedia source statistics for IWPT 2020 shared task languages

	Team									
Metric	**adapt**	**clasp**	**emory**	**fastparse**	**koebsala**	**orange**	**robert**	**shanghai**	**turku**	**unipi**
Tokens	99.54	99.72	99.66	99.66	99.66	99.68	5.85	99.67	**99.74**	99.63
Words	98.96	99.12	99.06	99.06	99.06	99.09	5.85	99.08	**99.13**	99.03
Sentences	89.22	92.34	91.25	91.18	91.25	90.24	5.07	91.97	**92.41**	90.56
UPOS	95.88	95.48	93.63	93.60	93.63	96.69	5.63	0.63	**96.75**	92.78
UFeats	91.36	90.66	87.35	88.11	87.35	**93.98**	5.57	32.84	92.77	86.02
Lemmas	95.40	95.15	92.30	92.23	92.30	95.80	5.62	0.02	**95.96**	91.35
UAS	87.18	86.41	88.95	82.55	79.97	89.45	5.26	13.01	**89.92**	84.90
LAS	84.09	82.66	86.14	77.57	75.41	86.79	5.11	0.99	**87.31**	80.74
CLAS	81.56	79.66	83.81	72.97	71.18	84.42	5.00	1.22	**85.23**	77.42
MLAS	72.57	69.55	67.84	60.82	60.54	**77.75**	4.51	0.01	76.63	62.73
BLEX	78.11	76.00	76.11	66.70	65.38	80.86	4.73	0.00	**81.93**	70.03
EULAS	69.42	80.18	81.26	75.96	64.93	84.62	5.26	73.01	**85.83**	78.82
ELAS	67.23	67.85	79.84	74.04	62.91	82.60	5.23	71.74	**84.50**	72.76

Table 10: Average results for different metrics for submissions to IWPT 2020 shared task. Team names abbreviated for space: emory = emorynlp, orange = orange_deskin, robert = robertnlp, shanghai = shanghaitech_alibaba, turku = turkunlp.