82 Treebanks, 34 Models: Universal Dependency Parsing with Multi-Treebank Models

Aaron Smith[*]   **Bernd Bohnet**[†]   **Miryam de Lhoneux**[*]

Joakim Nivre[*]   **Yan Shao**[*]   **Sara Stymne**[*]

[*]Department of Linguistics and Philology
Uppsala University
Uppsala, Sweden

[†]Google Research
London, UK

Abstract

We present the Uppsala system for the CoNLL 2018 Shared Task on universal dependency parsing. Our system is a pipeline consisting of three components: the first performs joint word and sentence segmentation; the second predicts part-of-speech tags and morphological features; the third predicts dependency trees from words and tags. Instead of training a single parsing model for each treebank, we trained models with multiple treebanks for one language or closely related languages, greatly reducing the number of models. On the official test run, we ranked 7th of 27 teams for the LAS and MLAS metrics.

Our system obtained the best scores overall for word segmentation, universal POS tagging, and morphological features.

1 Introduction

The CoNLL 2018 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018) requires participants to build systems that take as input raw text, without any linguistic annotation, and output full labelled dependency trees for 82 test treebanks covering 46 different languages. Besides the labeled attachment score (LAS) used to evaluate systems in the 2017 edition of the Shared Task (Zeman et al., 2017), this year's task introduces two new metrics: morphology-aware labeled attachment score (MLAS) and bi-lexical dependency score (BLEX). The Uppsala system focuses exclusively on LAS and MLAS, and consists of a three-step pipeline. The first step is a model for joint sentence and word segmentation which uses the BiRNN-CRF framework of Shao et al. (2017, 2018) to predict sentence and word boundaries in the raw input

and simultaneously marks multiword tokens that need non-segmental analysis. The second component is a part-of-speech (POS) tagger based on Bohnet et al. (2018), which employs a sentence-based character model and also predicts morphological features. The final stage is a greedy transition-based dependency parser that takes segmented words and their predicted POS tags as input and produces full dependency trees. While the segmenter and tagger models are trained on a single treebank, the parser uses multi-treebank learning to boost performance and reduce the number of models.

After evaluation on the official test sets (Nivre et al., 2018), which was run on the TIRA server (Potthast et al., 2014), the Uppsala system ranked 7th of 27 systems with respect to LAS, with a macro-average F1 of 72.37, and 7th of 27 systems with respect to MLAS, with a macro-average F1 of 59.20. It also reached the highest average score for word segmentation (98.18), universal POS (UPOS) tagging (90.91), and morphological features (87.59).

Corrigendum: After the test phase was over, we discovered that we had used a non-permitted resource when developing the UPOS tagger for Thai PUD (see Section 4). Setting our LAS, MLAS and UPOS scores to 0.00 for Thai PUD gives the corrected scores: LAS 72.31, MLAS 59.17, UPOS 90.50. This does not affect the ranking for any of the three scores, as confirmed by the shared task organizers.

2 Resources

All three components of our system were trained principally on the training sets of Universal Dependencies v2.2 released to coincide with the shared task (Nivre et al., 2018). The tagger and parser also make use of the pre-trained word

embeddings provided by the organisers, as well as Facebook word embeddings (Bojanowski et al., 2017), and both word and character embeddings trained on Wikipedia text[1] with word2vec (Mikolov et al., 2013). For languages with no training data, we also used external resources in the form of Wikipedia text, parallel data from OPUS (Tiedemann, 2012), the Moses statistical machine translation system (Koehn et al., 2007), and the Apertium morphological transducer for Breton.[2]

3 Sentence and Word Segmentation

We employ the model of Shao et al. (2018) for joint sentence segmentation and word segmentation. Given the input character sequence, we model the prediction of word boundary tags as a sequence labelling problem using a BiRNN-CRF framework (Huang et al., 2015; Shao et al., 2017). This is complemented with an attention-based LSTM model (Bahdanau et al., 2014) for transducing non-segmental multiword tokens. To enable joint sentence segmentation, we add extra boundary tags as in de Lhoneux et al. (2017a).

We use the default parameter settings introduced by Shao et al. (2018) and train a segmentation model for all treebanks with at least 50 sentences of training data. For treebanks with less or no training data (except Thai discussed below), we substitute a model for another treebank/language:

- For Japanese Modern, Czech PUD, English PUD and Swedish PUD, we use the model trained on the largest treebank from the same language (Japanese GSD, Czech PDT, English EWT and Swedish Talbanken).

- For Finnish PUD, we use Finnish TDT rather than the slightly larger Finnish FTB, because the latter does not contain raw text suitable for training a segmenter.

- For Naija NSC, we use English EWT.

- For other test sets with little or no training data, we select models based on the size of the intersection of the character sets measured on Wikipedia data (see Table 2 for details).[3]

Thai Segmentation of Thai was a particularly difficult case: Thai uses a unique script, with no spaces between words, and there was no training data available. Spaces in Thai text can function as sentence boundaries, but are also used equivalently to commas in English . For Thai sentence segmentation, we exploited the fact that four other datasets are parallel, i.e., there is a one-to-one correspondence between sentences in Thai and in Czech PUD, English PUD, Finnish PUD and Swedish PUD.[4] First, we split the Thai text by white space and treat the obtained character strings as potential sentences or sub-sentences. We then align them to the segmented sentences of the four parallel datasets using the Gale-Church algorithm (Gale and Church, 1993). Finally, we compare the sentence boundaries obtained from different parallel datasets and adopt the ones that are shared within at least three parallel datasets.

For word segmentation, we use a trie-based segmenter with a word list derived from the Facebook word embeddings.[5] The segmenter retrieves words by greedy forward maximum matching (Wong and Chan, 1996). This method requires no training but gave us the highest word segmentation score of 69.93% for Thai, compared to the baseline score of 8.56%.

4 Tagging and Morphological Analysis

We use two separate instantiations of the tagger[6] described in Bohnet et al. (2018) to predict UPOS tags and morphological features, respectively. The tagger uses a Meta-BiLSTM over the output of a sentence-based character model and a word model. There are two features that mainly distinguishes the tagger from previous work. The character BiLSTMs use the full context of the sentence in contrast to most other taggers which use words only as context for the character model. This character model is combined with the word model in the Meta-BiLSTM relatively late, after two layers of BiLSTMs.

For both the word and character models, we use two layers of BiLSTMs with 300 LSTM cells per layer. We employ batches with 8000 words and 20000 characters. We keep all other hyperparameters as defined in Bohnet et al. (2018). From the training schema described in the above

---

[1] https://dumps.wikimedia.org/backup-index-by
[2] https://github.com/apertium/apertium-bre
[3] North Sami Giella was included in this group by mistake, as we underestimated the size of the treebank.

[4] This information was available in the README files distributed with the training data and available to all participants.
[5] github.com/facebookresearch/fastText
[6] https://github.com/google/meta_tagger

paper, we deviate slightly in that we perform early stopping on the word, character and meta-model independently. We apply early stopping due to the performance of the development set (or training set when no development set is available) and stop when no improvement occurs in 1000 training steps. We use the same settings for UPOS tagging and morphological features.

To deal with languages that have little or no training data, we adopt three different strategies:

- For the PUD treebanks (except Thai), Japanese Modern and Naija NSC, we use the same model substitutions as for segmentation (see Table 2).

- For Faroese we used the model for Norwegian Nynorsk, as we believe this to be the most closely related language.

- For treebanks with small training sets we use only the provided training sets for training. Since these treebanks do not have development sets, we use the training sets for early stopping as well.

- For Breton and Thai, which have no training sets and no suitable substitution models, we use a bootstrapping approach to train taggers as described below.

Bootstrapping We first annotate an unlabeled corpus using an external morphological analyzer. We then create a (fuzzy and context-independent) mapping from the morphological analysis to universal POS tags and features, which allows us to relabel the annotated corpus and train taggers using the same settings as for other languages . For Breton, we annotated about 60,000 sentences from Breton OfisPublik, which is part of OPUS,[7] using the Apertium morphological analyzer. The Apertium tags could be mapped to universal POS tags and a few morphological features like person, number and gender. For Thai, we annotated about 33,000 sentences from Wikipedia using PyThaiNLP[8] and mapped only to UPOS tags (no features). Unfortunately, we realized only after the test phase that PyThaiNLP was not a permitted resource , which invalidates our UPOS tagging scores for Thai, as well as the LAS and MLAS scores which depend on the tagger. Note, however, that the score for morphological features

is not affected, as we did not predict features at all for Thai. The same goes for sentence and word segmentation, which do not depend on the tagger.

Lemmas Due to time constraints we chose not to focus on the BLEX metric in this shared task. In order to avoid zero scores, however, we simply copied a lowercased version of the raw token into the lemma column.

5 Dependency Parsing

We use a greedy transition-based parser (Nivre, 2008) based on the framework of Kiperwasser and Goldberg (2016b) where BiLSTMs (Hochreiter and Schmidhuber, 1997; Graves, 2008) learn representations of tokens in context, and are trained together with a multi-layer perceptron that predicts transitions and arc labels based on a few BiLSTM vectors. allow the construction of non-projective dependency trees (Nivre, 2009). We also introduce a static-dynamic oracle to allow the parser to learn from non-optimal configurations at training time in order to recover better from mistakes at test time (de Lhoneux et al., 2017b).

In our parser, the vector representation x_i of a word type w_i before it is passed to the BiLSTM feature extractors is given by:

$$x_i = e(w_i) \circ e(p_i) \circ \text{BiLSTM}(ch_{1:m}).$$

Here, $e(w_i)$ represents the word embedding and $e(p_i)$ the POS tag embedding (Chen and Manning, 2014); these are concatenated to a character-based vector, obtained by running a BiLSTM over the characters $ch_{1:m}$ of w_i.

With the aim of training multi-treebank models, we additionally created a variant of the parser which adds a treebank embedding $e(tb_i)$ to input vectors in a spirit similar to the language embeddings of Ammar et al. (2016) and de Lhoneux et al. (2017a):

$$x_i = e(w_i) \circ e(p_i) \circ \text{BiLSTM}(ch_{1:m}) \circ e(tb_i).$$

an effective way to combine multiple monolingual heterogeneous treebanks (Stymne et al., 2018) and applied them to low-resource languages (de Lhoneux et al., 2017a). In this shared task, the treebank embedding model was used both monolingually, to combine several treebanks for a single language, and multilingually, mainly for closely related languages, both for languages with no or small treebanks, and for languages with medium and large treebanks, as described in Section 6.

During training, a word embedding for each word type in the training data is initialized using the pre-trained embeddings provided by the organizers where available. For the remaining languages, we use different strategies:

- For Afrikaans, Armenian, Buryat, Gothic, Kurmanji, North Sami, Serbian and Upper Sorbian, we carry out our own pre-training on the Wikipedia dumps of these languages, tokenising them with the baseline UDPipe models and running the implementation of word2vec in the Gensim Python library[9] with 30 iterations and a minimum count of 1.

- For Breton and Thai, we use specially-trained multilingual embeddings (see Section 6).

- For Naija and Old French, we substitute English and French embeddings, respectively.

- For Faroese, we do not use pre-trained embeddings. While it is possible to train such embeddings on Wikipedia data, as there is no UD training data for Faroese we choose instead to rely on its similarity to other Scandinavian languages (see Section 6).

Word types in the training data that are not found amongst the pre-trained embeddings are initialized randomly using Glorot initialization (Glorot and Bengio, 2010), as are all POS tag and treebank embeddings. Character vectors are also initialized randomly, except for Chinese, Japanese and Korean, in which case we pre-train character vectors using word2vec on the Wikipedia dumps of these languages. At test time, we first look for out-of-vocabulary (OOV) words and characters (i.e., those that are not found in the treebank training data) amongst the pre-trained embeddings and otherwise assign them a trained OOV vector.[10] A variant of word dropout is applied to the word embeddings, as described in Kiperwasser and Goldberg (2016a), and we apply dropout also to the character vectors.

plus first item on the buffer with its leftmost dependent). We train all models for 30 epochs with hyper-parameter settings shown in Table 1. Note our unusually large character embedding sizes; we have previously found these to be effective, especially for morphologically rich lan-

Character embedding dimension	500
Character BiLSTM layers	1
Character BiLSTM output dimension	200
Word embedding dimension	100
POS embedding dimension	20
Treebank embedding dimension	12
Word BiLSTM layers	2
Word BiLSTM hidden/output dimension	250
Hidden units in MLP	100
Word dropout	0.33
α (for OOV vector training)	0.25
Character dropout	0.33
p_{agg} (for exploration training)	0.1

Table 1: Hyper-parameter values for parsing.

guages (Smith et al., 2018). Our code is publicly available. We release the version used here as UU-Parser 2.3.[11]

Using Morphological Features Having a strong morphological analyzer, we were interested in finding out whether or not we can improve parsing accuracy using predicted morphological information. We conducted several experiments on the development sets for a subset of treebanks. However, no experiment gave us any improvement in terms of LAS and we decided not to use this technique for the shared task.

What we tried was to create an embedding representing either the full set of morphological features or a subset of potentially useful features, for example case (which has been shown to be useful for parsing by Kapociute-Dzikiene et al. (2013) and Eryigit et al. (2008)), verb form and a few others. That embedding was concatenated to the word embedding at the input of the BiLSTM. We varied the embedding size (10, 20, 30, 40), tried different subsets of morphological features, and tried with and without using dropout on that embedding. We also tried creating an embedding of a concatenation of the universal POS tag and the Case feature and replace the POS embedding with this one. We are currently unsure why none of these experiments were successful and plan to investigate this in the future. It would be interesting to find out whether or not this information is captured somewhere else. A way to test this would be to use diagnostic classifiers on vector representations, as is done for example in Hupkes et al. (2018) or in Adi et al. (2017).

---

[9] https://radimrehurek.com/gensim/

[10] An alternative strategy is to have the parser store embeddings for all words that appear in either the training data or pre-trained embeddings, but this uses far more memory.

[11] https://github.com/UppsalaNLP/uuparser

6 Multi-Treebank Models

One of our main goals was to leverage information across treebanks to improve performance and reduce the number of parsing models. We use two different types of models:

1. Single models, where we train one model per treebank (17 models applied to 18 treebanks, including special models for Breton KEB and Thai PUD).

2. Multi-treebank models

   - Monolingual models, based on multiple treebanks for one language (4 models, trained on 10 treebanks, applied to 11 treebanks).
   - Multilingual models, based on treebanks from several (mostly) closely related languages (12 models, trained on 48 treebanks, applied to 52 treebanks; plus a special model for Naija NSC).

When a multi-treebank model is applied to a test set from a treebank with training data, we naturally use the treebank embedding of that treebank also for the test sentences. However, when parsing a test set with no corresponding training data, we have to use one of the other treebank embeddings. In the following, we refer to the treebank selected for this purpose as the *proxy* treebank (or simply *proxy*).

In order to keep the training times and language balance in each model reasonable, we cap the number of sentences used from each treebank to 15,000, with a new random sample selected at each epoch. This only affects a small number of treebanks, since most training sets are smaller than 15,000 sentences. For all our multi-treebank models, we apply the treebank embeddings described in Section 5. Where two or more treebanks in a multilingual model come from the same language, we use separate treebank embeddings for each of them. We have previously shown that multi-treebank models can boost LAS in many cases, especially for small treebanks, when applied monolingually (Stymne et al., 2018), and applied it to low-resource languages (de Lhoneux et al., 2017a). In this paper, we add POS tags and pre-trained embeddings to that framework, and extend it to also cover multilingual parsing for languages with varying amounts of training data.

Treebanks sharing a single model are grouped together in Table 2. To decide which languages to combine in our multilingual models, we use two sources: knowledge about language families and language relatedness, and clusterings of treebank embeddings from training our parser with all available languages. We created clusterings by training single parser models with treebank embeddings for all treebanks with training data, capping the maximum number of sentences per treebank to 800. We then used Ward's method to perform a hierarchical cluster analysis.

We found that the most stable clusters were for closely related languages. There was also a tendency for treebanks containing old languages (i.e., Ancient Greek, Gothic, Latin and Old Church Slavonic) to cluster together. One reason for these languages parsing well together could be that several of the 7 treebanks come from the same annotation projects, four from PROIEL, and two from Perseus, containing consistently annotated and at least partially parallel data, e.g., from the Bible.

For the multi-treebank models, we performed preliminary experiments on development data investigating the effect of different groupings of languages. The main tendency we found was that it was better to use smaller groups of closely related languages rather than larger groups of slightly less related languages. For example, using multilingual models only for Galician-Portuguese and Spanish-Catalan was better than combining all Romance languages in a larger model, and combining Dutch-German-Afrikaans was better than also including English.

A case where we use less related languages is for languages with very little training data (31 sentences or less), believing that it may be beneficial in this special case. We implemented this for Buryat, Uyghur and Kazakh, which are trained with Turkish, and Kurmanji, which is trained with Persian, even though these languages are not so closely related. For Armenian, which has only 50 training sentences, we could not find a close enough language, and instead train a single model on the available data. For the four languages that are not in a multilingual cluster but have more than one available treebank, we use monolingual multi-treebank models (English, French, Italian and Korean).

For the nine treebanks that have no training data we use different strategies:

- For Japanese Modern, we apply the mono-treebank Japanese GSD model.

- For the four PUD treebanks, we apply the multi-treebank models trained using the other treebanks from that language, with the largest available treebank as proxy (except for Finnish, where we prefer Finnish TDT over FTB; cf. Section 3 and Stymne et al. (2018)).

- For Faroese, we apply the model for the Scandinavian languages, which are closely related, with Norwegian Nynorsk as proxy (cf. Section 4). In addition, we map the Faroese characters {Íýúð}, which do not occur in the other Scandinavian languages, to {Iyud}.

- For Naija, an English-based creole, whose treebank according to the README file contains spoken language data, we train a special multilingual model on English EWT and the three small spoken treebanks for French, Norwegian, and Slovenian, and usd English EWT as proxy.[12]

- For Thai and Breton, we create multilingual models trained with word and POS embeddings only (i.e., no character models or treebank embeddings) on Chinese and Irish, respectively. These models make use of multilingual word embeddings provided with Facebook's MUSE multilingual embeddings,[13] as described in more detail below.

For all multi-treebank models, we choose the model from the epoch that has the best mean LAS score among the treebanks that have available development data. This means that treebanks without development data rely on a model that is good for other languages in the group. In the cases of the mono-treebank Armenian and Irish models, where there is no development data, we choose the model from the final training epoch. This also applies to the Breton model trained on Irish data.

Thai–Chinese For the Thai model trained on Chinese, we were able to map Facebook's monolingual embeddings for each language to English using MUSE, thus creating multilingual Thai-Chinese embeddings. We then trained a monolingual parser model using the mapped Chinese embeddings to initialize all word embeddings, and

ensuring that these were not updated during training (unlike in the standard parser setup described in Section 5). At test time, we look up all OOV word types, which are the great majority, in the mapped Thai embeddings first, otherwise assign them to a learned OOV vector. Note that in this case, we had to increase the word embedding dimension in our parser to 300 to accomodate the larger Facebook embeddings.

Breton–Irish For Breton and Irish, the Facebook software does not come with the necessary resources to map these languages into English. Here we instead created a small dictionary by using all available parallel data from OPUS (Ubuntu, KDE and Gnome, a total of 350K text snippets), and training a statistical machine translation model using Moses (Koehn et al., 2007). From the lexical word-to-word correspondences created, we kept all cases where the translation probabilities in both directions were at least 0.4 and the words were not identical (in order to exclude a lot of English noise in the data), resulting in a word list of 6027 words. We then trained monolingual embeddings for Breton using word2vec on Wikipedia data, and mapped them directly to Irish using MUSE. A parser model was then trained, similarly to the Thai-Chinese case, using Irish embeddings as initialization, turning off updates to the word embeddings, and applying the mapped Breton embeddings at test time.

7 Results and Discussion

Table 2 shows selected test results for the Uppsala system, including the two main metrics LAS and MLAS (plus a mono-treebank baseline for LAS),[14] the sentence and word segmentation accuracy, and the accuracy of UPOS tagging and morphological features (UFEATS). To make the table more readable, we have added a simple color coding. Scores that are significantly higher/lower than the mean score of the 21 systems that successfully parsed all test sets are marked with two shades of green/red. The lighter shade marks differences that are outside the interval defined by the standard error of the mean ($\mu \pm \text{SE}, \text{SE} = \sigma/\sqrt{N}$) but within one standard deviation (std dev) from the mean. The darker shade marks differences that are more than one std dev above/below the mean

---

[12]We had found this combination to be useful in preliminary experiments where we tried to parse French Spoken without any French training data.

[13]https://github.com/facebookresearch/MUSE

[14]Since our system does not predict lemmas, the third main metric BLEX is not very meaningful.

LANGUAGE	TREEBANK	LAS		MLAS	SENTS	WORDS	UPOS	UFEATS	SEGMENTATION	TAGGING	PARSING
ARABIC	PADT	73.54	73.54	61.04	68.06	96.19	90.70	88.25			
ARMENIAN	ARMTDP	23.90	23.90	6.97	57.44	93.20	75.39	54.45			
BASQUE	BDT	78.12	78.12	67.67	100.00	100.00	96.05	92.50			
BULGARIAN	BTB	88.69	88.69	81.20	95.58	99.92	98.85	97.51			
BRETON	KEB	33.62	33.62	13.91	91.43	90.97	85.01	70.26	FRENCH GSD		SPECIAL*
CHINESE	GSD	69.17	69.17	59.53	99.10	93.52	89.15	92.35			
GREEK	GDT	86.39	86.39	72.29	91.92	99.69	97.26	93.65			
HEBREW	HTB	67.72	67.72	44.19	100.00	90.98	80.26	79.49			
HUNGARIAN	SZEGED	73.97	73.97	56.22	94.57	99.78	94.60	86.87			
INDONESIAN	GSD	78.15	78.15	67.90	93.47	99.99	93.70	95.83			
IRISH	IDT	68.14	68.14	41.72	94.90	99.60	91.55	81.78			
JAPANESE	GSD	79.97	79.97	65.47	94.92	93.32	91.73	91.66			
JAPANESE	MODERN	28.27	28.27	11.82	0.00	72.76	54.60	71.06	JAPANESE GSD		
LATVIAN	LVTB	76.97	76.97	63.90	96.97	99.67	94.95	91.73			
OLD FRENCH	SRCMF	78.71	78.71	69.82	59.15	100.00	95.48	97.26			
ROMANIAN	RRT	84.33	84.33	76.00	95.81	99.74	97.46	97.25			
THAI	PUD	4.86	4.86	2.22	11.69	69.93	33.75	65.72			SPECIAL*
VIETNAMESE	VTB	46.15	46.15	40.03	88.69	86.71	78.89	86.43			
AFRIKAANS	AFRIBOOMS	79.47	78.89	66.35	99.65	99.37	96.28	95.39			
DUTCH	ALPINO	83.58	81.73	71.11	89.04	99.62	95.78	95.89			
	LASSYSMALL	82.25	79.59	70.88	73.62	99.87	96.18	95.85			
GERMAN	GSD	75.48	75.15	53.67	79.36	99.37	94.02	88.13			
ANCIENT GREEK	PERSEUS	65.17	62.95	44.31	98.93	99.97	92.40	90.12			
	PROIEL	72.24	71.58	54.98	51.17	99.99	97.05	91.04			
GOTHIC	PROIEL	63.40	60.58	49.79	31.97	100.00	93.43	88.60			
LATIN	ITTB	83.00	82.55	75.38	94.54	99.99	98.34	96.78			
	PERSEUS	58.32	49.86	37.57	98.41	100.00	88.73	78.86			
	PROIEL	64.10	63.85	51.45	37.64	100.00	96.21	91.46			
OLD CHURCH SLAVONIC	PROIEL	70.44	70.31	58.31	44.56	99.99	95.76	88.91			
BURYAT	BDT	17.96	8.45	1.26	93.18	99.04	50.83	40.63	RUSSIAN SYNTAGRUS		
KAZAKH	KTB	31.93	23.85	8.62	94.21	97.40	61.72	48.45	RUSSIAN SYNTAGRUS		
TURKISH	IMST	61.34	61.77	51.23	96.63	97.80	93.72	90.42			
UYGHUR	UDT	62.94	62.38	42.54	83.47	99.69	89.19	87.00			
CATALAN	ANCORA	88.94	88.68	81.39	99.35	99.79	98.38	97.90			
SPANISH	ANCORA	88.79	88.65	81.75	97.97	99.92	98.69	98.23			
CROATIAN	SET	84.62	84.13	70.53	96.97	99.93	97.93	91.70			
SERBIAN	SET	86.99	85.14	75.54	93.07	99.94	97.61	93.70			
SLOVENIAN	SSJ	87.18	87.28	77.81	93.23	99.62	97.99	94.73			
	SST	56.06	53.27	41.22	23.98	100.00	93.18	84.75			
CZECH	CAC	89.49	88.94	82.25	100.00	99.94	99.17	95.84			
	FICTREE	89.76	87.78	80.63	98.72	99.85	98.42	95.52			
	PDT	88.15	88.09	82.39	92.29	99.96	99.07	96.89			
	PUD	84.36	83.35	74.46	96.29	99.62	97.02	93.66	CZECH PDT		
POLISH	LFG	93.14	92.85	84.09	99.74	99.91	98.57	94.68			
	SZ	89.80	88.48	77.28	98.91	99.94	97.95	91.82			
SLOVAK	SNK	86.34	83.80	71.15	88.11	99.98	96.57	89.51			
UPPER SORBIAN	UFAL	28.85	2.70	3.43	73.40	95.15	58.91	42.10	SPANISH ANCORA		
DANISH	DDT	80.08	79.68	71.19	90.10	99.85	97.14	97.03			
FAROESE	OFT	41.69	39.94	0.70	95.32	99.25	65.54	34.56	DANISH DDT		NORWEGIAN NYNORSK
NORWEGIAN	BOKMAAL	88.30	87.68	81.68	95.13	99.94	98.04	97.18			
	NYNORSK	87.40	86.23	79.42	92.09	99.94	97.57	96.88			
	NYNORSKLIA	59.66	55.51	45.51	99.86	99.99	90.02	89.62			
SWEDISH	LINES	80.53	78.33	65.38	85.17	99.99	96.64	89.54			
	PUD	78.15	75.52	49.73	91.57	98.78	93.12	78.53	SWEDISH TALBANKEN		
	TALBANKEN	84.26	83.29	76.74	96.45	99.96	97.45	96.82			
ENGLISH	EWT	81.47	81.18	72.98	75.41	99.10	95.28	96.02			
	GUM	81.28	79.23	69.62	81.16	99.71	94.67	95.80			
	LINES	78.64	76.28	70.18	88.18	99.96	96.47	96.52			
	PUD	84.09	83.67	72.49	97.02	99.69	95.23	95.16	ENGLISH EWT		
ESTONIAN	EDT	81.09	81.47	74.11	92.16	99.96	97.16	95.80			
FINNISH	FTB	84.19	83.12	76.40	87.91	99.98	96.30	96.73			
	PUD	86.48	86.48	80.52	92.95	99.69	97.59	96.84	FINNISH TDT		
	TDT	84.33	84.24	77.50	91.12	99.78	97.06	95.58			
NORTH SAAMI	GIELLA	64.85	64.14	51.67	98.27	99.32	90.44	85.03	GERMAN GSD		
FRENCH	GSD	85.61	85.16	76.79	95.40	99.30	96.86	96.26			
	SEQUOIA	87.39	86.26	79.97	87.33	99.44	97.92	97.47			
	SPOKEN	71.26	69.44	60.12	23.54	100.00	95.51	100.00			
GALICIAN	CTG	78.41	78.27	65.52	96.46	98.01	95.80	97.78			
	TREEGAL	72.67	70.16	58.22	82.97	97.90	93.25	92.15			
PORTUGUESE	BOSQUE	84.41	84.27	71.76	90.89	99.00	95.90	95.41			
HINDI	HDTB	89.37	89.23	74.62	99.02	100.00	97.44	93.55			
URDU	UDTB	80.40	79.85	52.15	98.60	100.00	93.66	80.78			
ITALIAN	ISDT	89.43	89.37	81.17	99.38	99.55	97.79	97.36			
	POSTWITA	76.75	76.46	66.46	54.00	99.04	95.61	95.63			
KOREAN	GSD	81.92	81.12	77.25	92.78	99.87	95.61	99.63			
	KAIST	84.98	84.74	78.90	100.00	100.00	95.21	100.00			
KURMANJI	MG	29.54	7.61	5.77	90.85	96.97	61.33	48.26	SPANISH ANCORA		
PERSIAN	SERAJI	83.39	83.22	76.97	99.50	99.60	96.79	97.02			
NAIJA	NSC	20.44	19.44	3.55	0.00	98.53	57.19	36.09	ENGLISH EWT		SPECIAL*
RUSSIAN	SYNTAGRUS	89.00	89.39	81.01	98.79	99.61	98.59	94.89			
	TAIGA	65.49	59.32	46.07	66.40	97.81	89.32	82.15			
UKRAINIAN	IU	82.70	81.41	59.15	93.42	99.76	96.89	81.95			
ALL	OFFICIAL	72.37	70.71	59.20	83.80	98.18	90.91	87.59			
ALL	CORRECTED	72.31	70.65	59.17	83.80	98.18	90.50	87.59			
BIG		80.25	79.61	68.81	87.23	99.10	95.59	93.65			
PUD		72.27	71.46	57.80	75.57	94.11	87.51	87.05			
SMALL		63.60	60.06	46.00	80.68	99.23	90.93	84.91			
LOW-RESOURCE	OFFICIAL	25.87	18.26	5.16	67.50	93.38	61.07	48.95			
LOW-RESOURCE	CORRECTED	25.33	17.72	4.91	67.50	93.38	57.32	48.95			

Table 2: Results for LAS (+ mono-treebank baseline), MLAS, sentence and word segmentation, UPOS tagging and morphological features (UFEATS). Treebanks sharing a parsing model grouped together; substitute and proxy treebanks for segmentation, tagging, parsing far right (SPECIAL models detailed in the text). Confidence intervals for coloring: $|$ $< \mu - \sigma <$ $|$ $< \mu - \mathrm{SE} < \mu < \mu + \mathrm{SE} <$ $|$ $< \mu + \sigma <$ $|$.

($\mu \pm \sigma$). Finally, scores that are no longer valid because of the Thai UPOS tagger are crossed out in yellow cells, and corrected scores are added where relevant.

Looking first at the LAS scores, we see that our results are significantly above the mean for all aggregate sets of treebanks (ALL, BIG, PUD, SMALL, LOW-RESOURCE) with an especially strong result for the low-resource group (even after setting the Thai score to 0.00). If we look at specific languages, we do particularly well on low-resource languages like Breton, Buryat, Kazakh and Kurmanji, but also on languages like Arabic, Hebrew, Japanese and Chinese, where we benefit from having better word segmentation than most other systems. Our results are significantly worse than the mean only for Afrikaans AfriBooms, Old French SRCMF, Galician CTG, Latin PROIEL, and Portuguese Bosque. For Galician and Portuguese, this may be the effect of lower word segmentation and tagging accuracy.

To find out whether our multi-treebank and multi-lingual models were in fact beneficial for parsing accuracy, we ran a post-evaluation experiment with one model per test set, each trained only on a single treebank. We refer to this as the mono-treebank baseline, and the LAS scores can be found in the second (uncolored) LAS column in Table 2. The results show that merging treebanks and languages did in fact improve parsing accuracy in a remarkably consistent fashion. For the 64 test sets that were parsed with a multi-treebank model, only four had a (marginally) higher score with the mono-treebank baseline model: Estonian EDT, Russian SynTagRus, Slovenian SSJ, and Turkish IMST. Looking at the aggregate sets, we see that, as expected, the pooling of resources helps most for LOW-RESOURCE (25.33 vs. 17.72) and SMALL (63.60 vs. 60.06), but even for BIG there is some improvement (80.21 vs. 79.61). We find these results very encouraging, as they indicate that our treebank embedding method is a reliable method for pooling training data both within and across languages. It is also worth noting that this method is easy to use and does not require extra external resources used in most work on multilingual parsing, like multilingual word embeddings (Ammar et al., 2016) or linguistic re-write rules (Aufrant et al., 2016) to achieve good results.

Turning to the MLAS scores, we see a very similar picture, but our results are relatively speaking stronger also for PUD and SMALL. There are a few striking reversals, where we do significantly better than the mean for LAS but significantly worse for MLAS, including Buryat BDT, Hebrew HTB and Ukrainian IU. Buryat and Ukrainian are languages for which we use a multilingual model for parsing, but not for UPOS tagging and morphological features, so it may be due to sparse data for tags and morphology, since these languages have very little training data. This is supported by the observation that low-resource languages in general have a larger drop from LAS to MLAS than other languages.

For sentence segmentation, the Uppsala system achieved the second best scores overall, and results are significantly above the mean for all aggregates except SMALL, which perhaps indicates a sensitivity to data sparseness for the data-driven joint sentence and word segmenter (we see the same pattern for word segmentation). However, there is a much larger variance in the results than for the parsing scores, with altogether 23 treebanks having scores significantly below the mean .

For word segmentation, we obtained the best results overall, strongly outperforming the mean for all groups except SMALL. We know from previous work (Shao et al., 2018) that our word segmenter performs well on more challenging languages like Arabic, Hebrew, Japanese, and Chinese (although we were beaten by the Stanford team for the former two and by the HIT-SCIR team for the latter two). By contrast, it sometimes falls below the mean for the easier languages, but typically only by a very small fraction (for example 99.99 vs. 100.00 for 3 treebanks). Finally, it is worth noting that the maximum-matching segmenter developed specifically for Thai achieved a score of 69.93, which was more than 5 points better than any other system.

Our results for UPOS tagging indicate that this may be the strongest component of the system, although it is clearly helped by getting its input from a highly accurate word segmenter. The Uppsala system ranks first overall with scores more than one std dev above the mean for all aggregates. There is also much less variance than in the segmentation results, and scores are significantly below the mean only for five treebanks: Galician CTG, Gothic PROIEL, Hebrew HTB, Upper Sorbian UFAL, and Portuguese Bosque. For Galician

and Upper Sorbian, the result can at least partly be explained by a lower-than-average word segmentation accuracy.

The results for morphological features are similar to the ones for UPOS tagging, with the best overall score but with less substantial improvements over the mean. The four treebanks where scores are significantly below the mean are all languages with little or no training data: Upper Sorbian UFAL, Hungarian Szeged, Naija NSC and Ukrainian IU.

All in all, the 2018 edition of the Uppsala parser can be characterized as a system that is strong on segmentation (especially word segmentation) and prediction of UPOS tags and morphological features, and where the dependency parsing component performs well in low-resource scenarios thanks to the use of multi-treebank models, both within and across languages. For what it is worth, we also seem to have the highest ranking single-parser transition-based system in a task that is otherwise dominated by graph-based models, in particular variants of the winning Stanford system from 2017 (Dozat et al., 2017).

8 Extrinsic Parser Evaluation

In addition to the official shared task evaluation, we also participated in the 2018 edition of the Extrinsic Parser Evaluation Initiative (EPE) (Fares et al., 2018), where parsers developed for the CoNLL 2018 shared task were evaluated with respect to their contribution to three downstream systems: biological event extraction, fine-grained opinion analysis, and negation resolution. The downstream systems are available for English only, and we participated with our English model trained on English EWT, English LinES and English GUM, using English EWT as the proxy.

In the extrinsic evaluation, the Uppsala system ranked second for event extraction, first for opinion analysis, and 16th (out of 16 systems) for negation resolution. Our results for the first two tasks are better than expected, given that our system ranks in the middle with respect to intrinsic evaluation on English (9th for LAS, 6th for UPOS). By contrast, our performance is very low on the negation resolution task, which we suspect is due to the fact that our system only predicts universal part-of-speech tags (UPOS) and not the language specific PTB tags (XPOS), since the three systems that only predict UPOS are all ranked at the bot-

tom of the list.

9 Conclusion

We have described the Uppsala submission to the CoNLL 2018 shared task, consisting of a segmenter that jointly extracts words and sentences from a raw text, a tagger that provides UPOS tags and morphological features, and a parser that builds a dependency tree given the words and tags of each sentence. For the parser we applied multi-treebank models both monolingually and multilingually, resulting in only 34 models for 82 treebanks as well as significant improvements in parsing accuracy especially for low-resource languages. We ranked 7th for the official LAS and MLAS scores, and first for the unofficial scores on word segmentation, UPOS tagging and morphological features.

Acknowledgments

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations*.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah Smith. 2016. Many Languages, One Parser. *Transactions of the Association for Computational Linguistics* 4:431–444.

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2016. Zero-resource Dependency Parsing: Boosting Delexicalized Cross-lingual Transfer with Linguistic Knowledge. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*. pages 119–130.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473* .

Bernd Bohnet, Ryan McDonald, Goncalo Simoes, Daniel Andor, Emily Pitler, and Joshua Maynez.

2018. Morphosyntactic Tagging with a Meta-BiLSTM Model over Context Sensitive Token Encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Lingustics* 5:135–146.

Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 740–750.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017a. From Raw Text to Universal Dependencies – Look, No Tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2017b. Arc-Hybrid Non-Projective Dependency Parsing with a Static-Dynamic Oracle. In *Proceedings of the 15th International Conference on Parsing Technologies*. pages 99–104.

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.

Gülsen Eryigit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency Parsing of Turkish. *Computational Linguistics* 34.

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 Shared Task on Extrinsic Parser Evaluation. On the downstream utility of English universal dependency parsers. In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*.

William A Gale and Kenneth W Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics* 19(1):75–102.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. pages 249–256.

Alex Graves. 2008. *Supervised Sequence Labelling with Recurrent Neural Networks*. Ph.D. thesis, Technical University Munich.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* .

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'Diagnostic Classifiers' Reveal how Recurrent and Recursive Neural Networks Process Hierarchical Structure. *Journal of Artificial Intelligence Research* 61:907–926.

Jurgita Kapociute-Dzikiene, Joakim Nivre, and Algis Krupavicius. 2013. Lithuanian dependency parsing with rich morphological features. In *Proceedings of the fourth workshop on statistical parsing of morphologically-rich languages*. pages 12–21.

Eliyahu Kiperwasser and Yoav Goldberg. 2016a. Easy-First Dependency Parsing with Hierarchical Tree LSTMs. *Transactions of the Association for Computational Linguistics* 4:445–461.

Eliyahu Kiperwasser and Yoav Goldberg. 2016b. Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. *Transactions of the Association for Computational Linguistics* 4:313–327.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL, Demo and Poster Sessions*. Prague, Czech Republic, pages 177–180.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*. pages 3111–3119.

Joakim Nivre. 2008. Algorithms for Deterministic Incremental Dependency Parsing. *Computational Linguistics* 34(4):513–553.

Joakim Nivre. 2009. Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP)*. pages 351–359.

Joakim Nivre, Mitchell Abrams, Željko Agić, et al. 2018. Universal Dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. http://hdl.handle.net/11234/1-2837.

Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In Evangelos

Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation. Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*. pages 268–299.

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2018. Universal Word Segmentation: Implementation and Interpretation. *Transactions of the Association for Computational Linguistics* 6:421–435.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-Based Joint Segmentation and POS Tagging for Chinese using Bidirectional RNN-CRF. In *The 8th International Joint Conference on Natural Language Processing*. pages 173–183.

Aaron Smith, Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2018. An Investigation of the Interactions Between Pre-Trained Word Embeddings, Character Models and POS Tags in Dependency Parsing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser Training with Heterogeneous Treebanks. In *Proceedings of the 56th Annual Meeting of the ACL, Short papers*. pages 619–625.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA).

Pak-Kwong Wong and Chorkin Chan. 1996. Chinese Word Segmentation Based on Maximum Matching and Word Binding Force. In *Proceedings of the 16th International Conference on Computational Linguistics*. pages 200–203.

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*.

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 Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung,

Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Hěctor Martínez Alonso, 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 Strnadova, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.