Joint Learning of POS and Dependencies for Multilingual Universal Dependency Parsing

Zuchao Li[1,2,*], **Shexia He**[1,2,*], **Zhuosheng Zhang**[1,2], **Hai Zhao**[1,2,†]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
{charlee,heshexia,zhangzs}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

This paper describes the system of team *LeisureX* in the *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Our system predicts the part-of-speech tag and dependency tree jointly. For the basic tasks, including tokenization, lemmatization and morphology prediction, we employ the official baseline model (UDPipe). To train the low-resource languages, we adopt a sampling method based on other rich-resource languages. Our system achieves a macro-average of 68.31% LAS F1 score, with an improvement of 2.51% compared with the UDPipe.

1 Introduction

The goal of Universal Dependencies (UD) (Nivre et al., 2016; Zeman et al., 2017) is to develop multilingual treebank, whose annotations of morphology and syntax are cross-linguistically consistent. In this paper, we describe our system for the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018), and we focus only on the subtasks of part-of-speech (POS) tagging and dependency parsing. For the intermediate steps, including tokenization, lemmatization and morphology prediction, we tackle them by the official baseline model (UDPipe)[1].

Dependency parsing that aims to predict the existence and type of linguistic dependency relations between words, is a fundamental part in natural language processing (NLP) tasks (Li et al., 2018c; He et al., 2018). Many referential natural language processing studies (Zhang et al., 2018; Bai and Zhao, 2018; Cai et al., 2018; Li et al., 2018b; Wang et al., 2018; Qin et al., 2017) can also contribute to the universal dependency parsing system. Universal dependency parsing focuses on learning syntactic dependency structure over many typologically different languages, even low-resource languages in a real-world setting. Within the dependency parsing literature, there are two dominant techniques, graph-based (McDonald et al., 2005; Ma and Zhao, 2012; Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017) and transition-based parsing (Nivre, 2003; Dyer et al., 2015; Zhang et al., 2017). Graph-based dependency parsers enjoy the advantage of the global search which learns the scoring functions for all possible parsing trees to find the globally highest scoring one while transition-based dependency parsers build dependency trees from left to right incrementally, which makes the series of multiple choice decisions locally.

In our system, we adopt the transition-based dependency parsing in view of its relatively lower time complexity. Our system implements universal dependency parsing based on the stack-pointer networks (STACKPTR) parser introduced by (Ma et al., 2018). Furthermore, previous work (Straka et al., 2016; Nguyen et al., 2017) showed that POS tags are helpful to dependency parsing. In particular, (Nguyen et al., 2017) pointed out that parsing performance could be improved by the merit of accurate POS tags and the context of syntactic parse tree could help resolve POS ambiguities. Therefore, we seek to jointly learn POS tagging and dependency parsing.

* These authors made equal contribution.† Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. 61672343 and No. 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04).

[1]https://ufal.mff.cuni.cz/udpipe/

Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 65–73
Brussels, Belgium, October 31 – November 1, 2018.
https://doi.org/10.18653/v1/K18-2006

As Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have shown significant representational effectiveness to a wide range of NLP tasks, we leverage bidirectional LSTMs (BiLSTM) to learn shared representations for both POS tagging and dependency parsing. In addition, to train the low-resource languages, we adopt a sampling method based on other rich-resource languages.

In terms of all the above model improvement, compared to the UDPipe baseline, our system achieves a macro-average of 68.31% LAS F1 score, with an improvement of 2.51% in this task.

2 Our Model

In this section, we describe our joint model[2] for POS tagging and dependency parsing in the CoNLL 2018 Shared Task, which is built on the STACKPTR parser introduced by (Ma et al., 2018). Our model is mainly composed of three components, the representation (Section 2.1), POS tagger (Section 2.2) and dependency parser (Section 2.3). Figure 1 illustrates the overall model.

2.1 Representation

Representation is a key component in various NLP models, and good representations should ideally model both complex characteristics and linguistic contexts. In our system, we follow the bidirectional LSTM-CNN architecture (BiLSTM-CNNs) (Chiu and Nichols, 2016; Ma and Hovy, 2016), where CNNs encode word information into character-level representation and BiLSTM models context information of each word.

Character Level Representation Though word embedding is popular in many existing parsers, they are not ideal for languages with high out-of-vocabulary (OOV) ratios. Hence, our system introduces the character-level (Li et al., 2018a) representation to address the challenge. Formally, given a word $w = \{BOW, c_1, c_2, ..., c_n, EOW\}$, where two special BOW (begin-of-word) and EOW (end-of-word) tags indicate the begin and end positions respectively, we use the CNN to extract character-level representation as follows:

$$e^c = MaxPool(Conv(w))$$

where the CNN is similar to the one in (Chiu and Nichols, 2016), but we use only characters as the inputs to CNN, without character type features.

Word Level Representation Word embedding is a standard component of most state-of-the-art NLP architectures. Due to their ability to capture syntactic and semantic information of words from large scale unlabeled texts, we pre-train the word embeddings from the given training dataset by word2vec (Mikolov et al., 2013) toolkit. For low-resource languages without available training data, we sample the training dataset from similar languages to generate a mixed dataset.

2.2 POS Tagger

To enrich morphological information, we also incorporate UPOS tag embeddings into the representation. Therefore, we jointly predict the UPOS tag in our system. The architecture for the POS tagger in our model is almost identical to that of the parser (Dozat et al., 2017). The tagger uses a BiLSTM over the concatenation of word embeddings and character embeddings:

$$s_i^{pos} = BiLSTM^{pos}(e_i^w \odot e_i^c)$$

Then we calculate the probability of tag for each type using affine classifiers as follows:

$$\begin{aligned} h_i^{pos} &= MLP^{pos}(s_i^{pos}) \\ r_i^{pos} &= W^{pos} h_i^{pos} + b^{pos} \\ y_i^{pos} &= arg\max(r_i) \end{aligned}$$

The tag classifier is trained jointly using cross-entropy losses that are summed together with the dependency parser loss during optimization.

Context-sensitive Representation In order to integrate contextual information, we concatenate the character embedding e_c, pre-trained word embedding e_w and UPOS tag embedding e_{pos}, then feed them into the BiLSTM. We take the bidirectional vectors at the final layer as the context-sensitive representation:

$$\begin{aligned} \overrightarrow{s_i} &= LSTM_{forward}(e_i^w \odot e_i^c \odot e_i^{pos}) \\ \overleftarrow{s_i} &= LSTM_{backward}(e_i^w \odot e_i^c \odot e_i^{pos}) \\ s_i &= \overrightarrow{s_i} \odot \overleftarrow{s_i} \end{aligned}$$

Notably, we use the UPOS tag from the output of our POS tagging model.

[2]Our code will be available here: https://github.com/bcmi220/joint_stackptr.

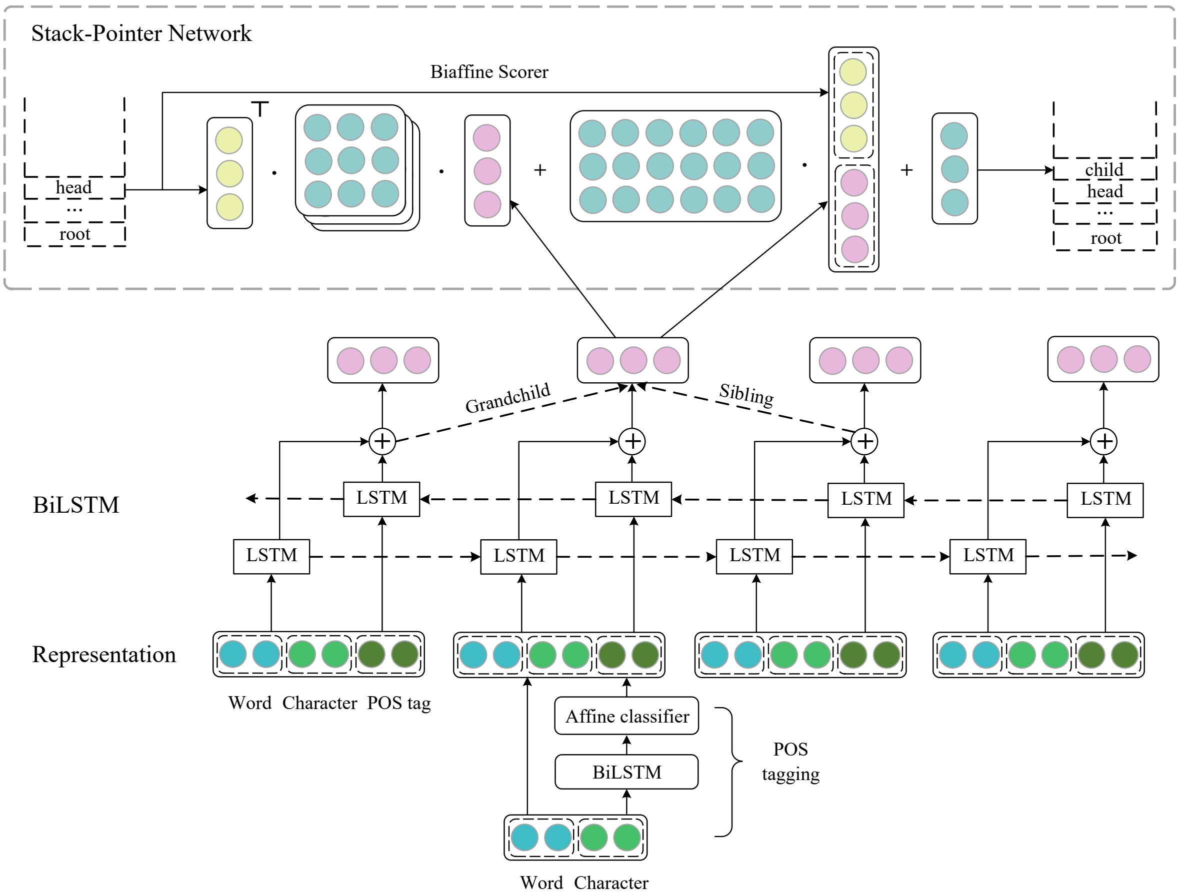

Figure 1: The joint model for POS tagging and dependency parsing.

2.3 Dependency Parsing

The universal dependency parsing component of our system is built on the current state-of-the-art approach STACKPTR, which combines pointer networks (Vinyals et al., 2015) with an internal stack for tracking the status of depth-first search. It benefits from the global information of the sentence and all previously derived subtree structures, and removes the left-to-right restriction in classical transition-based parsers.

The STACKPTR parser mainly consists of two parts: encoder and decoder. The encoder based on BiLSTM-CNNs architecture takes the sequence of tokens and their POS tags as input, then encodes it into encoder hidden state s_i. The internal stack σ is initialized with dummy *ROOT*. For decoder (a uni-directional RNN), it receives the input from last step and outputs decoder hidden state h_t. The pointer neural network takes the top element w_h in the stack σ at each timestep t as current head to select a specific child w_c with biaffine attention mechanism (Dozat and Manning, 2017) for attention score function in all possible head-dependent pairs. Then the child w_c will be pushed onto the stack σ for next step when $c \neq h$, otherwise it indicates that all children of the current head h have been selected, therefore the head w_h will be popped out of the stack σ. The attention scoring function used is given as follows and the pointer neural network uses a^t as pointer to select the child element:

$$e_i^t = h_t^T \mathbf{W} s_i + \mathbf{U}^T h_t + \mathbf{V}^T s_i + \mathbf{b}$$
$$a^t = softmax(e^t)$$

More specifically, the decoder maintains a list of available words in test phase. For each head h at each decoding step, the selected child will be removed from the list to make sure that it cannot be selected as a child of other head words.

Given a dependency tree, there may be multiple children for a specific head. This results in more than one valid selection for each time step, which might confuse the decoder. To address this problem, the parser introduces an inside-outside order to utilize second-order sibling information, which has been proven to be an important feature for parsing process (McDonald and Pereira, 2006; Koo and Collins, 2010). To utilize the second-order information, the parser replaces the input of decoder from s_i as follows:

$$\beta_i = s_s \circ s_h \circ s_i$$

where s and h indicate the sibling and head index of node i, $\circ$ is the element-wise sum operation to ensure no additional model parameters.

2.4 Loss Function

The training objective of pur system is to learn the probability of UPOS tags $P_{\theta^{pos}}(y_{pos}|x)$ and the dependency trees $P_{\theta^{dep}}(y_{dep}|x, y'_{pos})$. Given a sentence x, the probabilities are factorized as:

$$P_{\theta^{pos}}(y_{pos}|x) = \sum_{i=1}^{k} P_{\theta^{pos}}(p_i|x)$$

$$y'_{pos} = \arg \max_{y_{pos} \in Y_{pos}} (P_{\theta^{pos}}(y_{pos}|x))$$

$$P_{\theta^{dep}}(y_{dep}|x, y'_{pos}) = \sum_{i=1}^{k} P_{\theta^{dep}}(p_i|p_{<i}, x, y'_{pos})$$

$$= \prod_{i=1}^{k} \prod_{j=1}^{l_i} P_{\theta^{dep}}(c_{i,j}|c_{i,<j}, p_{<i}, x, y'_{pos})$$

where θ^{pos} and θ^{dep} represent the model parameters respectively. $p_{<i}$ denotes the preceding dependency paths that have already been generated. $c_{i,j}$ represents the j_{th} word in p_i and $c_{i,j}$ denotes all the proceeding words on the path p_i.

Therefore, the whole loss is the sum of three objectives:

$$Loss = Loss_{pos} + Loss_{arc} + Loss_{label}$$

where the $Loss_{pos}$, $Loss_{arc}$ and $Loss_{label}$ are the conditional likehood of their corresponding target, using the cross-entropy loss. Specifically, we train a dependency label classifier following Dozat and Manning (2017), which takes the dependency head-child pair as input features.

3 System Implements

Our system focuses on three targets: the UPOS tag, dependency arc and dependency relation. Therefore, we rely on the UDPipe model (Straka et al., 2016) to provide a pipeline from raw text to basic dependency structures, including a tokenizer, tagger and the dependency predictor.

Treebank	Sampling
Breton KEB	English, Irish
Czech PUD	Czech PDT
English PUD	English EWT
Faroese OFT	Norwegian, English, Danish, Swedish, German, Dutch
Finnish PUD	Finnish TDT
Japanese Modern	Japanese GSD
Naija NSC	English
Swedish PUD	Swedish Talbanken
Thai PUD	English, Chinese, Hindi, Vietnamese

Table 1: Language substitution for treebanks without training data

For treebanks with non-empty training dataset (including treebanks whose training set is very small), we utilize the baseline model UDPipe trained on corresponding treebank, which has been provided by the organizer. For the remaining nine treebanks without training data, we construct the train dataset by sampling from the other training datasets according to the language similarity inspired by (Zhao et al., 2009, 2010; Wang et al., 2015, 2016), as detailed in Table 1.

Our system adopts the hyper-parameter configuration in (Ma et al., 2018), with a few exceptions. We initialize word vectors with 50-dimensional pretrained word embeddings, 100-dimensional tag embeddings and 512-dimensional recurrent states (in each direction). Our system drops embeddings and hidden states independently with 33% probability. We optimize with Adam (Kingma and Ba, 2015), setting the learning rate to $1e^{-3}$ and $\beta_1 = \beta_2 = 0.9$. Moreover, we train models for up to 100 epochs with batch size 32 on 3 NVIDIA GeForce GTX 1080Ti GPUs with 200 to 500 sentences per second and occupying 2 to 3 GB graphic memory each model. A full run over the test datasets on the TIRA virtual machine (Potthast et al., 2014) takes about 12 hours.

4 Results

Table 2 reports the official evaluation results of our system in several metrics of treebanks from the CoNLL 2018 shared task (?). For dependency parsing, our model outperforms the baseline

Results	Ours	Baseline	Best
LAS	68.31	65.80	75.84
MLAS	53.70	52.42	61.25
BLEX	58.42	55.80	66.09
UAS	74.03	71.64	80.51
CLAS	63.85	60.77	72.36
UPOS	87.15	87.32	90.91
XPOS	83.91	85.00	86.67
Morphological features	83.46	83.74	87.59
Morphological tags	76.68	77.62	80.30
Lemmas	87.77	87.84	91.24
Sentence segmentation	83.01	83.01	83.87
Word segmentation	96.97	96.97	98.18
Tokenization	97.39	97.39	98.42

Table 2: Results on all treebanks.

with absolute gains (1.28-3.08%) on average LAS, UAS, MLAS and CLAS. These results show that our joint model could improve the performance of universal dependency parsing. Surprisingly, in the case of POS tagging, our joint model obtains lower averaged accuracy in both UPOS and XPOS. The possible reason for performance degradation may be that we select all hyper-parameters based on English and do not tune them individually.

Furthermore, we also compare the performance of our system with the baseline and the best scorer on big treebanks (Table 3), PUD treebanks (Table 4), low-resource languages (Table 5), respectively.

Since our model applies the baseline model for tokenization and segmentation, we show all results of focused metrics on each treebank in Table 6. In addition, we compare our model with the best and the average results of top ten models on each treebank, using LAS F1 for the evaluation metric, as shown in Figure 2.

5 Conclusion

In this paper, we describe our system in the CoNLL 2018 shared task on UD parsing. Our system uses a transition-based neural network architecture for dependency parsing, which predicts the UPOS tag and dependencies jointly. Combining pointer networks with an internal stack to track the status of the top-down, depth-first search in the parsing decoding procedure, the STACKPTR parser is able to capture information from the whole sentence and all the previously derived subtrees, removing the left-to-right restriction in classical transition-based parsers, while maintaining

Results	Ours	Baseline	Best
LAS	77.98	74.14	84.37
MLAS	63.79	61.27	72.67
BLEX	68.55	64.67	75.83
UAS	82.27	78.78	87.61
CLAS	73.59	69.13	81.29
UPOS	93.71	93.71	96.23
XPOS	91.81	91.81	95.16
Morphological features	90.85	90.85	94.14
Morphological tags	87.56	87.56	91.50
Lemmas	93.34	93.34	96.08
Sentence segmentation	86.09	86.09	89.52
Word segmentation	98.81	98.81	99.21
Tokenization	99.24	99.24	99.51

Table 3: Results on big treebank only.

Results	Ours	Baseline	Best
LAS	61.05	66.63	74.20
MLAS	41.95	51.75	58.75
BLEX	50.60	54.87	63.25
UAS	67.88	71.22	78.42
CLAS	57.34	61.29	69.86
UPOS	82.45	85.23	87.51
XPOS	35.66	54.27	55.98
Morphological features	78.89	83.41	87.05
Morphological tags	34.68	50.32	51.90
Lemmas	82.24	83.37	85.76
Sentence segmentation	75.53	75.53	76.04
Word segmentation	92.61	92.61	94.57
Tokenization	92.61	92.61	94.57

Table 4: Results on PUD treebank only.

Results	Ours	Baseline	Best
LAS	17.16	17.17	27.89
MLAS	3.43	3.44	6.13
BLEX	7.63	7.63	13.98
UAS	30.07	30.08	39.23
CLAS	13.42	13.42	22.18
UPOS	45.17	45.20	61.07
XPOS	54.68	54.23	54.73
Morphological features	38.03	38.03	48.95
Morphological tags	25.86	25.72	25.91
Lemmas	54.25	54.25	64.42
Sentence segmentation	65.99	65.99	67.50
Word segmentation	84.95	84.95	93.38
Tokenization	85.76	85.76	93.34

Table 5: Results on low-resource languages only.

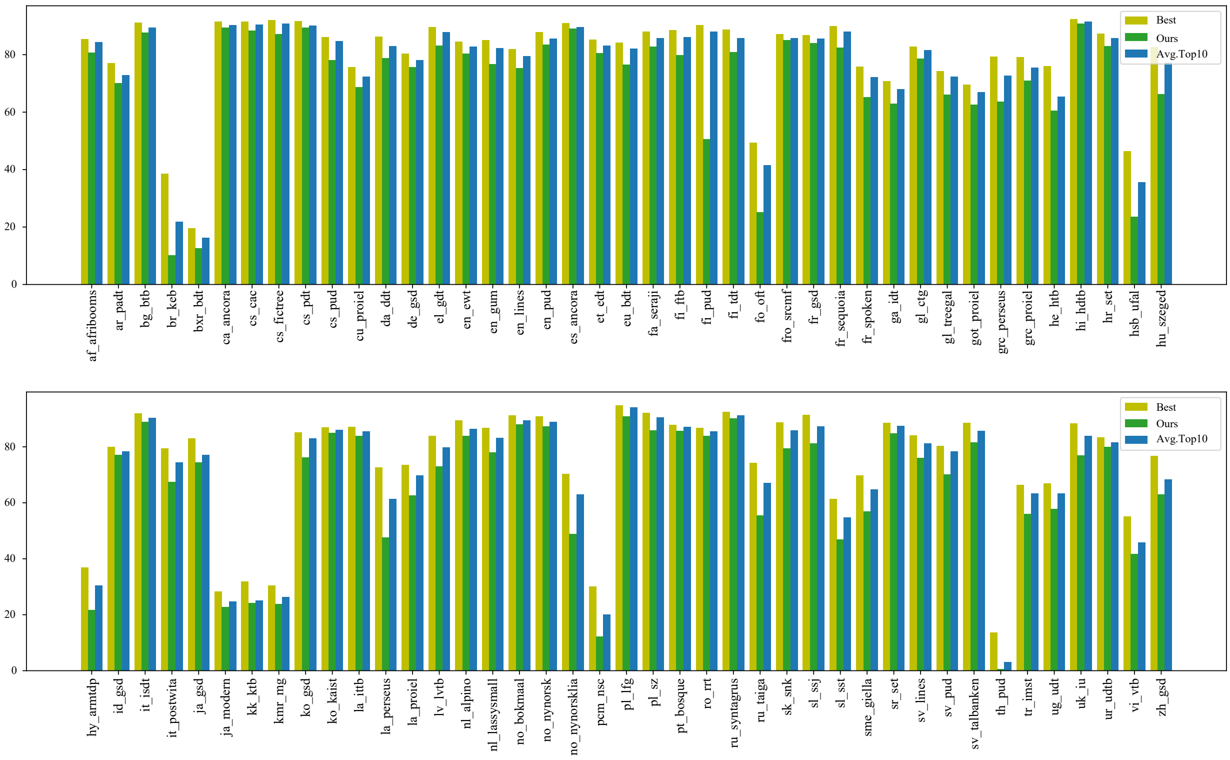

Figure 2: LAS F1 score per treebank. For comparison, we include the best official result and the average of the top ten results on each treebank.

linear parsing steps. Furthermore, our model is single instead of ensemble, and it does not utilize lemmas or morphological features. Results show that our system achieves 68.31% in macro-averaged LAS F1-score on the official blind test. Further improvements could be obtained by multilingual embeddings and adopting ensemble methods.

References

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 571–583.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 2753–2765.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4:357–370.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. *ICLR* .

Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* pages 20–30.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory pages 334–343.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 2061–2071.

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

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics* 4:313–327.

	UPOS	UAS	LAS	MLAS		UPOS	UAS	LAS	MLAS
af_afribooms	95.12	84.64	80.75	66.96	ar_padt	89.34	74.45	70.11	57.21
bg_btb	97.72	91.24	87.69	77.56	br_keb	30.74	27.80	10.25	0.37
bxr_bdt	41.66	29.20	12.61	2.09	ca_ancora	98.00	91.87	89.38	80.87
cs_cac	98.32	91.07	88.46	74.28	cs_fictree	97.28	91.07	87.12	71.98
cs_pdt	98.21	91.59	89.37	78.20	cs_pud	94.67	84.09	78.17	59.57
cu_proiel	93.70	75.18	68.68	55.36	da_ddt	95.44	82.21	78.74	67.34
de_gsd	91.58	80.31	75.73	36.39	el_gdt	95.63	86.64	83.17	65.02
en_ewt	93.62	83.32	80.46	70.58	en_gum	93.24	81.09	76.68	63.05
en_lines	94.71	80.71	75.26	65.04	en_pud	94.15	86.77	83.49	70.23
es_ancora	98.14	91.35	89.09	81.01	et_edt	95.50	84.18	80.59	70.39
eu_bdt	92.34	81.06	76.49	60.75	fa_seraji	96.01	86.76	82.78	75.38
fi_ftb	92.28	84.23	79.83	66.53	fi_pud	84.86	62.87	50.67	36.39
fi_tdt	94.37	84.72	80.88	70.42	fo_oft	44.66	42.64	25.19	0.36
fro_srcmf	94.30	90.32	85.15	75.66	fr_gsd	95.75	87.25	84.08	74.58
fr_sequoia	95.84	85.16	82.50	71.23	fr_spoken	92.94	71.81	65.30	52.73
ga_idt	89.21	72.66	62.93	37.66	gl_ctg	96.26	81.60	78.60	65.00
gl_treegal	91.09	71.61	66.16	49.13	got_proiel	94.31	69.71	62.62	48.19
grc_perseus	82.37	70.08	63.68	33.28	grc_proiel	95.87	75.19	71.05	52.44
he_htb	80.87	64.90	60.53	46.03	hi_hdtb	95.75	94.18	90.83	72.03
hr_set	96.33	88.39	83.06	60.93	hsb_ufal	65.75	35.02	23.64	3.55
hu_szeged	90.59	73.91	66.23	50.36	hy_armtdp	65.40	36.81	21.79	6.84
id_gsd	92.99	83.49	77.12	64.70	it_isdt	97.05	91.01	88.91	79.66
it_postwita	93.94	72.74	67.48	54.38	ja_gsd	87.85	76.14	74.43	60.32
ja_modern	48.44	29.36	22.71	8.10	kk_ktb	48.94	39.45	24.21	7.62
kmr_mg	59.31	32.86	23.92	5.47	ko_gsd	93.44	80.91	76.27	68.93
ko_kaist	93.32	87.43	85.11	76.91	la_ittb	97.21	86.64	83.96	73.55
la_perseus	83.34	58.45	47.61	30.16	la_proiel	94.84	68.02	62.62	49.11
lv_lvtb	91.70	78.74	73.13	55.05	nl_alpino	94.04	87.76	83.91	68.47
nl_lassysmall	94.06	82.34	78.13	64.55	no_bokmaal	96.51	90.30	88.11	78.94
no_nynorsk	96.07	89.67	87.26	76.85	no_nynorsklia	85.15	57.92	48.95	37.60
pcm_nsc	44.44	26.11	12.18	4.60	pl_lfg	96.77	93.67	90.94	74.89
pl_sz	95.50	89.64	85.83	64.03	pt_bosque	95.99	88.48	85.80	70.70
ro_rrt	96.62	89.06	83.94	74.60	ru_syntagrus	97.84	92.09	90.28	80.63
ru_taiga	86.53	63.58	55.51	36.79	sk_snk	93.15	83.42	79.43	55.02
sl_ssj	94.46	84.01	81.18	65.00	sl_sst	88.50	54.16	46.95	34.19
sme_giella	87.69	63.80	56.98	46.05	sr_set	96.84	89.50	84.90	70.68
sv_lines	93.97	81.32	76.04	59.25	sv_pud	90.12	76.30	70.19	35.44
sv_talbanken	95.36	85.27	81.57	71.64	th_pud	5.65	0.71	0.62	0.01
tr_imst	91.64	64.02	56.07	44.49	ug_udt	87.48	71.29	57.89	37.46
uk_iu	94.80	81.43	77.01	56.96	ur_udtb	92.13	86.14	79.99	51.65
vi_vtb	75.29	47.32	41.77	34.18	zh_gsd	83.47	66.45	63.05	51.64

Table 6: Performances of focused metrics on each treebank.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1–11.

Haonan Li, Zhisong Zhang, Yuqi Ju, and Hai Zhao. 2018a. Neural character-level dependency parsing for Chinese. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018b. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 3203–3214.

Zuchao Li, Shexia He, Jiaxun Cai, Zhuosheng Zhang, Hai Zhao, Gongshen Liu, Linlin Li, and Luo Si. 2018c. A unified syntax-aware framework for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1064–1074.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-pointer networks for dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1403–1414.

Xuezhe Ma and Hai Zhao. 2012. Fourth-order dependency parsing. In *24th International Conference on Computational Linguistics*. page 785.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 91–98.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR Workshop*.

Dat Quoc Nguyen, Mark Dras, and Mark Johnson. 2017. A novel neural network model for joint pos tagging and graph-based dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Vancouver, Canada, pages 134–142.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. Citeseer, pages 149–160.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. Portoro, Slovenia, pages 1659–1666.

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 meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14)*. Berlin Heidelberg New York, pages 268–299.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1006–1017.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. Portoro, Slovenia.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*. pages 2692–2700.

Rui Wang, Masao Utiyama, Isao Goto, Eiichiro Sumita, Hai Zhao, and Bao-Liang Lu. 2016. Converting continuous-space language models into n-gram language models with efficient bilingual pruning for statistical machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing* 15(3):11.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2015. Bilingual continuous-space language model growing for statistical machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23(7):1209–1220.

Rui Wang, Hai Zhao, Sabine Ploux, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2018. Graph-based bilingual word embedding for statistical machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17(4):31.

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*. Association for Computational Linguistics, Brussels, Belgium, pages 1–20.

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, 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 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*. Vancouver, Canada, pages 1–19.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2017. Stack-based multi-layer attention for transition-based dependency parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark, pages 1677–1682.

Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2018. Subword-augmented embedding for cloze reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 1802–1814.

Hai Zhao, Yan Song, and Chunyu Kit. 2010. How large a corpus do we need: Statistical method versus rule-based method. *Training (M)* 8(2.71):0–83.

Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. 2009. Cross language dependency parsing using a bilingual lexicon. 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: Volume 1-Volume 1*. Association for Computational Linguistics, pages 55–63.