

# Multilingual Universal Dependency Parsing from Raw Text with Low-resource Language Enhancement

Yingting Wu<sup>1,2</sup>, Hai Zhao<sup>1,2,\*</sup>, Jia-Jun Tong<sup>1,2</sup>

<sup>1</sup>Department of Computer Science and Engineering, Shanghai Jiao Tong University

<sup>2</sup>Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

wuyingting@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, tong2j1994@gmail.com

## Abstract

This paper describes the system of our team Phoenix for participating *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Given the annotated gold standard data in CoNLL-U format, we train the tokenizer, tagger and parser separately for each treebank based on an open source pipeline tool UDPipe. Our system reads the plain texts for input, performs the pre-processing steps (tokenization, lemmas, morphology) and finally outputs the syntactic dependencies. For the low-resource languages with no training data, we use cross-lingual techniques to build models with some close languages instead. In the official evaluation, our system achieves the macro-averaged scores of 65.61%, 52.26%, 55.71% for LAS, MLAS and BLEX respectively.

## 1 Introduction

Universal Dependencies (UD) (Nivre et al., 2016) is a framework that provides cross-linguistically consistent grammatical annotations for various languages, which enables comparative evaluations for some cross-lingual learning tasks. As a follow up of *CoNLL 2017 UD Shared Task* (Zeman et al., 2017), the goal of *CoNLL 2018 UD Shared Task* (Zeman et al., 2018) is to develop multilingual dependency parsers from raw text for many typologically different languages with training data from

UD project. The task comprises 82 test sets from 57 languages. However, there are a category of low-resource languages that have little or no training data, which requires cross-lingual techniques (Zeman and Resnik, 2008; Tiedemann, 2015) with the help of the data from other languages.

In this paper, we present the system of our team Phoenix for multilingual universal dependency parsing from raw text. The targeted task is a challenging one in terms of deep learning based natural language processing (Wu and Zhao, 2018; Bai and Zhao, 2018; Cai et al., 2018; Li et al., 2018; Zhang et al., 2018; Zhang and Zhao, 2018; He et al., 2018; Wang et al., 2017; Qin et al., 2017). We adopt the trainable open source tool UDPipe 1.2 (Straka et al., 2016; Straka and Straková, 2017) to train the dependency parser for each test set with UD version 2.2 (Nivre et al., 2018) treebanks as training data. There are three main components of our model to perform, tokenization, Part-of-Speech (POS) tagging and dependency parsing. When evaluated on the web interface of TIRA (Potthast et al., 2014) platform, the system reads the raw text for input and chooses the corresponding model for a particular test set with a model selector. After the tokenization and tagging on the raw text, the system finally outputs the syntactic dependencies in the CoNLL-U format. To deal with the low-resource languages which have no training data, some cross-lingual techniques are applied by training with other related or close languages. Our official submission obtains macro-averaged scores of 65.61%, 52.26%, 55.71% for LAS, MLAS and BLEX on all treebanks.

The rest of this paper is organized as follows. Section 2 introduces the architecture overview of our system. Section 3 gives the implementation details and the specific strategies applied for the low-resource languages. Finally, we report and an-

\* 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).

alyze the official results with the three main evaluation metrics in Section 4.

## 2 System Overview

Figure 1 illustrates the overall architecture of our system for training and predicting. In the training procedure, the system takes as input the treebanks (training set) of CoNLL-U format and trains a model for each of them. Every model has three components, tokenizer, tagger and parser. As we train the parser, pretrained word embeddings for word forms in the word2vec format are applied. In the predicting procedure, the system takes as input the raw text (test set) and selects a model according to the language code and treebank code. With the selected model, the system outputs the syntactic head and the type of the dependency relation for each word.

### 2.1 Training Data

Our models are trained by using only UD 2.2 treebanks provided by the *CoNLL 2018 UD Shared Task* without any other additional data. There are 82 test sets from 57 languages, and 61 of the 82 treebanks are large enough to provide training and development sets. However, the other 21 treebanks lack of development data and some of them even have no training data. Among these low-resource treebanks, 7 have training data with still reasonable size; 5 are extra test sets in languages where another large treebank exists; 9 are low-resource languages with no training data available (Breton, Faroese, Naija, Thai) or the training set being just a tiny sample (Armenian, Buryat, Kazakh, Kurmanji, Upper Sorbian). For the treebanks that still have training data but without development data, we split the last 10% of the training data as the development set to tune model hyperparameters even if the training set is very small. To deal with those having no training data, we train the parsers with the treebanks of the same languages or related languages. The details will be given in Section 3.4.

### 2.2 Word Embedding

We adopt pretrained embeddings for word forms with the provided training data by word2vec (Mikolov et al., 2013). The parameter settings of word2vec are shown in Table 1. We use the skip-gram model to train the word vectors with a dimension of 50. The context window is set

to 10 words and the word will be dropped if its frequency is less than twice. After converting CoNLL-U to the horizontal format (replacing spaces within word forms with a Unicode character), we train the word embeddings of each treebank on the UD data for 15 iterations.

Parameters	Value
algorithm	skip-gram
size	50
window	10
min-count	2
iterations	15

Table 1: Parameters for training word embeddings.

### 2.3 Model Selector

Since we train models individually for every languages and treebanks, a model selector is needed to decide which model to use when predicting. The model selector in our system simply reads the json file in the test file folder and assigns the corresponding trained model for each test set according to the language code and treebank code.

## 3 Model Description

### 3.1 Tokenizer

In our system pipeline, the first step is the sentence segmentation and tokenization which is performed jointly in UDPipe. A single-layer bidirectional GRU network is used to train the tokenizer which predicts for each character whether it is the last one in a sentence or the last one in a token. In UD treebanks, the text is structured on several levels: document, paragraph, sentence and token. A MISC feature `SpaceAfter=No` is defined to denote that a given token is not followed by a space. Thus, the tokenizer is trained according to the `SpaceAfter=No` features in the CoNLL-U files.

The parameters used for training the tokenizer are listed in Table 2. The segmenter and tokenizer network employs character embeddings and is trained using dropout both before and after the recurrent units. The GRU dimension, dropout probability and learning rate are tuned on the development set. All the tokenizers are trained for 100 epochs. Other parameters like `tokenize_url` and `allow_spaces` are set as default.

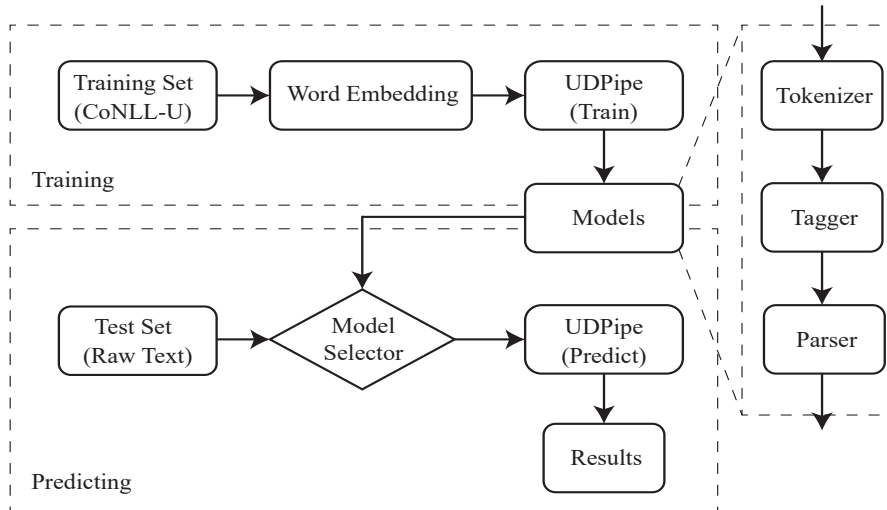


Figure 1: System architecture

Parameters	Value
tokenize_url	1
allow_spaces	1
batch_size	50,100
dimension	24, 64
dropout	0.1, 0.2, 0.3
early_stop	1
epochs	100
learning_rate	0.002, 0.005, 0.01

Table 2: Parameters for training the tokenizers.

Parameters	Value
models	2
templates_1	tagger
guesser_suffix_rules_1	4, 6, 8, 10, 12
guesser_enrich_dictionary_1	4, 5, 6
guesser_prefixes_max_1	0, 4
templates_2	lemmatizer
guesser_suffix_rules_2	4, 6, 8
guesser_enrich_dictionary_2	4, 5, 6
guesser_prefixes_max_2	0, 4
iterations	20
early_stopping	1

Table 3: Parameters for training the taggers.

### 3.2 Tagger

The second step in our system pipeline is to generate some POS tags and other morphological features for the tokenized data, which will be utilized as the input for the final dependency parser. We adopt the built-in tagger in UDPipe, which is

based on an open source morphological analysis tool MorphoDita (Straková et al., 2014). In this step, the tagger will produce the following outputs:

**Lemma:** Lemma or stem of word form.

**UPOS:** Universal POS tag.

**XPOS:** Language-specific POS tag.

**FEATS:** List of morphological features from the universal feature inventory or from a defined language-specific extension.

We use two MorphoDita models to produce different features, whose effectiveness has been verified in (Straka et al., 2016). The first model called *tagger* generates the UPOS, XPOS and FEATS tags while the second one called *lemmatizer* performs lemmatization.

The tagger consists of a guesser and an averaged perceptron. The guesser generates several triplets (UPOS, XPOS, FEATS) for each word according to its last four characters. The averaged perceptron with a fixed set of features disambiguates the generated tags (Straka et al., 2016; Straková et al., 2014).

The structure of the lemmatizer is similar to the tagger. A guesser produces (lemma rule, UPOS) tuples and an averaged perceptron performs disambiguation. The lemmatizer generates a lemma from a word by stripping some affix and adding new affix according to the last four characters of a word and its prefix.

The training parameters of the two models for different treebanks are provided in Table 3.

Parameters	Value
embedding_form	50
embedding_upostag	20
embedding_feats	20
embedding_deprel	20
hidden_layer	200
batch_size	10
iterations	30
l2	0.3, 0.5
learning_rate	0.01, 0.02
learning_rate_final	0.001
structured_interval	0, 8, 10
transition_oracle	static, dynamic, static_lazy
transition_system	projective, link2, swap
use_gold_tags	1

Table 4: Parameters for training the parsers.

### 3.3 Parser

The final parsing step is performed using Parsito (Straka et al., 2015), which is a transition-based parser with a neural-network classifier. The parser supports several transition systems including a projective arc-standard system (Nivre, 2008), a partially non-projective link2 system (Gómez-Rodríguez et al., 2014) and a fully non-projective swap system (Nivre, 2009). Meanwhile, the transition oracles can be configured into static oracles, dynamic oracle for the arc-standard system (Goldberg et al., 2014) or a search-based oracle (Straka et al., 2015).

We use the golden Lemmas, UPOS, XPOS and FEATS tags for both the training and development data when training the parser. The parser employs FORM embeddings of dimension 50, and UPOS, FEATS, DEPREL embeddings of dimension 20. The FORM embeddings are pretrained with word2vec using the training data, and the other embeddings are initialized randomly. All the embeddings are updated for each iteration during training. The hidden layer size is set to 200 and the batch size is limited to 30. All the parsing models are trained for 10 iterations. The training parameters for different datasets are reported in Table 4. The optimal parameters are chosen to maximize the accuracy on the development set.

### 3.4 Low-resource Treebanks

In the UD 2.2 datasets provided by the shared task, there are low-resource treebanks with little or even no training data. As we have stated before, in this work, we mainly focus on the strategy to deal with the treebanks without any training data in this section.

For the ones with other treebanks in the same languages, we trained models both on the mixture of all those treebanks and on the largest treebank as the official baseline did. For those without other treebanks in the same language, the direct solution is to use other related treebanks as training data. Hence, we take advantage of the cross-lingual knowledge and train the mixture models with the treebanks of similar or related languages. Specifically, we manually selected treebanks with similar languages as the ingredients of a mixture dataset according to many factors such as grammar, morphology and vocabulary. The training and development sets of the selected treebanks are merged together on which we train and evaluate the results. The no-training-data treebanks and their corresponding training sets are shown in Table 5.

## 4 Results

The final results are evaluated blindly on TIRA platform. There are three main scoring metrics, LAS, MLAS and BLEX. Our system ranks 19 in LAS, 17 in MLAS and 13 in BLEX on the main metric ranking board. The main evaluation scores of our system on all treebanks, big treebanks, PUD treebanks, small treebanks and low-resource languages are shown in Table 6. Overall, our system gives a similar performance to the BASELINE UDPipe 1.2 system, which is not surprising as we closely followed the hyper-parameter settings and data splitting of the baseline system on big and small treebanks.

As described in Section 3.4, we select training sets differently for the low-resource treebanks including PUD treebanks and other treebanks without training data. Table 7 shows the results comparison with the baseline system on those treebanks. Our system shows a consistent improvement over the baseline model for all the three metrics on PUD treebanks, which suggests that enlarging training data with different types of treebanks of the same language indeed helps building a better model. Our system shows a slight ad-

Low-resource Treebanks	Training Treebanks
Czech PUD	Czech PDT / Czech (PDT, CAC, FicTree)
English PUD	English EWT / English (EWT, GUM, LinES)
Finnish PUD	Finnish TDT / Finnish (TDT, FTB)
Japanese Modern	Japanese GSD
Swedish PUD	Swedish Talbanken / Swedish (Talbanken, LinES)
Breton KEB	Irish (IDT), Latin (ITTB, Persus, PROIEL), Old French (SRCMF), French (GSD, Sequoia, Spoken)
Faroese OFT	Danish (DDT), Norwegian (Bokmaal, Nynorsk, NynorskLIA)
Naija NSC	English (EWT, GUM, LinES), Portuguese (Bosque)
Thai PUD	Hindi (HDTB), Vietnamese (VTB), Chinese (GSD), English (EWT, GUM, LinES)

Table 5: The training data for the low-resource treebanks.

Target Treebanks	LAS	MLAS	BLEX
Big treebanks	73.93	61.12	64.47
PUD treebanks	66.97	52.26	55.69
Small treebanks	54.63	38.38	40.72
Low-resource	16.99	3.02	8.00
All treebanks	65.61	52.26	55.71

Table 6: Evaluation scores of our system on different types of treebanks.

Metrics	PUD Treebanks		Low-resource	
	Ours	Base	Ours	Base
LAS	<b>66.97</b>	66.63	16.99	17.17
MLAS	<b>52.26</b>	51.75	3.02	3.44
BLEX	<b>55.69</b>	54.87	<b>8.00</b>	7.63

Table 7: Comparison of our system (Ours) and baseline system (Base) on PUD treebanks and low-resource languages.

Target Treebanks	Selected Model	LAS	Rank
cs_pud	cs_pdt	80.34	16
en_pud	en_all	79.69	16
fi_pud	fi_tdt	80.19	14
ja_modern	ja_gsd	22.90	6
sv_pud	sv_all	71.75	15
Macro-average		66.97	13

Table 8: LAS F1 scores and rankings of our system on PUD treebanks.

vantage over baseline on low-resource treebanks in BLEX, which indicates that low-resource languages can be trained with similar languages.

Target Treebanks	LAS	Rank
bxr_bdt	9.04	20
hsb_ufal	23.43	21
hy_armtdp	23.37	14
kk_ktb	23.00	14
kmr_mg	19.08	19
br_keb	8.88	20
fo_ofst	29.13	12
pcm_nsc	16.25	8
th_pud	0.75	5
Macro-average	16.99	17

Table 9: LAS F1 scores and rankings of our system on low-resource languages.

Table 8 shows the results on each PUD treebanks and our selected model for testing. The models with suffix ‘\_all’ represent those trained with all treebanks of the same language. During the test phase, we evaluated both models trained with all treebanks of the same language and with the largest treebank of that language, and compared the rounded results to decide which one to take for our final system. The model trained by mixed English and Swedish treebanks with all data of the same languages shows better performance than those of the single largest treebank. However, the models trained by the largest Czech and Finnish treebanks get a higher score. We conjecture that more training data may provide more information for the modeling while too large training sets will also bring noise for a specific domain.



Metrics	F1 Score	Rank
UAS	71.27	19
CLAS	60.62	19
UPOS	87.49	15
XPOS	84.83	13
Morphological features	83.87	11
All morphological tags	77.47	10
Lemmas	87.69	14
Sentence segmentation	82.91	17
Word segmentation	97.03	8
Tokenization	97.46	7

Table 10: Other metrics and rankings of our system.

Table 9 shows the LAS F1 scores and rankings of each low-resource language. Note that our system has a good performance on pcm\_nsc and th\_pud treebanks when most teams get an unsatisfying result. It also proves that our data selection method is effective for improving the model performance to some extent. Table 10 shows the F1 scores and our system rankings on other metrics. In particular, our system ranks first in the Sentence Segmentation F1 score of PUD treebanks.

## 5 Conclusion

In this paper, we describe our system to *CoNLL 2018 UD Shared Task*. In our system, we focus on the accuracy improvement of the low resource treebanks against the baseline. The results of the official blind test show that our system achieves 65.61%, 52.26%, 55.71% in macro-averaged LAS, MLAS and BLEX.

## 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*. Association for Computational Linguistics, pages 571–583. <http://aclweb.org/anthology/C18-1048>.
- 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*. Association for Computational Linguistics, pages 2753–2765. <http://aclweb.org/anthology/C18-1233>.
- Yoav Goldberg, Francesco Sartorio, and Giorgio Satta. 2014. A tabular method for dynamic oracles in transition-based parsing. *Transactions of the association for Computational Linguistics* 2:119–130.
- Carlos Gómez-Rodríguez, Francesco Sartorio, and Giorgio Satta. 2014. [A polynomial-time dynamic oracle for non-projective dependency parsing](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Doha, Qatar, pages 917–927. <http://www.aclweb.org/anthology/D14-1099>.
- 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)*. Association for Computational Linguistics, pages 2061–2071. <http://aclweb.org/anthology/P18-1192>.
- Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. [Seq2seq dependency parsing](#). In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 3203–3214. <http://aclweb.org/anthology/C18-1271>.
- 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*. Association for Computational Linguistics, Suntec, Singapore, pages 351–359. <http://www.aclweb.org/anthology/P/P09/P09-1040>.
- 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*. European Language Resources Association, Portoro, Slovenia, pages 1659–1666.
- Joakim Nivre et al. 2018. [Universal Dependencies 2.2](#). LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague. <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 meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative*. Springer, Berlin Heidelberg New York, pages 268–299. [https://doi.org/10.1007/978-3-319-11382-1\\_22](https://doi.org/10.1007/978-3-319-11382-1_22).
- 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)*. Association for Computational Linguistics, pages 1006–1017. <https://doi.org/10.18653/v1/P17-1093>.
- Milan Straka, Jan Hajič, and Jana Straková. 2016. UD-Pipe: 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*. European Language Resources Association, Portoro, Slovenia.
- Milan Straka, Jan Hajič, Jana Straková, and Jan Hajič jr. 2015. Parsing universal dependency treebanks using neural networks and search-based oracle. In *Proceedings of 14th International Workshop on Treebanks and Linguistic Theories*.
- Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, pages 88–99. <https://doi.org/10.18653/v1/K17-3009>.
- Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, pages 13–18. <http://www.aclweb.org/anthology/P/P14/P14-5003.pdf>.
- Jörg Tiedemann. 2015. Cross-lingual dependency parsing with universal dependencies and predicted pos labels. In *Proceedings of the 3rd International Conference on Dependency Linguistics*. pages 340–349.
- Hao Wang, Hai Zhao, and Zhisong Zhang. 2017. A transition-based system for universal dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, pages 191–197. <https://doi.org/10.18653/v1/K17-3020>.
- Yingting Wu and Hai Zhao. 2018. Finding better subword segmentation for neural machine translation. In *Proceedings of the 17th China National Conference on Computational Linguistics*.
- 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 Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Mäsilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Drogonova, Héctor Martínez Alonso, Çağr Çöltekin, Umut Sulubacak, Hans Uszkor-eit, 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*. Association for Computational Linguistics, Vancouver, Canada, pages 1–19. <http://www.aclweb.org/anthology/K17-3001>.
- Daniel Zeman and Philip Resnik. 2008. Cross-language parser adaptation between related languages. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing Workshop on NLP for Less Privileged Languages*.
- Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 3740–3752. <http://aclweb.org/anthology/C18-1317>.
- Zhuosheng Zhang and Hai Zhao. 2018. One-shot learning for question-answering in gaokao history challenge. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 449–461. <http://aclweb.org/anthology/C18-1038>.