Boosting Dependency Parsing Performance by Incorporating Additional Features for Agglutinative Languages

Mücahit Altıntaş[1,2], A. Cüneyd Tantuğ[1]

[1] *Faculty of Computer Science, Natural Language Processing and Social Robotic Lab, Istanbul Technical University, 34469, Maslak, Istanbul, Turkey*

[2] *Faculty of Engineering, Bayburt University, 69002, Bayburt, Turkey*

Abstract

In recent studies, the use of language models has increased noticeably and has made quite good contributions. However, using the proper representation and taking into account the complementary components are still among the issues to be considered. In this research, the impact of sub-word level sentence piece based word representation on the performance of dependency parsing has been demonstrated for agglutinative languages. Furthermore, we propose to use the sentence representation that holds all meaning of the sentence as an additional feature to improve dependency parsing. Our proposed enhancements are experimented on nine agglutinative languages; Estonian, Finnish, Hungarian, Indonesian, Japanese, Kazakh, Korean, Turkish, and Uyghur. We found that the sentence piece based token encoding has contributed parsing performance for the majority of the experimented languages. Using the entire meaning of the sentence as a complementary feature has enhanced parsing performance for six languages out of nine.

Keywords

agglutinative languages, dependency parsing, sentence piece, sentence representation

1. Introduction

Dependency parsing is one of the core components of natural language computation that identifies syntactic relationships among the words within a sentence. It is crucial for several natural language processing (NLP) downstream tasks. Zhou et al. [1] employed dependency parsing to obtain semantic representation in order to enhance text-to-speech. Luo et al. [2] applied dependency parsing knowledge as supplementary information, which allows the question answering (QA) model to better match within the semantic component of the question. Zhang et al. [3] utilized the encoder outputs of dependency parser as the inputs for the Seq2Seq neural machine translation (NMT) model by training both dependency parsing and machine translation model parameters concurrently. Cai and Lapata [4], Xia et al. [5] reported that syntax-aware representation improves the semantic role labeling (SRL) performance.

In linguistic typology, agglutinative languages are a subcategory of morphologically rich languages that present a significant challenge for NLP research. With their rich morpho-syntax,

The International Conference and Workshop on Agglutinative Language Technologies as a challenge of Natural Language Processing (ALTNLP), June 7-8, Koper, Slovenia

maltintas@itu.edu.tr (M. Altıntaş); tantug@itu.edu.tr (A. C. Tantuğ)

a word may contain many morphemes, each of which is responsible for supplying the word with grammatical function or endowing new meaning. A word may have numerous different surface form, that entails the out of vocabulary (OOV) or data sparsity problems. To abate these problems, sub-word level representations have been proposed in the literature. Dos Santos and Zadrozny [6], Kim et al. [7] reported that characters level word representation improves performances for word-level tasks. Yu et al. [8] proposed to use syllable-level word embedding in morphologically rich languages such as Korean. Bojanowski et al. [9] introduced an extension of the continuous skip-gram model in which words are represented as the sum of the n-gram character vectors. However, agglutinative languages convey grammatical information through inflections, so they tend to have more flexible word order. This case causes discontinuous constituents that impose non-projectivity in dependency structures [10]. Fortunately, splitting their morphemes is simple since each piece of grammatical information is contained in a single morpheme or vice versa. Eryiğit and Oflazer [11] demonstrated that considering morphemes as the primary units of syntactic structure rather than word forms improves parsing accuracy for an agglutinative language, Turkish. Özateş et al. [12] made use of the morpheme information and hand-crafted rules to improve the word vector representation in dependency parsing.

In this paper, we propose two enhancements to increase dependency parsing accuracy of agglutinative languages in particular, but not restricted with them only.

- We employ sub-word level sentence piece [13] based on word representation to capture morphemes more precisely and also attenuate the OOV (out of vocabulary) and data sparsity problems. Sentence piece is a neural network-based universal sub-word tokenizer that is language independent.
- As a complementary feature to token features, we use sentence representation that holds the whole meaning of the sentence. It is based on the fact that sentences with the same meaning but different word orders have the same dependency tree structure.

We investigate the impact of our proposed improvements to dependency parsing accuracy on nine widely used agglutinative languages; Estonian, Finnish, Hungarian, Indonesian, Japanese, Kazakh, Korean, Turkish, and Uyghur.

2. Approach

Our proposed model is an enhancement on the experiment described by Dozat and Manning [14]. The enhanced model comprises an LSTM-based encoder and biaffine classifiers. Sub-word level representations; character based and sentence piece [15] based are obtained by using attention mechanism over hidden states of a single LSTM layer. Three bi-directional LSTM layers are utilized to make the concatenation of token and sub-token embeddings context-aware. Pre-trained word embedding is added to the model after these bi-LSTM layers. Sentence representation that is obtained by concatenating the last hidden states of bi-LSTM and sentence vectors that comes from the pre-trained model is also employed as an extra feature by broadcasting for each word in the sentence. Figure 1 illustrates our proposed neural dependency parser architecture.

To express in formulas, s is a sentence that includes n words and is represented as $s = w_0, w_1, ..., w_n$ where w_0 is added synthetically as the ROOT token. Each word w_i can be represented

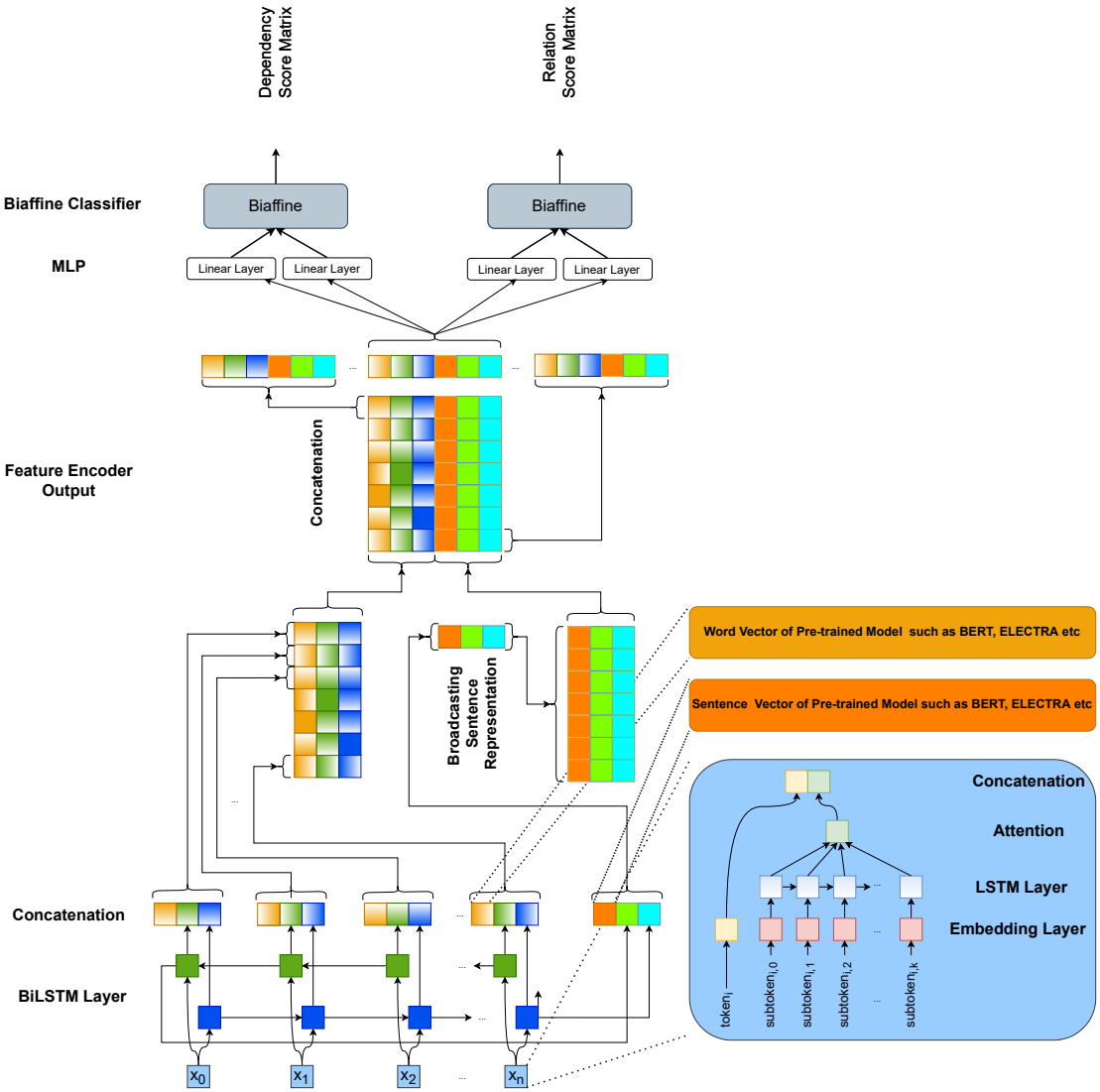

Figure 1: Our neural dependency parser model architecture

by a combination of surface form (u_i), lemma (l_i), POS tag (t_i), morphological feature (m_i), character (c_i), and sentence piece (p_i) based on characteristics of the word, respectively, as given below (Equ. 1).

$$w_i = u_i, l_i, t_i, m_i, c_i, p_i \tag{1}$$

Here, c_i and p_i are sub-word level features of the word while u_i, l_i, t_i and m_i are word level features.

Encoder: The concatenation of word level (Equ. 3) and sub-word level (Equ. 4) embedding vectors yields the vector x_i that is used as input to the bi-LSTM layers. (Equ. 2).

$$x_i = token_i \oplus subtoken_i \tag{2}$$

$$token_i = e(u_i) \oplus e(l_i) \oplus e(t_i) \oplus e(m_i) \tag{3}$$

$$subtoken_i = f(c_i) \oplus f(p_i) \tag{4}$$

A sub-word representation is obtained by using attention on the stacked hidden states of a single layer LSTM. (Equ. 5).

$$f(p) = H_p^T a_p \tag{5}$$

$$a_p = sigmoid(H_p w_p^{attention}) \tag{6}$$

$$H_p = [h_0; h_1; ...; h_P] \tag{7}$$

$$r_i = LSTM((e(p_{i,0}), ..., e(p_{i,P})) \tag{8}$$

$$(\overrightarrow{h_k}, \overrightarrow{h_P}) = split(r_i, k) \tag{9}$$

where $p_i = p_{i,1}, p_{i,2}, ..., p_{i,P}$ is the sequence of the sub-word features of the word and P is the number of sub-word features that may be sentence piece or characters of the word.

A multi-layer Bi-LSTM (Equ. 10) is used to generate contextual word representations over x_is. External contextualized word representation that may be obtained from ELECTRA or BERT is concatenated with right and left hidden states of the corresponding word on the last Bi-LSTM layer(Equ. 12).

$$r = BiLSTM((x_0, ..., x_n)) \tag{10}$$

$$(\overleftarrow{h_i}, \overrightarrow{h_i}), (\overleftarrow{h_0}, \overrightarrow{h_n}) = split(r, i) \tag{11}$$

$$z_i = T(u_i) \oplus \overleftarrow{h_i} \oplus \overrightarrow{h_i} \tag{12}$$

where $T(u_i)$ denotes pre-trained model vector of the word surface form u_i.

To represent a sentence, the pre-trained model sentence embedding vector is concatenated with the final hidden states of the last bi-LSTM layer's backward and forward directions respectively (Equ. 13).

$$sentence_{vector} = T('CLS') \oplus \overleftarrow{h_0} \oplus \overrightarrow{h_n} \tag{13}$$

where $T('CLS')$ provides the sentence representation.

Classifier: Deep bi-affine attention, as proposed in Dozat and Manning [14], is employed as a classifier. Multi-layer perceptron (MLP) are used to get concentrated characteristics of word representation as head (Equ. 15) and dependent (Equ. 14). Then, these representations are input into a bi-affine attention mechanism, which provides a score vector expressing the likelihood of being the parent for each word in the sentence (Equ. 17).

$$h_i^{(arc-dep)} = MLP^{(arc-dep)}(z_i) \tag{14}$$

$$h_i^{(arc-head)} = MLP^{(arc-head)}(z_i) \tag{15}$$

$$H^{(arc-head)} = [h_0^{(arc-head)}; ...; h_n^{(arc-head)}] \tag{16}$$

$$s_i^{(arc)} = biaffine^{(arc)}(H^{(arc-head)}, h_i^{(arc-dep)}) \tag{17}$$

Similarly, another bi-affine classifier is employed to compute the dependency label probabilities of the relevant word with each probable head (Equ. 21).

$$h_i^{(rel-dep)} = MLP^{(rel-dep)}(z_i) \tag{18}$$

$$h_i^{(rel-head)} = MLP^{(rel-head)}(z_i) \tag{19}$$

$$H^{(rel-head)} = [h_0^{(rel-head)}; ...; h_n^{(rel-head)}] \tag{20}$$

$$s_i^{(rel)} = biaffine^{(rel)}(H^{(rel-head)}, h_i^{(rel-dep)}) \tag{21}$$

These two bi-affine classifiers are jointly trained in the training phase with respect to the sum of cross-entropy losses. The Chu-Lie-Edmonds approach is utilized during testing to extract the greatest spanning tree from the resultant score matrices.

3. Experiment and Results

Our proposed enhancements on dependency parsing have been evaluated on nine agglutinative languages, namely, Estonian, Finnish, Hungarian, Indonesian, Japanese, Kazakh, Korean, Turkish, and Uyghur. Table 1 lists some details of utilized treebanks. Indonesian belongs to the Austronesian language family. Official splits of treebanks have been used. All of the scores in this research are acquired on the test set by the associated model, which was trained on the related treebank's training set. Uyghur UDT treebank has no validation set. Thus, we have used the test set to ensure that the training process was not over-training.

Table 1
Some prominent properties of used treebanks in our experiment.

	Estonian EDT	Finnish TDT	Hungarian Szeged	Indonesian GSD	Japanese GSD	Kazakh KTB	Korean GSD	Turkish IMST	Uyghur UDT
Sentence Count in Train Set	24633	8054	910	4477	7050	31	4400	3664	1656
Sentence Count in Test Set	3214	1555	449	557	543	1047	989	983	900
Sentence Count in Validation Set	3125	1364	441	559	507	0	950	988	-
Unique Token Count	80195	53881	13469	20179	21680	4387	35846	17105	12068
Average Token Count in Sentences	14.13	13.35	23.35	21.55	23.90	9.77	12.67	10.26	11.64
Language Family	Uralic	Uralic	Uralic	Austro.	Japonic	Turkic	Korean	Turkic	Turkic

The hyper-parameters are listed in Table 3a. For all languages, the same hyper-parameters were employed. The pre-trained word embeddings have been used in the following order; ELECTRA [16], BERT [17], ELMo, and word2vec. In other words, if there is no trained ELECTRA language model (LM) for the relevant language, BERT LM has been used, if BERT LM does

not exist, ELMo has been employed to obtain pre-trained word vector, if there is no ELMo, word2vec pre-trained vectors have been exploited. Table 3b explains which pre-trained word vectors have been used for each language. We only utilized the corresponding vector of the first word piece per word, disregarding the remainders for words that may consist of multiple word pieces in BERT and ELECTRA. We have used the Xaiver uniform initialization [18] with the same random seed for all our experiments.

Table 2
(a)Hyper-parameters and (b)how pre-trained word vectors are obtained per language

Hyper Parameters	
Num. of word-level Bi-LSTM layers	3
Word embedding dim.	75
Tag embedding dim.	50
Sub-token embedding dim.	100
Arc vector dim.	512
Label vector dim.	128
Dropout rate	0.5
Optimizer	AdamW
$\beta 1$	0.900
$\beta 2$	0.999
Learning rate	5e-5

(a)

Language	**Pre-trained Vec.**	**Ref.**
Estonian	BERT	
Finnish	BERT	[19]
Hungarian	BERT	[20]
Indonesian	BERT	
Japanese	BERT	
Kazakh	word2vec	[21]
Korean	ELECTRA	[22]
Turkish	ELECTRA	
Uyghur	ELMo	[23]

(b)

To obtain sub-token based word representations, the first ten sentence pieces of words and the first twenty characters of words have been used and the rests have been ignored. During training, one more layer is fine-tuned in each iteration, starting from the final layer of the pre-trained model. The AdamW optimizer [24] is employed with a linear schedule warm-up.

As evaluation metrics, the word-based unlabeled attachment score (UAS) and labeled attachment score (LAS) are utilized. CoNLL 2018 UD Shared Task evaluation script[1] has been used to calculate UAS and LAS.

To manifest the impact of our proposed enhancements; sentence piece based word representation and complimentary sentence representation, we provide the UAS and LAS of our three models. Our benchmark model uses sentence piece based word representation but not sentence representation. The other two models are the model without using sentence piece based word representation, and the model with using sentence representation. Table 4 shows the performances of our models and some previous models that are trained with gold annotations for the same treebanks; Udify [25], UDPipe 2.0 [26], UDPipe 2.0 with using BERT and Flair pre-trained word embeddings [26].

The UDify [25] model intends to create a single parsing model for 75 languages in the UD dataset, leveraging the multilingual BERT model which has been trained on the top largest 104 languages on Wikipedias. This parser demonstrates that languages with minimal labeled data can be parsed by using data from other languages. The encoder output was obtained using an

[1]The evaluation script can be downloaded from http://universaldependencies.org/conll18/conll18_ud_eval.py

Table 4
The UAS and LAS of our models and previous models on the test set of the corresponding treebank

		UDify [25]	UDPipe 2.0 [26]	UDPipe 2.0 with BERT+Flair [26]	Our Model w/o Sentence Piece	Our Model	Our Model with Sent. Repr.
Estonian EDT	UAS	89.53	88.00	89.46	**90.97**	90.72	90.81
	LAS	86.67	85.18	86.77	**88.38**	88.18	88.31
Finnish TDT	UAS	86.42	89.88	91.66	94.10	**94.39**	94.29
	LAS	82.03	87.46	89.49	92.65	**92.85**	92.70
Hungarian Szeged	UAS	89.68	84.04	88.76	90.44	90.52	**90.76**
	LAS	84.88	79.73	85.12	86.77	87.00	**87.34**
Indonesian GSD	UAS	86.45	85.31	**86.47**	85.41	85.50	85.19
	LAS	80.10	78.99	**80.40**	78.11	78.23	78.39
Japanese GSD	UAS	94.37	95.06	**95.55**	94.24	94.28	94.54
	LAS	92.08	93.73	**94.24**	93.08	93.25	93.57
Kazakh KTB	UAS	**74.77**	53.30	57.02	64.85	63.35	62.95
	LAS	**63.66**	33.38	38.72	46.30	44.70	44.34
Korean GSD	UAS	82.74	87.70	89.38	92.00	92.06	**92.24**
	LAS	74.26	84.24	86.05	89.36	89.42	**89.48**
Turkish IMST	UAS	74.56	74.19	76.30	81.44	82.18	**82.70**
	LAS	67.44	67.56	70.11	75.72	76.19	**76.51**
Uyghur UDT	UAS	65.89	78.46	**79.10**	76.30	78.68	76.45
	LAS	48.80	67.09	**67.46**	64.00	67.44	64.44

attention mechanism through layers of the pre-trained model.

UDPipe 2.0 [27] is an NLP tool that also includes a dependency parser. Except for a few minor differences, its architecture is nearly identical to that of our base parser. It utilizes character-based word representation obtained by bi-directional GRU (gated recurrent units) as only sub-word level representation. They employ three forms of embeddings to represent each input word: pre-trained word embedding, trained word embedding, and character-based word embedding. Straka et al. [26] looked into the impact of utilizing both BERT and Flair word vectors on UDPipe 2.0.

The results show that the sentence piece based word representation has contributed to all experimented languages other than Estonian and Kazakh. Sentence representation has improved parsing performance for Estonian, Hungarian, Japanese, Korean and Turkish. In Indonesian,

sentence representation has boosted the LAS while slightly decreased the UAS. In Finnish, Kazakh and Uyghur, sentence representation has had a little unfavorable affect on the UAS and the LAS. We have achieved higher scores than previously reported in [25, 26] for Estonian, Finnish, Hungarian, Korean, and Turkish.

4. Discussion and Conclusion

In this study, we propose to employ sub-word level sentence piece based word representation and sentence representation that stores the entire meaning of the sentence in order to boost dependency parsing performance. Although the proposed improvements are applicable to all languages, we experiment their influence on a subset of languages; the nine agglutinative languages. We intend to alleviate the challenges of dependency parsing for agglutinative languages due to their unique characteristics such as rich morpho-syntax, flexible word order, and so on.

With the exception of Estonian and Kazakh, sentence piece based token encoding improves parsing performance by capturing morphemes in all languages tested. Despite being an agglutinative language, Estonian borrows about a third of its vocabulary from Germanic languages. We think that this is why sentence piece-based word encoding does not increase parsing accuracy in this language. The obtained result for Kazakh is attributed to a data shortage, because the Kazakh training set has just 31 sentences. Due to a lack of learning data, parsing accuracy diminishes as the number of learned parameters grows with each additional feature. In Estonian, Hungarian, Japanese, Korean, Turkish and partially Indonesian, employing sentence representation as an additional feature improves the parsing accuracy. Because the entire meaning of the sentence contributes to extract syntactic information. We construct our sentence representation by concatenating the latest hidden states of bi-LSTM backward and forward directions, as well as ELECTRA or BERT-based sentence vectors where they are available. However, because there are no publicly accessible ELECTRA or BERT pre-trained LMs for the Kazakh and Uyghur languages, the sentence representations of both of these languages rely only on the final hidden states of backward and forward directions of bi-LSTM. Additionally, training data of these languages are relatively small to fit to provide well-learned sentence representation. As a result, using sentence representation in these languages is ineffective in improving parsing accuracy. For Finnish, we received an unexpected result. Finnish has a large vocabulary because it is a highly morphological rich language. Because of the vast quantity of the vocabulary, pre-trained LM tokenizers of this language mostly granulates the token into word pieces that represent morphemes rather than words. We only used the matching vector of the first word piece per word when fine-tuning BERT or ELECTRA LM, ignoring the remainders. We suspect that sentence vectors loses syntactic information, because of disregarding some word pieces carry syntactic information. This might be why the sentence representation is unable to increase parsing performance for Finnish.

In conclusion, sub-word units and morpho-syntactic features are critical to identifying the syntactic function of the word for agglutinative languages. Sentence piece based word representation contributes to capturing morphemes of the word and enhances parsing accuracy. Furthermore, with a few exceptions, sentence representation that stores the whole meaning of

the sentence increases parsing performance for the majority of languages.

Acknowledgments

We would like to thank Wiseborn M. Danquah and my dear wife Şeyma Altıntaş for their insightful remarks, as well as all of the other anonymous reviewers who took the time and effort to review this research.

References

[1] Y. Zhou, C. Song, J. Li, Z. Wu, H. Meng, Dependency parsing based semantic representation learning with graph neural network for enhancing expressiveness of text-to-speech, arXiv preprint arXiv:2104.06835 (2021).

[2] K. Luo, F. Lin, X. Luo, K. Zhu, Knowledge base question answering via encoding of complex query graphs, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2185–2194.

[3] M. Zhang, Z. Li, G. Fu, M. Zhang, Syntax-enhanced neural machine translation with syntax-aware word representations, arXiv preprint arXiv:1905.02878 (2019).

[4] R. Cai, M. Lapata, Syntax-aware semantic role labeling without parsing, Transactions of the Association for Computational Linguistics 7 (2019) 343–356.

[5] Q. Xia, Z. Li, M. Zhang, M. Zhang, G. Fu, R. Wang, L. Si, Syntax-aware neural semantic role labeling, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 7305–7313.

[6] C. Dos Santos, B. Zadrozny, Learning character-level representations for part-of-speech tagging, in: International Conference on Machine Learning, PMLR, 2014, pp. 1818–1826.

[7] Y. Kim, Y. Jernite, D. Sontag, A. M. Rush, Character-aware neural language models, in: Thirtieth AAAI conference on artificial intelligence, 2016, pp. 2741–2749.

[8] S. Yu, N. Kulkarni, H. Lee, J. Kim, Syllable-level neural language model for agglutinative language, arXiv preprint arXiv:1708.05515 (2017).

[9] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the association for computational linguistics 5 (2017) 135–146.

[10] R. Tsarfaty, D. Seddah, Y. Goldberg, S. Kübler, Y. Versley, M. Candito, J. Foster, I. Rehbein, L. Tounsi, Statistical parsing of morphologically rich languages (spmrl) what, how and whither, in: Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, 2010, pp. 1–12.

[11] G. Eryiğit, K. Oflazer, Statistical dependency parsing of turkish, Sabanci University Research Database (2006).

[12] Ş. B. Özateş, A. Özgür, T. Güngör, B. Öztürk, A hybrid approach to dependency parsing: Combining rules and morphology with deep learning, arXiv preprint arXiv:2002.10116 (2020).

[13] T. Kudo, Subword regularization: Improving neural network translation models with multiple subword candidates, arXiv preprint arXiv:1804.10959 (2018).

[14] T. Dozat, C. D. Manning, Deep biaffine attention for neural dependency parsing, arXiv preprint arXiv:1611.01734 (2016).
[15] T. Kudo, J. Richardson, Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, arXiv preprint arXiv:1808.06226 (2018).
[16] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555 (2020).
[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[18] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
[19] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, S. Pyysalo, Multilingual is not enough: Bert for finnish, arXiv preprint arXiv:1912.07076 (2019).
[20] D. M. Nemeskey, Natural Language Processing Methods for Language Modeling, Ph.D. thesis, Eötvös Loránd University, 2020.
[21] F. Ginter, J. Hajič, J. Luotolahti, M. Straka, D. Zeman, CoNLL 2017 shared task - automatically annotated raw texts and word embeddings, 2017. URL: http://hdl.handle.net/11234/1-1989, LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
[22] K. Kim, Pretrained language models for korean, https://github.com/kiyoungkim1/LMkor, 2020.
[23] W. Che, Y. Liu, Y. Wang, B. Zheng, T. Liu, Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation, in: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 55–64. URL: http://www.aclweb.org/anthology/K18-2005.
[24] I. Loshchilov, F. Hutter, Fixing weight decay regularization in adam, 2018. URL: https://openreview.net/forum?id=rk6qdGgCZ.
[25] D. Kondratyuk, M. Straka, 75 languages, 1 model: Parsing universal dependencies universally, arXiv preprint arXiv:1904.02099 (2019).
[26] M. Straka, J. Straková, J. Hajič, Evaluating contextualized embeddings on 54 languages in pos tagging, lemmatization and dependency parsing, arXiv preprint arXiv:1908.07448 (2019).
[27] M. Straka, Udpipe 2.0 prototype at conll 2018 ud shared task, in: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, 2018, pp. 197–207.