

Dependency Parsing as Sequence Labeling with Head-Based Encoding and Multi-Task Learning

Ophélie Lacroix

Siteimprove

Sankt Annæ Plads 28

DK-1250 Copenhagen, Denmark

ola@siteimprove.com

Abstract

Dependency parsing as sequence labeling has recently proved to be a relevant alternative to the traditional transition- and graph-based approaches. It offers a good trade-off between parsing accuracy and speed. However, recent work on dependency parsing as sequence labeling ignore the pre-processing time of Part-of-Speech tagging – which is required for this task – in the evaluation of speed while other studies showed that Part-of-Speech tags are not essential to achieve state-of-the-art parsing scores. In this paper, we compare the accuracy and speed of *shared* and *stacked* multi-task learning strategies – as well as a strategy that combines both – to learn Part-of-Speech tagging and dependency parsing in a single sequence labeling pipeline. In addition, we propose an alternative encoding of the dependencies as labels which does not use Part-of-Speech tags and improves dependency parsing accuracy for most of the languages we evaluate.

1 Introduction

Traditional dependency parsers are transition based (Kuhlmann et al., 2011) or graph based (McDonald, 2006). In contrast to previous studies, Strzyz et al. (2019) recently showed that dependency parsing reframed as a sequence labeling problem is also a competitive strategy. The idea is, for a given token in a sentence, to encode into a single tag the information about which token is its parent in the dependency tree (and the label of the incoming dependency). These tags can be predicted in a sequence labeling process and then be decoded in order to rebuild the dependency tree. Strzyz et al. (2019) compare the performance of dependency parsing as sequence labeling using several encodings of the dependencies which have been presented in previous work¹ and show that the best encoding leads to state-of-the-art performance.

One of the main arguments for performing dependency parsing as sequence labeling is to achieve a good speed-accuracy tradeoff (leveraging the efficiency of deep learning frameworks running on GPUs). However, the encoding that is reported as the best one in (Strzyz et al., 2019), requires Part-of-Speech (PoS) tags to encode and decode the dependencies. The method thus involves a pre-processing step of PoS-tagging which is not considered in the evaluation of the parsing speed, whereas previous studies (Ballesteros et al., 2015; de Lhoneux et al., 2017a) showed that PoS-tagging is not a requirement for neural transition-based parsers – using word embeddings as input – in order to achieve state-of-the-art performance. In this work, we set up a single pipeline that performs both PoS-tagging and dependency parsing in order to study the performance of several architectures. We compare the *shared* (Søgaard and Goldberg, 2016) and *stacked* (Hashimoto et al., 2017) multi-task learning strategies to a strategy that combines both, with the aim of identifying a proper trade-off between parsing accuracy and speed.

We also present an alternative encoding that does not use PoS-tags to encode the dependencies. It, however, requires an additional step of *head* tagging which consists of predicting which tokens in a sentence are parents of other tokens (i.e., have dependents in the dependency tree). Hence, the following task of dependency parsing consists of predicting to which of these parents the tokens are attached. We use a similar encoding as in Strzyz et al. (2019). This new encoding aims at reducing the complexity

¹Such as: the relative positional encoding (Li et al., 2018; Kiperwasser and Ballesteros, 2018), the relative PoS-based encoding (Spoustová and Spousta, 2010) and the bracketing-based encoding (Yli-Jyrä and Gómez-Rodríguez, 2017).

of the attachment step correcting some of the flaws of the original PoS-based encoding. We finally evaluate whether ablating PoS-tagging in the pipeline using the new encoding affects dependency parsing performance.

Contribution We (i) combine two multi-task learning strategies to set up an efficient pipeline for PoS-tagging and dependency parsing as sequence labeling and (ii) propose a new encoding of the dependencies as labels that does not use PoS-tags.

2 Sequence Labeling Pipeline

We propose to perform several sequence labeling tasks, such as PoS-tagging and dependency parsing, in a neural network pipeline architecture which combines *shared* and *stacked* strategies for multi-task learning.

In the *shared* multi-task learning architecture of Søgaard and Goldberg (2016), several tasks are trained simultaneously through the same layers (they share parameters). A single input is given to the network but it feeds different outputs. While Hashimoto et al. (2017) propose a *stacked* multi-task learning architecture, in which each layer is dedicated to the training of one task and layers are stacked on top of each other in a specific order. The calculated output of the final layer dedicated to a given task is concatenated with the input sequence of the network and then feeds the first layer dedicated to the next task.

In our architecture, we combine the two strategies in order to benefit from the strength of both. We define groups of tasks to train sequentially. In a given group, tasks are trained simultaneously using the *shared* multi-task learning strategy (multiple layers can be stacked for one group). They share the same input and feed different outputs. The outputs of the final layer of each group are concatenated with the input sequence to feed the first layer dedicated to the training of the next group of tasks. We name it the *combined* strategy.

In all strategies, each layer is a bi-LSTM (Graves and Schmidhuber, 2005). The input sequence of the network is a concatenation of word embeddings (pre-trained) and character embeddings (trained using an additional bi-LSTM layer). The outputs are calculated through a Softmax layer.

3 Dependency Encodings

Strzyz et al. (2019) observe that the *relative PoS-based* encoding of the dependencies inspired by Spoustová and Spousta (2010) outperforms other encodings. Given a sentence $w_1 \dots w_n$ and its respective sequence of PoS-tags $p_1 \dots p_n$, an incoming dependency to a token w_j , such as w_i is its parent (i.e., $w_i \rightarrow w_j$), is described as a tuple of:

- the PoS-tag p_i of its parent w_i , and
- the relative position n of p_i to w_j with respect to the PoS-tags of the same value p , i.e., p_i is the n th PoS-tag of value p to the right (if $n > 0$) or to the left (if $n < 0$) of w_j .²

See the RPT tags in Figure 1 as an example of *relative PoS-based* encoding. Note that, in contrast to Strzyz et al. (2019) who predict the relation and the encoding of a dependency as one concatenated tag, in this work, we predict the dependency relations (labels) independently from the dependencies (attachment), as it has been applied to constituent parsing as sequence labelling (vil, 2019). This approach reduces the size of the tagset for each task (label tagging and dependency attachment).

In particular, we identify two flaws with the *PoS-based* encoding:

- the tagset includes many infrequent tags (due to infrequent PoS-tags and long distance dependencies) which are difficult to predict;³

²The root has no actual parent, thus no relative position.

³For instance, on the trainset of the Universal Dependencies (UD) for English (EWT) (Nivre et al., 2018), 90% of the tokens are tagged with the same 15 tags among the 198 encoded tags.

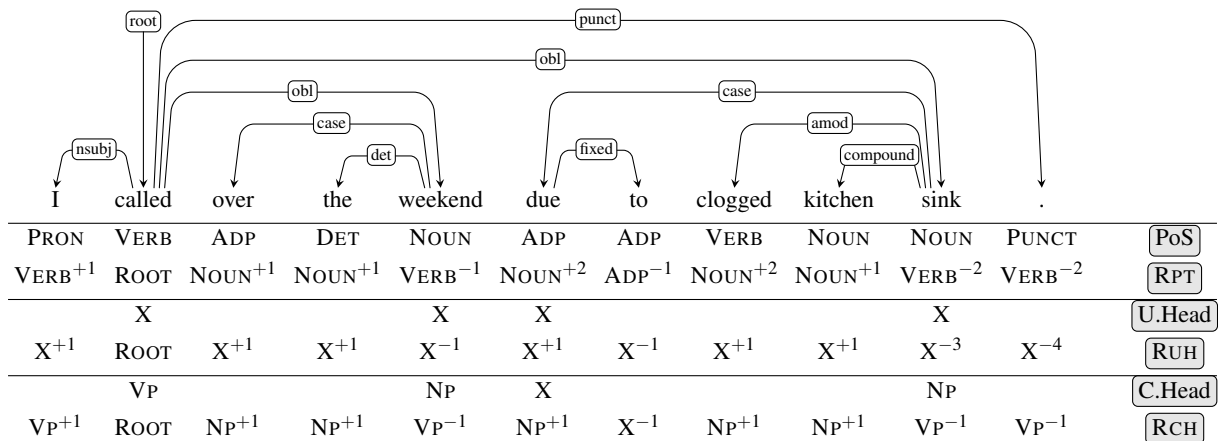


Figure 1: Dependencies and encoded tags on an English sentence from the EWT treebank. RPT is the encoding based on PoS-tags (PoS). RUH and RCH are the relative encodings based on head tags (respectively: unique head –U.Head– tags and chunk head –C.Head– tags).

- consecutive PoS-tags which have similar roles (such as NOUN and PRON or VERB and AUX) make the prediction of the relative position less accurate (i.e., biased towards short relative position) due to the difficulty of identifying which token is the head of a subtree (in a group of tokens which constitute a phrase, e.g., the main noun in a noun phrase or verb in a verb phrase).

In order to alleviate the impact of these flaws, we propose a new encoding strategy that we name *relative head-based* encoding. It requires a first step of *head* tagging in which we identify the heads/parents, i.e., the tokens which have children in the dependency tree. We propose two approaches for tagging the heads:

- a first approach (*Unique Head*) is to tag all parents with a unique tag X (and all non-parents with a NONE tag);
- a second approach (*Chunk Head*) is to see parents as heads of syntactic chunks. We define their roles (tags) as such. In this case, the tagset of the head tagging task includes 5 tags: VP (for heads which are VERBs and AUXs), NP (for NOUNS, PRONs and PRONS), AP (for ADJs and NUMs), X (for the remaining heads) and NONE for the non-parents.⁴

With this approach, disambiguating between PoS-tags with similar roles rests on the head tagging step instead of the actual dependency parsing step which focuses on attaching the children to the correct head.

The *relative head-based* encodings (RUH for *Relative Unique Head* and RCH for *Relative Chunk Head*) are thus deduced from these head tags in the same way as the *relative PoS-based* encoding with the PoS-tags. The encoding of a dependency is defined as a tuple of **the head tag of the parent** and **its relative position** to the child in regards to other head tags with the same role. Hence, the dependency attachment step consists in predicting these encoded tags and then building the dependency tree using in addition the information about the heads from the previous head tagging step.

Using the *relative head-based* encoding reduces the size of the tagset (for the dependency attachment task) by 65% (RUH) and 52% (RCH) on average⁵ compared to the *relative PoS-based* encoding. See an example of the *relative head-based* encodings (RUH and RCH) in Figure 1. In this sentence, “*clogged*” and “*kitchen*” have the same tag with both *head-based* encodings because they have the same head while they have different tags with the *PoS-based* encoding.

⁴The NPs and VPs represent respectively 57% and 27% of the heads of the UD trainsets we train on (see section 4).

⁵On the UD trainsets we train on (see section 4).

PoS-tag based dep. encoding											
Lang.	STR19		Shared			Stacked			Combined		
	UAS	LAS	UAS	LAS	sent/s	UAS	LAS	sent/s	UAS	LAS	sent/s
cs	89.82	87.63	85.36	81.29	163 \pm 1	87.50 [†]	83.66 [†]	100 \pm 2	86.84	82.92	124 \pm 2
en	82.22	78.96	80.33	76.17	159 \pm 2	82.50	78.41	70 \pm 2	81.88	77.87	108 \pm 2
fi	80.31	76.39	77.05	71.37	143 \pm 2	80.80 [†]	75.95 [†]	63 \pm 1	79.85	74.85	97 \pm 1
grc	76.58	71.70	67.98	60.28	157 \pm 4	68.61	61.29	81 \pm 2	68.96	61.41	105 \pm 1
he	67.23	62.86	72.28	65.52	89 \pm 2	77.80 [†]	71.56 [†]	37 \pm 1	75.53	69.27	58 \pm 1
kk	32.14	17.03	42.89	18.88	180 \pm 2	41.27	17.36	82 \pm 2	44.08 [†]	19.36 [†]	127 \pm 3
ta	73.24	66.51	62.89	50.65	113 \pm 6	63.11	51.37	48 \pm 2	63.45	52.29 [†]	76 \pm 2
zh	61.01	57.28	68.28	61.90	92 \pm 1	70.91	64.66	40 \pm 1	71.00	65.00	62 \pm 1
avg	70.32	64.79	69.63	60.76	137	71.56	63.03	65	71.45	62.87	95

Table 1: Dependency parsing scores (+ average sentence per second on CPU) using the PoS-tag based encoding for the different learning strategies (best in bold; [†] marks statistical significance; T-test with $p < 0.05$). **STR19** scores are reported from Strzyz et al. (2019) (besides from Tamil for which they use gold PoS-tags).

4 Experiments

Models We design three types of experiments. In a first set of experiments, we compare the *shared* and *stacked* learning strategies with the *combined* strategy. For each experiment, we train four tasks (simultaneously or sequentially): PoS-tagging, (morphological) feature tagging, label (dependency relation) tagging and dependency attachment. For the *combined* strategy, we define two groups of tasks (trained in the following order): PoS-tagging/feature tagging, followed by label tagging/dependency attachment.

As a second experiment, we compare the performance of the *combined* system using different encodings of the dependencies (*PoS-based* and *head-based*). When using our proposed *head-based* encodings, the groups are (trained in this order): PoS-tagging/feature tagging/head tagging, followed by label tagging/dependency attachment.

Third, we train the pipeline without PoS-tagging and feature tagging (-PoS/feats), using only head tagging as a first group.

Setup We use the pre-trained word embeddings of Grave et al. (2018).⁶ For each task or group of tasks, we use 2 hidden layers of dimension 256. Dimension of the hidden layer for training character embeddings is 128.

Data We use the Universal Dependencies 2.2 (Nivre et al., 2018) dataset for training and evaluating. Following de Lhoneux et al. (2017b), we select a subset of the treebanks: Czech-PDT (cs), English-EWT (en), Finnish-TDT (fi), Ancient Greek-PROIEL (grc), Hebrew-HTB (he), Kazakh-KTB (kk), Tamil-TTB (ta) and Chinese-GSD (zh). Universal PoS-tags (UPOS) are used for PoS-tagging and *PoS-based* encoding. Head tags are deduced from the gold universal dependencies and PoS-tags as stated in Section 3.

Evaluation. We average the scores on 5 runs (with different random seeds) for each experiment. We calculate the unlabeled attachment score (UAS) and the labeled attachment score (LAS) following the guideline of the CoNLL 2018 Shared Task (Zeman et al., 2018).⁷ We also evaluate precision on heads, i.e., percentage of correctly tagged parents.⁸

5 Results

5.1 Multi-task Learning Strategies

We compare the learning strategies (*shared*, *stacked* and *combined* in Table 1) when using the *relative PoS-based* encoding. The *shared* strategy always leads to the lowest scores, which support the idea

⁶For Ancient Greek, we use the embeddings provided for the CoNLL 2017 Shared Task (Ginter et al., 2017).

⁷Only universal dependency labels are evaluated (ignoring language-specific subtypes); punctuation is included.

⁸For the RPT encoding, PoS-tagging on heads only is evaluated.

Lang.	Rel. PoS-Tag based enc.			Rel. Unique Head-based enc.			Relative Chunk Head-based encoding -PoS/feats				
	hd prec.	UAS	LAS	hd prec.	UAS	LAS	hd prec.	UAS	LAS	UAS	LAS
cs	98.16	86.84 [†]	82.92	94.79	86.24	83.11	93.28	86.09	82.31	85.96	82.06
en	92.61	81.88	77.87	93.11	81.48	77.34	90.17	82.70 [†]	78.76 [†]	81.61	77.33
fi	94.64	79.85	74.85	91.37	77.33	72.36	88.25	79.89	75.08	78.43	72.64
grc	84.90	68.96	61.41	88.31	67.61	59.72	78.59	68.71	61.39	67.91	60.44
he	91.85	75.53	69.27	94.98	81.48 [†]	74.12 [†]	88.89	76.93	70.13	77.49	69.97
kk	53.26	44.08	19.36	73.76	47.61 [†]	21.70 [†]	41.23	40.19	18.95	37.30	17.04
ta	83.13	63.45	52.29	79.75	62.13	50.52	76.67	65.48 [†]	54.32 [†]	60.70	49.04
zh	92.55	71.00	65.00	92.67	71.85	65.26	88.67	73.02 [†]	66.82 [†]	71.17	64.34
avg.	86.39	71.45	62.87	88.59	71.97	63.02	80.72	71.63	63.47	70.07	61.61

Table 2: Dependency parsing scores (+ precision on heads) with the different dependency encodings, using the *combined* learning strategy (best in bold; [†] marks statistical significance).

that PoS-tagging and dependency parsing must not be trained simultaneously. On average, the *stacked* learning strategy leads to slightly higher performance (+0.11 UAS/+0.16 LAS) than the *combined* strategy. Both strategies lead to highest performance for half of the languages, but the *stacked* strategy significantly outperforms the *combined* strategy on 3 of the 8 languages while this last strategy gives the (significantly) best scores for 2 languages. However, it is worth noting that the parsing speed is much lower with the *stacked* strategy than with the *combined* strategy, which increases the parsing speed by 48% on average. With comparable scores on average, the *combined* strategy is a good trade-off between speed and accuracy.

5.2 Relative Head-Based Encoding

We compare the scores of the *combined* models using the three different encodings (Table 2). We see that the *relative PoS-based* encoding outperforms the other encodings for only one language (cs: +0.6 UAS) while the RUH encoding is significantly better for 2 languages (he: +5.95/4.85 UAS/LAS; kk: +3.53/2.34) and the RCH for 3 languages (en: +0.82/0.89 UAS/LAS; ta: +2.03/2.03; zh: +2.02/1.82).

Overall, the *relative head-based* encoding is a good approach for parsing as sequence labeling. However, from these results, no clear decision can be made on which tagset for head tagging (RUH vs RCH) would be the most adapted to other languages. The intuition behind the RCH encoding is well-suited to languages which are adapted to a structure in syntactic chunks, such as English (which is reflected in the scores).⁹ It is worth noting that the variation in the scores between the two head-based encodings is substantial and when one is the best option the other often leads to low scores, which shows that the choice of the tagset for the head tagging is crucial and might require fine-tuning for the different languages. For instance, head tagging performs very well on Hebrew and Kazakh¹⁰ using the *Unique head* tagset leading to high parsing scores.

Furthermore, although the *relative head-based* encoding requires an additional step of head tagging (i.e., one more task in the pipeline), the parsing time is equivalent to the RPT encoding since the head tagging task is performed at the same level as PoS-tagging and feature tagging.

In general, long dependencies are especially difficult to predict correctly. While local dependencies (neighbouring child) achieve more than 80% UAS on average, dependencies of length more than 6 do not overcome 50% UAS. We expect the RCH encoding to alleviate the difficulty of the prediction by artificially reducing the distance between the tokens. We analyse the dependency parsing scores in regards to the length of the dependencies. See the comparison between the encodings in Figure 2. Overall, the RCH encoding outperforms the *PoS-based* encoding for all dependency length but the neighbouring children¹¹ while the RUH encoding is especially good on local dependencies but performs poorly on long

⁹Dependency parsing as sequence labeling seems to be more adapted to languages with few non-projective dependencies.

¹⁰Which could be explained by the small training set for Kazakh that makes learning of PoS-tags or chunk head tags more difficult.

¹¹The RCH encoding is particularly damaging on Kazakh: -10.3 UAS on dependencies of length 1 while positive on other

dependencies – the dependency attachment tagset for the RUH encoding includes more rare tags with high relative positions which are then more difficult to predict.

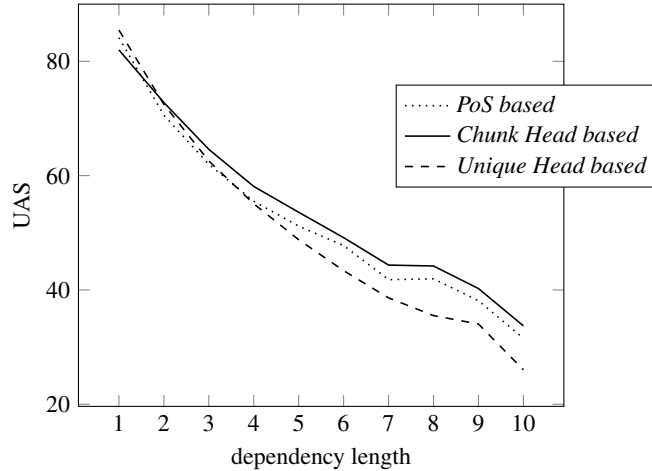


Figure 2: Averaged UAS (on the 8 languages) as a function of the dependency length for the three encodings (using the *combined* learning strategy).

5.3 Ablating PoS-tagging

As previously studied for transition-based parsers (de Lhoneux et al., 2017a; Smith et al., 2018), we want to assess whether dependency parsing as sequence labeling can achieve state-of-the-art performance without PoS-tagging (and feature tagging) as a pre-processing step. We compare the performance of the *combined* strategy using the RCH encoding with and without PoS-tagging (last two columns of Table 2) as part of the first group of tasks to train in the pipeline.

The results are noticeably lower for the ablated model (-1.56 UAS/-1.86 LAS on average) than when using PoS-tagging as an auxiliary task for training head tagging. Determining PoS-tags is essential for most of the languages. Only Hebrew does not suffer from the ablation.

Moreover, it is worth noting that ablating PoS-tagging does not increase the parsing speed since the tasks are performed simultaneously. The *combined* strategy (with PoS-tagging) thus remains a valid trade-off between speed and accuracy.

6 Conclusion

We showed that a combined strategy for multi-task learning using *shared* and *stacked* strategies is on par with a sequential approach while significantly faster at parsing sentences. It provides a good speed-accuracy tradeoff for PoS-tagging and dependency parsing in a single pipeline.

Besides, we proposed a new encoding of the dependencies as labels which does not use PoS-tags. It splits the parsing task in two steps but does not affect negatively the parsing time when performed simultaneously with PoS-tagging. We test two alternatives of this encoding, comparing fine and coarse tagsets for tagging the heads. It shows that the choice of the tagset is crucial: the performance of dependency attachment depends on the performance of head tagging and on how it performs regarding the length of the dependencies. Finally, this suggests that fine-tuning the tagset in regards to properties of the languages could improve overall performance. Globally, the *head-based* models outperform the *PoS-based* model for a majority of the languages.

lengths.

References

- Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved Transition-based Parsing by Modeling Characters instead of Words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- 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. Old School vs. New School: Comparing Transition-Based Parsers with and without Neural Network Enhancement. In *The 15th Treebanks and Linguistic Theories Workshop (TLT)*.
- Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 Shared Task - Automatically Annotated Raw Texts and Word Embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Alex Graves and Jürgen Schmidhuber. 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, pages 5–6.
- Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*.
- Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled Multi-Task Learning: From Syntax to Translation. *Transactions of the Association of Computational Linguistics*, 6:225–240.
- Marco Kuhlmann, Carlos Gómez-Rodríguez, and Giorgio Satta. 2011. Dynamic Programming Algorithms for Transition-Based Dependency Parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*.
- 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.
- Ryan McDonald. 2006. Discriminative Training and Spanning Tree Algorithms for Dependency Parsing. *University of Pennsylvania, PhD Thesis*.
- Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Belato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaz Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskiy,

- Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horňiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaraj, Hanna Nurmi, Stina Ojala, Adédayo Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Ceneil-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roşca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uriu, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 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.
- 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*.
- Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.
- Drahomíra Spoustová and Miroslav Spousta. 2010. Dependency Parsing as a Sequence Labeling Task. *The Prague Bulletin of Mathematical Linguistics*.
- Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable Dependency Parsing as Sequence Labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*.
2019. Better, Faster, Stronger Sequence Tagging Constituent Parsers, author = vilares, david and abdou, mostafa and søgaard, anders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.
- Anssi Yli-Jyrä and Carlos Gómez-Rodríguez. 2017. Generic Axiomatization of Families of Noncrossing Graphs in Dependency Parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for 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*.