Sequence Labeling Parsing by Learning Across Representations

Michalina Strzyz **David Vilares** **Carlos Gómez-Rodríguez**
Universidade da Coruña, CITIC
FASTPARSE Lab, LyS Research Group, Departamento de Computación
Campus de Elviña, s/n, 15071 A Coruña, Spain
{michalina.strzyz,david.vilares,carlos.gomez}@udc.es

Abstract

We use parsing as sequence labeling as a common framework to learn across constituency and dependency syntactic abstractions. To do so, we cast the problem as multitask learning (MTL). First, we show that adding a parsing paradigm as an auxiliary loss consistently improves the performance on the other paradigm. Secondly, we explore an MTL sequence labeling model that parses both representations, at almost no cost in terms of performance and speed. The results across the board show that on average MTL models with auxiliary losses for constituency parsing outperform single-task ones by 1.05 F1 points, and for dependency parsing by 0.62 UAS points.

1 Introduction

Constituency (Chomsky, 1956) and dependency grammars (Mel'cuk, 1988; Kübler et al., 2009) are the two main abstractions for representing the syntactic structure of a given sentence, and each of them has its own particularities (Kahane and Mazziotta, 2015). While in constituency parsing the structure of sentences is abstracted as a phrase-structure tree (see Figure 1a), in dependency parsing the tree encodes binary syntactic relations between pairs of words (see Figure 1b).

When it comes to developing natural language processing (NLP) parsers, these two tasks are usually considered as disjoint tasks, and their improvements therefore have been obtained separately (Charniak, 2000; Nivre, 2003; Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017; Ma et al., 2018; Kitaev and Klein, 2018).

Despite the potential benefits of learning across representations, there have been few attempts in the literature to do this. Klein and Manning (2003) considered a factored model that provides separate methods for phrase-structure and lexical dependency trees and combined them to obtain optimal parses. With a similar aim, Ren et al. (2013) first compute the *n* best constituency trees using a probabilistic context-free grammar, convert those into dependency trees using a dependency model, compute a probability score for each of them, and finally rerank the most plausible trees based on both scores. However, these methods are complex and intended for statistical parsers. Instead, we propose a extremely simple framework to learn across constituency and dependency representations.

Contribution (i) We use sequence labeling for constituency (Gómez-Rodríguez and Vilares, 2018) and dependency parsing (Strzyz et al., 2019) combined with multi-task learning (MTL) (Caruana, 1997) to learn across syntactic representations. To do so, we take a parsing paradigm (constituency or dependency parsing) as an auxiliary task to help train a model for the other parsing representation, a simple technique that translates into consistent improvements across the board. (ii) We also show that a single MTL model following this strategy can robustly produce both constituency and dependency trees, obtaining a performance and speed comparable with previous sequence labeling models for (either) constituency or dependency parsing. The source code is available at https://github.com/mstrise/seq2label-crossrep

2 Parsing as Sequence Labeling

Notation We use $w = [w_i, ..., w_{|w|}]$ to denote an input sentence. We use bold style lower-cased and math style upper-cased characters to refer to vectors and matrices (e.g. $\mathbf{x}$ and $\mathbf{W}$).

Sequence labeling is a structured prediction task where each token in the input sentence is mapped to a label (Rei and Søgaard, 2018). Many NLP tasks suit this setup, including part-of-speech tag-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5350–5357
Florence, Italy, July 28 - August 2, 2019.

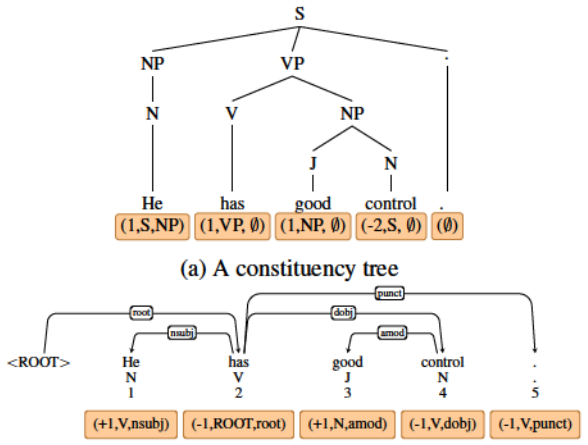
ging, named-entity recognition or chunking (Sang and Buchholz, 2000; Toutanova and Manning, 2000; Tjong Kim Sang and De Meulder, 2003). More recently, syntactic tasks such as constituency parsing and dependency parsing have been successfully reduced to sequence labeling (Spoustová and Spousta, 2010; Li et al., 2018; Gómez-Rodríguez and Vilares, 2018; Strzyz et al., 2019). Such models compute a tree representation of an input sentence using $|w|$ tagging actions.

We will also cast parsing as sequence labeling, to then learn across representations using multi-task learning. Two are the main advantages of this approach: (i) it does not require an explicit parsing algorithm nor explicit parsing structures, and (ii) it massively simplifies joint syntactic modeling. We now describe parsing as sequence labeling and the architecture used in this work.

Constituency parsing as tagging Gómez-Rodríguez and Vilares (2018) define a linearization method $\Phi_{|w|} : T_{c,|w|} \rightarrow L_c^{|w|}$ to transform a phrase-structure tree into a discrete sequence of labels of the same length as the input sentence. Each label $l_i \in L_c$ is a three tuple (n_i, c_i, u_i) where: n_i is an integer that encodes the number of ancestors in the tree shared between a word w_i and its next one w_{i+1} (computed as relative variation with respect to n_{i-1}), c_i is the non-terminal symbol shared at the lowest level in common between said pair of words, and u_i (optional) is a leaf unary chain that connects c_i to w_i. Figure 1a illustrates the encoding with an example.[1]

Dependency parsing as tagging Strzyz et al. (2019) also propose a linearization method $\Pi_{|w|} : T_{d,|w|} \rightarrow L_d^{|w|}$ to transform a dependency tree into a discrete sequence of labels. Each label $r_i \in L_d$ is also represented as a three tuple (o_i, p_i, d_i). If $o_i > 0$, w_i's head is the o_ith closest word with PoS tag p_i to the right of w_i. If $o_i < 0$, the head is the $-o_i$th closest word to the left of w_i that has as a PoS tag p_i. The element d_i represents the syntactic relation between the head and the dependent terms. Figure 1b depictures it with an example.

Tagging with LSTMs We use bidirectional LSTMs (BILSTMs) to train our models (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997). Briefly, let $\text{LSTM}_{\rightarrow}(\mathbf{x})$ be an abstraction of a LSTM that processes the input from left to right, and let $\text{LSTM}_{\leftarrow}(\mathbf{x})$ be another LSTM processing the input in the opposite direction, the output h_i of a BILSTM at a timestep i is computed as: $\text{BILSTM}(\mathbf{x}, i) = \text{LSTM}_{\rightarrow}(\mathbf{x}_{0:i}) \circ \text{LSTM}_{\leftarrow}(\mathbf{x}_{i:|w|})$. Then, h_i is further processed by a feed-forward layer to compute the output label, i.e. $P(y|\mathbf{h}_i) = softmax(\mathbf{W} * \mathbf{h}_i + \mathbf{b})$. To optimize the model, we minimize the categorical cross-entropy loss, i.e. $\mathcal{L} = -\sum log(P(y|\mathbf{h}_i))$. In Appendix A we detail additional hyperpameters of the network. In this work we use NCRFpp (Yang and Zhang, 2018) as our sequence labeling framework.



(a) A constituency tree

(b) A dependency tree

Figure 1: An example of constituency and dependency trees with their encodings.

3 Learning across representations

To learn across representations we cast the problem as multi-task learning. MTL enables learning many tasks jointly, encapsulating them in a single model and leveraging their shared representation (Caruana, 1997; Ruder, 2017). In particular, we will use a hard-sharing architecture: the sentence is first processed by stacked BILSTMs shared across all tasks, with a task-dependent feed-forward network on the top of it, to compute each task's outputs. In particular, to benefit from a specific parsing abstraction we will be using the concept of auxiliary tasks (Plank et al., 2016; Bingel and Søgaard, 2017; Coavoux and Crabbé, 2017), where tasks are learned together with the main task in the MTL setup even if they are not of actual interest by themselves, as they might help to find out hidden patterns in the data and lead to

[1] In this work we do not use the dual encoding by Vilares et al. (2019), which combines the relative encoding with a top-down absolute scale to represent certain relations.

better generalization of the model.[2] For instance, Hershcovich et al. (2018) have shown that semantic parsing benefits from that approach.

The input is the same for both types of parsing and the same number of timesteps are required to compute a tree (equal to the length of the sentence), which simplifies the joint modeling. In this work, we focus on parallel data (we train on the same sentences labeled for both constituency and dependency abstractions). In the future, we plan to explore the idea of exploiting joint training over disjoint treebanks (Barrett et al., 2018).

3.1 Baselines and models

We test different sequence labeling parsers to determine whether there are any benefits in learning across representations. We compare: (i) a single-task model for constituency parsing and another one for dependency parsing, (ii) a multi-task model for constituency parsing (and another for dependency parsing) where each element of the 3-tuple is predicted as a partial label in a separate subtask instead of as a whole, (iii) different MTL models where the partial labels from a specific parsing abstraction are used as auxiliary tasks for the other one, and (iv) an MTL model that learns to produce both abstractions as main tasks.

Single-paradigm, single-task models (S-S) For constituency parsing, we use the single-task model by Gómez-Rodríguez and Vilares (2018). The input is the raw sentence and the output for each token a single label of the form l_i=(n_i, c_i, u_i). For dependency parsing we use the model by Strzyz et al. (2019) to predict a single dependency label of the form r_i=(o_i, p_i, d_i) for each token.

Single-paradigm, multi-task models (S-MTL) For constituency parsing, instead of predicting a single label output of the form (n_i, c_i, u_i), we generate three partial and separate labels n_i, c_i and u_i through three task-dependent feed-forward networks on the top of the stacked BILSTMs. This is similar to Vilares et al. (2019). For dependency parsing, we propose in this work a MTL version too. We observed in preliminary experiments, as shown in Table 1, that casting the problem as 3-task learning led to worse results. Instead, we cast it as a 2-task learning problem, where the first task consists in predicting the head of a word w_i, i.e. predicting the tuple (o_i, p_i), and the second task predicts the type of the relation (d_i). The loss is here computed as $\mathcal{L}$=$\sum_t \mathcal{L}_t$, where $\mathcal{L}_t$ is the partial loss coming from the subtask t.

[2] Auxiliary losses are usually given less importance during the training process.

Model	UAS	LAS
S-S	93.81	91.59
S-MTL(2)	**94.03**	**91.78**
S-MTL(3)	93.66	91.47

Table 1: Comparison of the single-paradigm models for dependency parsing evaluated on the PTB dev set where each label is learned as single, 2- or 3-tasks.

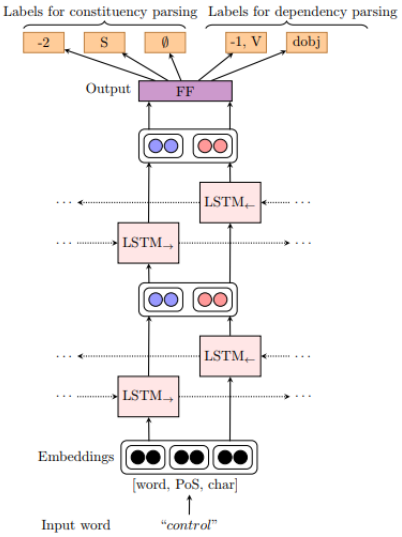


Figure 2: Architecture of our double-paradigm, MTL model with 3-task learning for constituency parsing and 2-task learning for dependency parsing.

Double-paradigm, multi-task models with auxiliary losses (D-MTL-AUX) We predict the partial labels from one of the parsing abstractions as main tasks. The partial labels from the other parsing paradigm are used as auxiliary tasks. The loss is computed as $\mathcal{L}$=$\sum_t \mathcal{L}_t + \sum_a \beta_a \mathcal{L}_a$, where $\mathcal{L}_a$ is an auxiliary loss and β_a its specific weighting factor. Figure 2 shows the architecture used in this and the following multi-paradigm model.

Double paradigm, multi-task models (D-MTL) All tasks are learned as main tasks instead.

4 Experiments

4.1 Data

In the following experiments we use two parallel datasets that provide syntactic analyses for both dependency and constituency parsing.

	Model	Dependency Parsing UAS	Dependency Parsing LAS	Constituency Parsing F1
English (PTB)	S-S	93.60	91.74	90.14
	S-MTL	93.84	91.83	90.32
	D-MTL-AUX	**94.05**	**92.01**	**90.39**
	D-MTL	93.96	91.90	89.81
Basque	S-S	86.20	81.70	89.54
	S-MTL	**87.42**	81.71	90.86
	D-MTL-AUX	87.19	81.73	**91.12**
	D-MTL	87.09	**81.77**	90.76
French	S-S	89.13	85.03	80.68
	S-MTL	**89.54**	84.89	**81.34**
	D-MTL-AUX	89.52	84.97	81.33
	D-MTL	89.45	**85.07**	81.19
German	S-S	91.24	88.76	84.19
	S-MTL	91.54	88.75	**84.46**
	D-MTL-AUX	**91.58**	**88.80**	84.38
	D-MTL	91.45	88.67	84.28
Hebrew	S-S	82.74	75.08	88.85
	S-MTL	83.42	74.91	**91.91**
	D-MTL-AUX	**83.90**	**75.89**	91.83
	D-MTL	82.60	73.73	91.10
Hungarian	S-S	88.24	84.54	90.42
	S-MTL	88.69	84.54	90.76
	D-MTL-AUX	**88.99**	**84.95**	90.69
	D-MTL	88.89	84.89	**90.93**
Korean	S-S	86.47	84.12	83.33
	S-MTL	86.78	84.39	**83.51**
	D-MTL-AUX	**87.00**	**84.60**	83.39
	D-MTL	86.64	84.34	83.08
Polish	S-S	91.17	85.64	92.59
	S-MTL	91.58	85.04	93.17
	D-MTL-AUX	91.37	85.20	93.36
	D-MTL	**92.00**	**85.92**	**93.52**
Swedish	S-S	86.49	80.60	83.81
	S-MTL	87.22	80.61	86.23
	D-MTL-AUX	**87.24**	80.34	**86.53**
	D-MTL	87.15	**80.71**	86.44
average	S-S	88.36	84.13	87.06
	S-MTL	88.89	84.07	88.06
	D-MTL-AUX	**88.98**	**84.28**	**88.11**
	D-MTL	88.80	84.11	87.90

Table 2: Results on the PTB and SPMRL test sets.

Model	Dependency parsing UAS	Dependency parsing LAS	Constituency Parsing F1
Chen and Manning (2014)	91.80	89.60	—
Kiperwasser and Goldberg (2016)	93.90	91.90	—
Dozat and Manning (2017)	95.74	94.08	—
Ma et al. (2018)	95.87	94.19	—
Fernández-G and Gómez-R (2019)	96.04	94.43	—
Vinyals et al. (2015)	—	—	88.30
Zhu et al. (2013)	—	—	90.40
Vilares et al. (2019)	—	—	90.60
Dyer et al. (2016)	—	—	91.20
Kitaev and Klein (2018)	—	—	95.13
D-MTL-AUX	**94.05**	**92.01**	**90.39**

Table 3: Comparison of existing models against the D-MTL-AUX model on the PTB test set.

PTB For the evaluation on English language we use the English Penn Treebank (Marcus et al., 1993), transformed into Stanford dependencies (De Marneffe et al., 2006) with the predicted PoS tags as in Dyer et al. (2016).

SPMRL We also use the SPMRL datasets, a collection of parallel dependency and constituency treebanks for morphologically rich languages (Seddah et al., 2014). In this case, we use the predicted PoS tags provided by the organizers. We observed some differences between the constituency and dependency predicted input features provided with the corpora. For experiments where dependency parsing is the main task, we use the input from the dependency file, and the converse for constituency, for comparability with other work. D-MTL models were trained twice (one for each input), and dependency and constituent scores are reported on the model trained on the corresponding input.

Metrics We use bracketing F-score from the original EVALB and EVAL_SPMRL official scripts to evaluate constituency trees. For dependency parsing, we rely on LAS and UAS scores where punctuation is excluded in order to provide a homogeneous setup for PTB and SPMRL.

4.2 Results

Table 2 compares single-paradigm models against their double-paradigm MTL versions. On average, MTL models with auxiliary losses achieve the best performance for both parsing abstractions. They gain 1.05 F1 points on average in comparison with the single model for constituency parsing, and 0.62 UAS and 0.15 LAS points for dependency parsing. In comparison to the single-paradigm MTL models, the average gain is smaller: 0.05 F1 points for constituency parsing, and 0.09 UAS and 0.21 LAS points for dependency parsing.

MTL models that use auxiliary tasks (D-MTL-AUX) consistently outperform the single-task models (S-S) in all datasets, both for constituency parsing and for dependency parsing in terms of UAS. However, this does not extend to LAS. This different behavior between UAS and LAS seems to be originated by the fact that 2-task dependency parsing models, which are the basis for the corresponding auxiliary task and MTL models, improve UAS but not LAS with respect to single-task dependency parsing models. The reason might be that

Model	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	*average*
Nivre et al. (2007)	70.11	77.98	77.81	69.97	70.15	82.06	75.63	73.21	*74.62*
Ballesteros (2013)	78.58	79.00	82.75	73.01	79.63	82.65	79.89	75.82	*78.92*
Ballesteros et al. (2015) (char+POS)	78.61	81.08	84.49	72.26	76.34	86.21	78.24	74.47	*78.96*
De La Clergerie (2013)	77.55	82.06	84.80	73.63	75.58	81.02	82.56	77.54	*79.34*
Björkelund et al. (2013) (ensemble)	85.14	85.24	89.65	80.89	86.13	86.62	87.07	82.13	*85.36*
D-MTL-AUX	84.02	83.85	88.18	74.94	80.26	85.93	85.86	79.77	*82.85*

Table 4: Dependency parsing: existing models evaluated with LAS scores on the SPMRL test set.

Model	Basque	French	German	Hebrew	Hungarian	Korean	Polish	Swedish	*average*
Fernández-González and Martins (2015)	85.90	78.75	78.66	88.97	88.16	79.28	91.20	82.80	*84.22*
Coavoux and Crabbé (2016)	86.24	79.91	80.15	88.69	90.51	85.10	92.96	81.74	*85.66*
Björkelund et al. (2013) (ensemble)	87.86	81.83	81.27	89.46	91.85	84.27	87.55	83.99	*86.01*
Coavoux and Crabbé (2017)	88.81	82.49	85.34	89.87	92.34	86.04	93.64	84.00	*87.82*
Vilares et al. (2019)	91.18	81.37	84.88	92.03	90.65	84.01	93.93	86.71	*88.10*
Kitaev and Klein (2018)	89.71	84.06	87.69	90.35	92.69	86.59	93.69	84.35	*88.64*
D-MTL-AUX	91.12	81.33	84.38	91.83	90.69	83.39	93.36	86.53	*87.83*

Table 5: Constituency parsing: existing models evaluated with F1 score on the SPMRL test set.

Model	Dependency parsing	Constituency parsing
S-S	$102_{\pm 6}$	$117_{\pm 6}$
S-MTL	$128_{\pm 11}$	$133_{\pm 1}$
D-MTL-AUX	$128_{\pm 11}$	$133_{\pm 1}$
D-MTL	$124_{\pm 1}$	$124_{\pm 1}$

Table 6: Sentences/second on the PTB test set.

the single-task setup excludes unlikely combinations of dependency labels with PoS tags or dependency directions that are not found in the training set, while in the 2-task setup, both components are treated separately, which may be having a negative influence on dependency labeling accuracy.

In general, one can observe different range of gains of the models across languages. In terms of UAS, the differences between single-task and MTL models span between 1.22 (Basque) and -0.14 (Hebrew); for LAS, 0.81 and -1.35 (both for Hebrew); and for F1, 3.06 (Hebrew) and -0.25 (Korean). Since the sequence labeling encoding used for dependency parsing heavily relies on PoS tags, the result for a given language can be dependent on the degree of the granularity of its PoS tags.

In addition, Table 3 provides a comparison of the D-MTL-AUX models for dependency and constituency parsing against existing models on the PTB test set. Tables 4 and 5 shows the results for various existing models on the SPMRL test sets.[3]

Table 6 shows the speeds (sentences/second) on a single core of a CPU[4]. The D-MTL setup comes at almost no added computational cost, so the very good speed-accuracy tradeoff already provided by the single-task models is improved.

5 Conclusion

We have described a framework to leverage the complementary nature of constituency and dependency parsing. It combines multi-task learning, auxiliary tasks, and sequence labeling parsing, so that constituency and dependency parsing can benefit each other through learning across their representations. We have shown that MTL models with auxiliary losses outperform single-task models, and MTL models that treat both constituency and dependency parsing as main tasks obtain strong results, coming almost at no cost in terms of speed. Source code will be released upon acceptance.

Acknowlegments

This work has received funding from the European Research Council (ERC), under the European Union's Horizon 2020 research and innovation programme (FASTPARSE, grant agreement No 714150), from the ANSWER-ASAP project (TIN2017-85160-C2-1-R) from MINECO, and from Xunta de Galicia (ED431B 2017/01).

[3]Note that we provide these SPMRL results for merely informative purposes. While they are the best existing results to our knowledge in these datasets, not all are directly comparable to ours (due to not all of them using the same kinds of information, e.g. some models do not use morphological features). Also, there are not many recent results for dependency parsing on the SPMRL datasets, probably due to the popularity of UD corpora. For comparison, we have included punctuation for this evaluation.

[4]Intel Core i7-7700 CPU 4.2 GHz.

References

Miguel Ballesteros. 2013. Effective morphological feature selection with maltoptimizer at the spmrl 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 63–70.

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*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

Maria Barrett, Joachim Bingel, Nora Hollenstein, Marek Rei, and Anders Søgaard. 2018. Sequence classification with human attention. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 302–312.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. *CoRR*, abs/1702.08303.

Anders Björkelund, Ozlem Cetinoglu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (re) ranking meets morphosyntax: State-of-the-art results from the spmrl 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Association for Computational Linguistics.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Noam Chomsky. 1956. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Maximin Coavoux and Benoit Crabbé. 2016. Neural greedy constituent parsing with dynamic oracles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 172–182.

Maximin Coavoux and Benoît Crabbé. 2017. Multilingual lexicalized constituency parsing with word-level auxiliary tasks. In *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, volume 2, pages 331–336. Association for Computational Linguistics.

Eric De La Clergerie. 2013. Exploring beam-based shift-reduce dependency parsing with dyalog: Results from the spmrl 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 53–62.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Lrec*, volume 6, pages 449–454.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209. Association for Computational Linguistics.

Daniel Fernández-González and Carlos Gómez-Rodríguez. 2019. Left-to-right dependency parsing with pointer networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, page to appear, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Daniel Fernández-González and André F. T. Martins. 2015. Parsing as reduction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1523–1533, Beijing, China. Association for Computational Linguistics.

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1314–1324. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018. Multitask parsing across semantic representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 373–385.

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

Sylvain Kahane and Nicolas Mazziotta. 2015. Syntactic polygraphs. a formalism extending both constituency and dependency. In *Mathematics of Language*.

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.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2676–2686.

Dan Klein and Christopher D Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in neural information processing systems*, pages 3–10.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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)*, volume 1, pages 1403–1414.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Igor Aleksandrovic Mel'cuk. 1988. *Dependency syntax: theory and practice*. SUNY press.

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

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 412.

Marek Rei and Anders Søgaard. 2018. Zero-shot sequence labeling: Transferring knowledge from sentences to tokens. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 293–302.

Xiaona Ren, Xiao Chen, and Chunyu Kit. 2013. Combine constituent and dependency parsing via reranking. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098.

Erik F Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the spmrl 2014 shared task on parsing morphologically-rich languages. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 103–109.

Drahomíra Spoustová and Miroslav Spousta. 2010. Dependency parsing as a sequence labeling task. *The Prague Bulletin of Mathematical Linguistics*, 94(1):7–14.

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: Human Language Technologies, Volume 2 (Short Papers)*, page to appear, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Kristina Toutanova and Christopher D Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

David Vilares, Mostafa Abdou, and Anders Søgaard. 2019. Better, faster, stronger sequence tagging constituent parsers. In *Proceedings of the 2019 Conference of the North American Chapter of the Associ-*

ation for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), page to appear, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in neural information processing systems*, pages 2773–2781.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. *Proceedings of ACL 2018, System Demonstrations*, pages 74–79.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 434–443.

A Model parameters

The models were trained up to 150 iterations and optimized with Stochastic Gradient Descent (SGD) with a batch size of 8. The best model for constituency parsing was chosen with the highest achieved F1 score on the development set during the training and for dependency parsing with the highest LAS score. The best double paradigm, multi-task model was chosen based on the highest harmonic mean among LAS and F1 scores.

Table 7 shows model hyperparameters.

Initial learning rate	0.02
Time-based learning rate decay	0.05
Momentum	0.9
Dropout	0.5
	Dimension
Word embedding	100
Char embedding	30
Self-defined features	20 [5]
Word hidden vector	800
Character hidden vector	50
Type of MTL model	Weighting factor for each task
2-task D	1
3-task C	1
D with auxiliary task C	D: 1 and C: 0.2
C with auxiliary task D	C: 1 and D: 0.1
Multi-task C and D	1

Table 7: Model hyperparameters. D indicates dependency parsing and C constituency parsing.

[5]Models trained on PTB treebank used PoS tag embedding size of 25 in order to assure the same setup for comparison with the previously reported results.