

Concurrent Parsing of Constituency and Dependency

Junru Zhou, Shuailiang Zhang, Hai Zhao*

Department of Computer Science and Engineering

Key Lab of Shanghai Education Commission

for Intelligent Interaction and Cognitive Engineering

MoE Key Lab of Artificial Intelligence, AI Institute

Shanghai Jiao Tong University, Shanghai, China

{zhoujunru, zsl123}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Constituent and dependency representation for syntactic structure share a lot of linguistic and computational characteristics, this paper thus makes the first attempt by introducing a new model that is capable of parsing constituent and dependency at the same time, so that lets either of the parsers enhance each other. Especially, we evaluate the effect of different shared network components and empirically verify that dependency parsing may be much more beneficial from constituent parsing structure. The proposed parser achieves new state-of-the-art performance for both parsing tasks, constituent and dependency on PTB and CTB benchmarks.

1 Introduction

Constituent and dependency are two typical syntactic structure representation forms as shown in Figure 1, which have been well studied from both linguistic and computational perspective (Chomsky, 1981; Bresnan, 2001). In earlier time, linguists and NLP researchers discussed how to encode lexical dependencies in phrase structures, like Tree-adjoining grammar (TAG) (Joshi and Schabes, 1997) and head-driven phrase structure grammar (HPSG) (Pollard and Sag, 1994).

Typical dependency treebanks are usually converted from constituent treebanks, though they may be independently annotated as well for the same languages. Meanwhile, constituent parsing can be accurately converted to dependencies (SD) representation by grammatical rules or machine learning methods (Marneffe et al., 2006; Ma et al., 2010). Such mutual convertibility shows a close

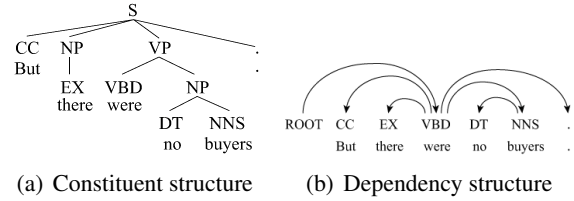


Figure 1: Constituent and dependency structures.

relation between constituent and dependency representation for the same sentence. Thus, it is a natural idea to study the relationship between constituent and dependency structures, and the joint learning of constituent and dependency parsing (Collins, 1997; Charniak, 2000; Klein and Manning, 2004; Charniak and Johnson, 2005; Farkas et al., 2011; Green and Žabokrtský, 2012; Ren et al., 2013; Xu et al., 2014; Yoshikawa et al., 2017; Strzyz et al., 2019).

For further exploit both strengths of the two representation forms for even better parsing, in this work, we propose a new model that is capable of synchronously parsing constituent and dependency.

Multitask learning (MTL) is a natural solution in neural models for multiple inputs and multiple outputs, which is adopted in this work to decode constituent and dependency in a single model. (Søgaard and Goldberg, 2016) indicates that when tasks are sufficiently similar, especially with syntactic nature, MTL would be useful. In contrast to previous work on deep MTL (Collobert et al., 2011; Hashimoto et al., 2017), our model focuses on more related tasks and benefits from the strong inherent relation. At last, our model is evaluated on two benchmark treebanks for both constituent and dependency parsing. The empirical results show that our parser reaches new state-of-the-art for all parsing tasks.

* Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100) and key projects of Natural Science Foundation of China (No. U1836222 and No. 61733011)

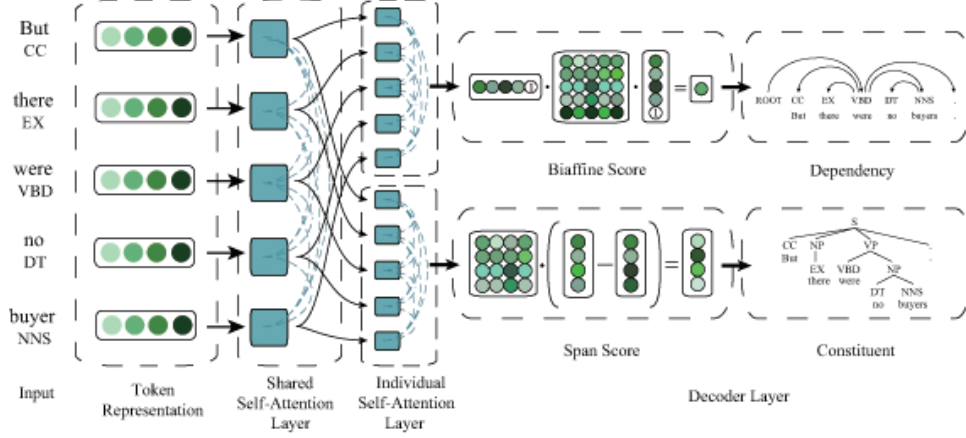


Figure 2: The framework of our joint learning model.

2 Our Model

Using an encoder-decoder backbone, our model may be regarded as an extension of the constituent parsing model of (Kitaev and Klein, 2018a) as shown in Figure 2. The difference is that in our model both constituent and dependency parsing share the same token representation and shared self-attention layers and each has its own individual Self-Attention Layers and subsequent processing layers. Our model includes four modules: token representation, self-attention encoder, constituent and dependency parsing decoder.

2.1 Token Representation

In our model, token representation x_i is composed by character, word and part-of-speech (POS) embeddings. For character-level representation, we explore two types of encoders, CharCNNs (Ma and Hovy, 2016; Chiu and Nichols, 2016) and CharLSTM (Kitaev and Klein, 2018a), as both types have been verified their effectiveness. For word-level representation, we concatenate randomly initialized and pre-trained word embeddings. We consider two ways to compose the final token representation, summing and concatenation, $x_i = x_{char} + x_{word} + x_{POS}$, $x_i = [x_{char}; x_{word}; x_{POS}]$.

2.2 Self-Attention Encoder

The encoder in our model is adapted from (Vaswani et al., 2017) to factor explicit content and position information in the self-attention process (Kitaev and Klein, 2018a). The input matrices $X = [x_1, x_2, \dots, x_n]$ in which x_i is concatenated with position embedding are transformed by a self-attention encoder. We factor the model between content and position information both in

self-attention sub-layer and feed-forward network, whose setting details follow (Kitaev and Klein, 2018a). We also try different numbers of shared self-attention layers in section 3.2.

2.3 Constituent Parsing Decoder

The score $s(T)$ of the constituent parsing tree T is to sum every scores of span (i, j) with label l ,

$$s(T) = \sum_{(i,j,\ell) \in T} s(i, j, \ell).$$

The goal of constituent parser is to find the tree with the highest score: $\hat{T} = \arg \max_T s(T)$. We use CKY-style algorithm to obtain the tree \hat{T} in $O(n^3)$ time complexity (Cocke, John, 1970; Younger, Daniel H., 1975; Kasami, Tadao, 1965; Stern et al., 2017a; Gaddy et al., 2018).

This structured prediction problem is handled with satisfying the margin constraint:

$$s(T^*) \geq s(T) + \Delta(T, T^*),$$

where T^* denote correct parse tree and Δ is the Hamming loss on labeled spans with a slight modification during the dynamic programming search. The objective function is the hinge loss,

$$J_1(\theta) = \max(0, \max_T [s(T) + \Delta(T, T^*)] - s(T^*)).$$

2.4 Dependency Parsing Decoder

Similar to the constituent case, dependency parsing is to search over all possible trees to find the globally highest scoring tree. We follow (Dozat and Manning, 2017) and (Zhang et al., 2017) to predict a distribution over the possible head for each word and find the globally highest scoring tree conditional on the distribution of each word only during testing.

We use the biaffine attention mechanism (Dozat and Manning, 2017) between each word and the candidates of the parent node:

$$\alpha_{ij} = h_i^T W g_j + U^T h_i + V^T g_j + b,$$

where h_i and g_i are calculated by a distinct one-layer perceptron network.

Dependency parser is to minimize the negative log likelihood of the golden tree Y , which is implemented as cross-entropy loss:

$$J_2(\theta) = -(\log P_\theta(h_i|x_i) + \log P_\theta(l_i|x_i, h_i)),$$

where $P_\theta(h_i|x_i)$ is the probability of correct parent node h_i for x_i , and $P_\theta(l_i|x_i, h_i)$ is the probability of the correct dependency label l_i for the child-parent pair (x_i, h_i) .

During parsing, we use the first-order Eisner algorithm (Eisner, 1996) to build projective trees.

2.5 Joint training

Our joint model synchronously predicts the dependency tree and the constituent tree over the same input sentence. The output of the self-attention encoder is sent to the different decoder to generate the different parse tree. Thus, the share components for two parsers include token representation layer and self-attention encoder.

We jointly train the constituent and dependency parser for minimizing the overall loss:

$$J_{model}(\theta) = J_1(\theta) + \lambda J_2(\theta),$$

where λ is a hyper-parameter to control the overall loss. The best performance can be achieved when λ is set to 1.0, which turns out that both sides are equally important.

3 Experiments

We evaluate our model on two benchmark treebanks, English Penn Treebank (PTB) and Chinese Penn Treebank (CTB5.1) following standard data splitting (Zhang and Clark, 2008; Liu and Zhang, 2017b). POS tags are predicted by the Stanford Tagger (Toutanova et al., 2003). For constituent parsing, we use the standard evalb¹ tool to evaluate the F1 score. For dependency parsing, we apply Stanford basic dependencies (SD) representation (Marneffe et al., 2006) converted by the Stanford parser². Following previous work (Dozat and Manning, 2017; Ma et al., 2018), we report the results without punctuations for both treebanks.

3.1 Setup

We use the same experimental settings as (Kiaev and Klein, 2018a). For dependency parsing,

¹<http://nlp.cs.nyu.edu/evalb/>

²<http://nlp.stanford.edu/software/lex-parser.html>

Lexical Representation	F1	UAS	LAS
Word, POS, CharLSTM	93.50	95.41	93.35
Word, POS, CharCNNs	93.46	95.43	93.30
Word, CharLSTM	93.83	95.71	93.68
Word, CharCNNs	93.70	95.43	93.35
POS, CharLSTM	93.16	95.06	92.82
CharLSTM	93.80	95.46	93.46

Table 1: PTB dev set performance on representations.

Joint Component	F1	UAS	LAS
Separate	93.44	94.59	92.15
self-att 0layers	93.76	95.63	93.56
self-att 2layers	93.70	95.48	93.41
Shared self-att 4layers	93.59	95.24	93.12
self-att 6layers	93.68	95.50	93.51
self-att 8layers	93.83	95.71	93.68

Table 2: PTB dev set performance on joint component.

we employ two 1024-dimensional multilayer perceptrons for learning specific representation and a 1024-dimensional parameter matrix for biaffine attention. We use 100D GloVe (Pennington et al., 2014) for English and structured-skipgram (Ling et al., 2015) embeddings for Chinese.

3.2 Ablation Studies

All experiments in this subsection are running from token representation with summing setting.

Token Representation Different token representation combinations are evaluated in Table 1. We find that CharLSTM performs a little better than CharCNNs. Moreover, POS tags on parsing performance show that predicted POS tags decreases parsing accuracy, especially without word information. If POS tags are replaced by word embeddings, the performance increases. Finally, we apply word and CharLSTM as token representation setting for our full model³.

Shared Self-attention Layers As our model providing two outputs from one input, there is a bifurcation setting for how much shared part should be determined. Both constituent and dependency parsers share token representation and 8 self-attention layers at most. Assuming that either parser always takes input information flow through 8 self-attention layers as shown in Figure 2, then the number of shared self-attention layers varying from 0 to 8 may reflect the shared degree in the model. When the number is set to 0, it indicates only token representation is shared for both

³We also evaluate POS tags on CTB which increases parsing accuracy, thus we employ the word, POS tags and CharLSTM as token representation setting for CTB.

Single Model	English		Chinese	
	UAS	LAS	UAS	LAS
Weiss et al. (2015)	94.26	92.41	-	-
Andor et al. (2016)	94.61	92.79	-	-
Dozat and Manning (2017)	95.74	94.08	89.30	88.23
Ma et al. (2018)	95.87	94.19	90.59	89.29
Our model ^{Separate} (Sum)	94.80	92.45	88.66	86.58
Our model ^{Separate} (Concat)	94.68	92.32	88.59	86.17
Our model (Sum)	95.90	93.99	90.89	89.34
Our model (Concat)	95.91	93.86	90.85	89.28
Pre-training				
Wang et al. (2018)(ELMo)	96.35	95.25	-	-
Our model (ELMo)	96.82	94.91	-	-
Our model (BERT)	96.88	95.12	-	-
Ensemble				
Choe and Charniak (2016)	95.9	94.1	-	-
Kuncoro et al. (2017)	95.8	94.6	-	-

Table 3: Dependency parsing on PTB and CTB.

parsers trained for the joint loss through each own 8 self-attention layers. When the number is set to less than 8, for example, 6, then it means that both parsers first shared 6 layers from token representation then have individual 2 self-attention layers.

For different numbers of shared layers, the results are in Table 2. We respectively disable the constituent and the dependency parser to obtain a separate learning setting for both parsers in our model. The comparison in Table 2 indicates that even though without any shared self-attention layers, joint training of our model may significantly outperform separate learning mode. At last, the best performance is still obtained from sharing full 8 self-attention layers.

Besides, comparing UAS and LAS to F1 score, dependency parsing is shown more beneficial from our model which has more than 1% gain in UAS and LAS from parsing constituent together.

3.3 Main Results

Tables 3, 4 and 5 compare our model to existing state-of-the-art, in which indicator *Separate* with our model shows the results of our model learning constituent or dependency parsing separately, (Sum) and (Concat) respectively represent the results with the indicated input token representation setting. On PTB, our model achieves 93.90 F1 score of constituent parsing and 95.91 UAS and 93.86 LAS of dependency parsing. On CTB, our model achieves a new state-of-the-art result on both constituent and dependency parsing. The comparison again suggests that learning jointly in our model is superior to learning sep-

Single Model	LR	LP	F1
Stern et al. (2017a)	93.2	90.3	91.8
Gaddy et al. (2018)	91.76	92.41	92.08
Stern et al. (2017b)	92.57	92.56	92.56
Kitaev and Klein (2018a)	93.20	93.90	93.55
Our model ^{Separate} (Sum)	92.92	93.90	93.41
Our model ^{Separate} (Concat)	93.26	93.95	93.60
Our model (Sum)	93.52	94.00	93.76
Our model (Concat)	93.71	94.09	93.90
Pre-training			
Kitaev and Klein (2018a)(ELMo)	94.85	95.40	95.13
Kitaev and Klein (2018b)(BERT)	95.46	95.73	95.59
Our model (ELMo)	94.73	95.25	94.99
Our model (BERT)	95.51	95.87	95.69
Ensemble			
Liu and Zhang (2017a)	-	-	94.2
Fried et al. (2017)	-	-	94.66
Kitaev and Klein (2018b)	95.51	96.03	95.77

Table 4: Comparison of constituent parsing on PTB.

Single Model	LR	LP	F1
Liu and Zhang (2017b)	85.9	85.2	85.5
Shen et al. (2018)	86.6	86.4	86.5
Fried and Klein (2018)	-	-	87.0
Teng and Zhang (2018)	87.1	87.5	87.3
Kitaev and Klein (2018b)	91.55	91.96	91.75
Our model ^{Separate} (Sum)	91.35	91.65	91.50
Our model ^{Separate} (Concat)	91.36	92.02	91.69
Our model (Sum)	91.79	92.31	92.05
Our model (Concat)	91.41	92.03	91.72

Table 5: Comparison of constituent parsing on CTB.

arately. In addition, we also augment our model with ELMo (Peters et al., 2018) or a larger version of BERT (Devlin et al., 2018) as the sole token representation to compare with other pre-training models. Since BERT is based on sub-word, we only take the last sub-word vector of the word in the last layer of BERT as our sole token representation x_i . Moreover, our single model of BERT achieves competitive performance with other ensemble models.

4 Conclusions

This paper presents a joint model with the constituent and dependency parsing which achieves new state-of-the-art results on both Chinese and English benchmark treebanks. Our ablation studies show that joint learning of both constituent and dependency is indeed superior to separate learning mode. Also, experiments show that dependency parsing is much more beneficial from knowing the constituent structure. Our parser predicts phrase structure and head-word simultaneously which can be regarded as an effective HPSG (Pollard and Sag, 1994) parser.

References

- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. [Globally Normalized Transition-Based Neural Networks](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2442–2452.
- Joan Bresnan. 2001. *Lexical-Functional Syntax*.
- Eugene Charniak. 2000. [A Maximum-Entropy-Inspired Parser](#). In *1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Eugene Charniak and Mark Johnson. 2005. [Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking](#). In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180.
- Jason Chiu and Eric Nichols. 2016. [Named Entity Recognition with Bidirectional LSTM-CNNs](#). *Transactions of the Association for Computational Linguistics (TACL)*, 4:357–370.
- Do Kook Choe and Eugene Charniak. 2016. [Parsing as Language Modeling](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2331–2336.
- N. Chomsky. 1981. *Lectures on government and binding*.
- Cocke, John. 1970. *Programming languages and their compilers :preliminary notes*.
- Michael Collins. 1997. [Three Generative, Lexicalised Models for Statistical Parsing](#). In *35th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, 12(1):2493–2537.
- James Cross and Liang Huang. 2016. [Span-Based Constituency Parsing with a Structure-Label System and Provably Optimal Dynamic Oracles](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–11.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). *CoRR*, abs/1810.04805.
- Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. [Transition-Based Dependency Parsing with Stack Long Short-Term Memory](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL) (Volume 1: Long Papers)*, pages 334–343.
- Jason Eisner. 1996. [Efficient Normal-Form Parsing for Combinatory Categorical Grammar](#). In *34th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Richard Farkas, Bernd Bohnet, and Helmut Schmid. 2011. [Features for Phrase-Structure Reranking from Dependency Parses](#). In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 209–214.
- Daniel Fried and Dan Klein. 2018. [Policy Gradient as a Proxy for Dynamic Oracles in Constituency Parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 469–476.
- Daniel Fried, Mitchell Stern, and Dan Klein. 2017. [Improving Neural Parsing by Disentangling Model Combination and Reranking Effects](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 161–166.
- David Gaddy, Mitchell Stern, and Dan Klein. 2018. [What’s Going On in Neural Constituency Parsers? An Analysis](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (NAACL: HLT)*, pages 999–1010.
- Nathan Green and Zdeněk Žabokrtský. 2012. [Hybrid Combination of Constituency and Dependency Trees into an Ensemble Dependency Parser](#). In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 19–26.
- Kazuma Hashimoto, caiming xiong, Yoshimasa Tsu-ruoka, 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)*, pages 1923–1933.
- Aravind K Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of formal languages*, pages 69–123.
- Kasami, Tadao. 1965. AN EFFICIENT RECOGNITION AND SYNTAXANALYSIS ALGORITHM FOR CONTEXT-FREE LANGUAGES. *Technical Report Air Force Cambridge Research Lab*.

- Nikita Kitaev and Dan Klein. 2018a. [Constituency Parsing with a Self-Attentive Encoder](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2676–2686.
- Nikita Kitaev and Dan Klein. 2018b. Multilingual Constituency Parsing with Self-Attention and Pre-Training. *arXiv preprint arXiv:1812.11760*.
- Dan Klein and Christopher Manning. 2004. [Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency](#). In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. [What Do Recurrent Neural Network Grammars Learn About Syntax?](#) In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers (EACL)*, pages 1249–1258.
- Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. [Two/Too Simple Adaptations of Word2Vec for Syntax Problems](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL: HLT)*, pages 1299–1304.
- Jiangming Liu and Yue Zhang. 2017a. [In-Order Transition-based Constituent Parsing](#). *Transactions of the Association for Computational Linguistics (TACL)*, 5:413–424.
- Jiangming Liu and Yue Zhang. 2017b. [Shift-Reduce Constituent Parsing with Neural Lookahead Features](#). *Transactions of the Association for Computational Linguistics (TACL)*, 5:45–58.
- Xuezhe Ma and Eduard Hovy. 2016. [End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1064–1074.
- Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. [Stack-Pointer Networks for Dependency Parsing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1403–1414.
- Xuezhe Ma, Xiaotian Zhang, Hai Zhao, and Bao-Liang Lu. 2010. [Dependency Parser for Chinese Constituent Parsing](#). In *CIPS-SIGHAN Joint Conference on Chinese Language Processing (CLP)*.
- M. Marneffe, B. Maccartney, and C. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *IN PROC. INTL CONF. ON LANGUAGE RESOURCES AND EVALUATION (LREC)*, pages 449–454.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global Vectors for Word Representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep Contextualized Word Representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologie (NAACL: HLT)*, pages 2227–2237.
- Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.
- Xiaona Ren, Xiao Chen, and Chunyu Kit. 2013. [Combine Constituent and Dependency Parsing via Reranking](#). In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2155–2161.
- Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordani, Aaron Courville, and Yoshua Bengio. 2018. [Straight to the Tree: Constituency Parsing with Neural Syntactic Distance](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1171–1180.
- 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)*, pages 231–235.
- Mitchell Stern, Jacob Andreas, and Dan Klein. 2017a. [A Minimal Span-Based Neural Constituency Parser](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 818–827.
- Mitchell Stern, Daniel Fried, and Dan Klein. 2017b. [Effective Inference for Generative Neural Parsing](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1695–1700.
- Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. [Sequence Labeling Parsing by Learning across Representations](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.
- Zhiyang Teng and Yue Zhang. 2018. [Two Local Models for Neural Constituent Parsing](#). In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 119–132.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. [Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network](#). In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL)*, pages 173–180.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). In *Advances in Neural Information Processing Systems 30(NIPS)*, pages 5998–6008.
- Wenhui Wang, Baobao Chang, and Mairgup Mansur. 2018. [Improved Dependency Parsing using Implicit Word Connections Learned from Unlabeled Data](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pages 2857–2863.
- David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. [Structured Training for Neural Network Transition-Based Parsing](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 323–333.
- Wenduan Xu, Stephen Clark, and Yue Zhang. 2014. [Shift-Reduce CCG Parsing with a Dependency Model](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 218–227.
- Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. [A* CCG Parsing with a Supertag and Dependency Factored Model](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 277–287.
- Younger, Daniel H. 1975. Recognition and parsing of context-free languages in time n^3 . *Information & Control*, 10(2):189–208.
- Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. [Dependency Parsing as Head Selection](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers(EACL)*, pages 665–676.
- Yue Zhang and Stephen Clark. 2008. [A Tale of Two Parsers: Investigating and Combining Graph-based and Transition-based Dependency Parsing](#). In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 562–571.