Grammar Induction with Neural Language Models: An Unusual Replication

Phu Mon Htut[1] pmh330@nyu.edu
Kyunghyun Cho[1,2] kyunghyun.cho@nyu.edu
Samuel R. Bowman[1,2,3] bowman@nyu.edu

[1]Center for Data Science, New York University, 60 Fifth Avenue, New York, NY 10011
[2]Dept. of Computer Science, New York University, 60 Fifth Avenue, New York, NY 10011
[3]Dept. of Linguistics, New York University, 10 Washington Place, New York, NY 10003

1 Introduction

Grammar induction is the task of learning syntactic structure without the expert-labeled treebanks (Charniak and Carroll, 1992; Klein and Manning, 2002). Recent work on *latent tree learning* offers a new family of approaches to this problem by inducing syntactic structure using the supervision from a downstream NLP task (Yogatama et al., 2017; Maillard et al., 2017; Choi et al., 2018). In a recent paper published at ICLR, Shen et al. (2018) introduce such a model and report near state-of-the-art results on the target task of language modeling, and the first strong latent tree learning result on constituency parsing. During the analysis of this model, we discover issues that make the original results hard to trust, including tuning and even training on what is effectively the test set. Here, we analyze the model under different configurations to understand what it learns and to identify the conditions under which it succeeds. We find that this model represents the first empirical success for neural network latent tree learning, and that neural language modeling warrants further study as a setting for grammar induction.

2 Background and Experiments

We analyze the **Parsing-Reading-Predict-Network** (PRPN; Shen et al., 2018), which uses convolutional networks with a form of structured attention (Kim et al., 2017) rather than recursive neural networks (Goller and Kuchler, 1996; Socher et al., 2011) to learn trees while performing straightforward backpropagation training on a language modeling objective. The structure of the model seems rather suboptimal: Since the parser is trained as part of a language model, it makes parsing greedily, with *no* access to any words to the right of the point where each parsing decision must be made.

The experiments on language modeling and parsing are carried out using different configurations of the model—PRPN-LM tuned for language modeling, and PRPN-UP for (unsupervised) parsing. PRPN-LM is much larger than PRPN-UP, with embedding layer that is 4 times larger and the number of units per layer that is 3 times larger. In the PRPN-UP experiments, we observe that the WSJ data is not split, such that the test data is used without parse information for training. This implies that the parsing results of PRPN-UP may not be generalizable in the way usually expected of machine learning evaluation results.

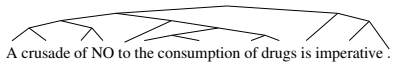

Figure 1: Parses from PRPN-LM trained on AllNLI.

We train PRPN on sentences from two datasets: The full WSJ and AllNLI, the concatenation of SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018b). We then evaluate the constituency trees produced by these models on the full WSJ, WSJ10[1], and the MultiNLI development set.

3 Results

Table 1 shows results with all the models under study, plus several baselines, on WSJ and WSJ10. Unexpectedly, the PRPN-LM models achieve *higher* parsing performance than PRPN-UP. This shows that any tuning done to separate PRPN-UP from PRPN-LM was not necessary, and that the results described in the paper can be largely reproduced by a unified model in a fair setting. Moreover, the PRPN models trained on the larger, out-of-domain AllNLI perform better than those trained on WSJ. Surprisingly, PRPN-LM tained on out-of-domain AllNLI achieves the best F1 score on full WSJ among all the models

[1]A processed subset of WSJ in which the sentences contain no punctuation and no more than 10 words.

Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 371–373
Brussels, Belgium, November 1, 2018.

Model	Training Data	Stopping Criterion	Vocab Size	Parsing F1 WSJ10 μ (σ)	WSJ10 max	WSJ μ (σ)	WSJ max	Depth WSJ	Accuracy on WSJ by Tag ADJP	NP	PP	INTJ
PRPN-UP	AllNLI Train	UP	76k	67.5 (0.6)	68.6	38.1 (0.7)	39.1	5.9	27.8	63.0	31.4	52.9
PRPN-UP	AllNLI Train	LM	76k	66.3 (0.8)	68.5	39.8 (0.6)	40.7	5.9	26.5	53.0	32.9	52.9
PRPN-LM	AllNLI Train	LM	76k	52.4 (4.9)	58.1	42.5 (0.7)	**43.6**	6.2	**34.2**	60.1	**60.0**	**64.7**
PRPN-UP	WSJ Full	UP	15.8k	64.7 (3.2)	70.9	26.6 (1.9)	31.6	5.9	19.3	48.7	19.2	44.1
PRPN-UP	WSJ Full	LM	15.8k	64.3 (3.3)	70.8	26.5 (1.9)	31.4	5.9	18.8	48.1	19.1	44.1
PRPN-UP	WSJ Train	UP	15.8k	63.5 (3.5)	70.7	26.6 (2.5)	34.2	5.9	21.3	57.2	19.4	47.1
PRPN-UP	WSJ Train	LM	15.8k	62.2 (3.9)	70.3	26.4 (2.5)	34.0	5.9	22.3	56.2	19.1	44.1
PRPN-LM	WSJ Train	LM	10k	70.5 (0.4)	71.3	38.3 (0.3)	38.9	5.9	26.0	**64.4**	25.5	50.0
PRPN-LM	WSJ Train	UP	10k	66.1 (0.5)	67.2	34.0 (0.9)	36.3	5.9	32.0	58.3	19.6	44.1
300D ST-Gumbel	AllNLI Train	NLI	–	–	–	*19.0 (1.0)*	*20.1*	–	*15.6*	*18.8*	*9.9*	59.4
w/o Leaf GRU	AllNLI Train	NLI	–	–	–	22.8 (1.6)	25.0	–	18.9	24.1	*14.2*	51.8
300D RL-SPINN	AllNLI Train	NLI	–	–	–	*13.2 (0.0)*	*13.2*	–	*1.7*	*10.8*	*4.6*	50.6
w/o Leaf GRU	AllNLI Train	NLI	–	–	–	*13.1 (0.1)*	*13.2*	–	*1.6*	*10.9*	*4.6*	50.0
CCM	WSJ10 Train	–	–	–	71.9	–	–	–	–	–	–	–
DMV+CCM	WSJ10 Train	–	–	–	77.6	–	–	–	–	–	–	–
UML-DOP	WSJ10 Train	–	–	–	**82.9**	–	–	–	–	–	–	–
Random Trees	–	–	–	–	34.7	21.3 (0.0)	21.4	5.3	17.4	22.3	16.0	40.4
Balanced Trees	–	–	–	–	–	21.3 (0.0)	21.3	4.6	22.1	*20.2*	*9.3*	55.9

Table 1: Unlabeled parsing F1 test results broken down by training data and by early stopping criterion. The *Accuracy* columns represent the fraction of ground truth constituents of a given type that correspond to constituents in the model parses. Italics mark results that are worse than the random baseline. Results with RL-SPINN and ST-Gumbel are from Williams et al. (2018a). WSJ10 baselines are from Klein and Manning (2002, CCM), Klein and Manning (2005, DMV+CCM), and Bod (2006, UML-DOP).

Model	Stopping Criterion	F1 wrt. LB	RB	SP	Depth
300D SPINN	NLI	19.3	36.9	70.2	6.2
w/o Leaf GRU	NLI	21.2	39.0	63.5	6.4
300D SPINN-NC	NLI	19.2	36.2	70.5	6.1
w/o Leaf GRU	NLI	20.6	38.9	64.1	6.3
300D ST-Gumbel	NLI	32.6	37.5	23.7	4.1
w/o Leaf GRU	NLI	30.8	35.6	27.5	4.6
300D RL-SPINN	NLI	95.0	13.5	18.8	8.6
w/o Leaf GRU	NLI	99.1	10.7	18.1	8.6
PRPN-LM	LM	25.6	26.9	45.7	4.9
PRPN-UP	UP	19.4	41.0	46.3	4.9
PRPN-UP	LM	19.9	37.4	**48.6**	4.9
Random Trees	–	27.9	28.0	27.0	4.4
Balanced Trees	–	21.7	36.8	21.3	3.9

Table 2: Unlabeled parsing F1 on the MultiNLI development set for models trained on AllNLI. *F1 wrt.* shows F1 with respect to strictly right- and left-branching (LB/RB) trees and with respect to the Stanford Parser (SP) trees supplied with the corpus; The evaluations of SPINN, RL-SPINN, and ST-Gumbel are from Williams et al. (2018a). SPINN is a supervised parsing model, and the others are latent tree models.

we experimented, even though its performance on WSJ10 is the lowest of all. Under all the configurations we tested, PRPN yields much better performance than that seen with the baselines from Yogatama et al. (2017, called RL-SPINN) and Choi et al. (2018, called ST-Gumbel), despite the fact that the model was tuned exclusively for WSJ10 parsing (Table 1 and 2). This suggests that PRPN is strikingly effective at latent tree learning.

Additionally, Table 2 shows that both PRPN-UP models achieve F1 scores of 46.3 and 48.6 respectively on the MultiNLI dev set, setting the state of the art in parsing on this dataset among latent tree models. We conclude that PRPN does acquire some substantial knowledge of syntax, and that this knowledge agrees with Penn Treebank (PTB) grammar significantly better than chance.

Moreover, we replicate the language modeling perplexity of 61.6 reported in the paper using PRPN-LM trained on WSJ, which indicates that PRPN-LM is effective at both parsing and language modeling.

4 Conclusion

In our analysis of the PRPN model, we find several experimental problems that make the results difficult to interpret. However, in the analyses going well beyond the scope of the original paper, we find that PRPN is nonetheless robust. It represents a viable method for grammar induction and the first success for latent tree learning. We expect that it heralds further work on language modeling as a tool for grammar induction research.

References

Rens Bod. 2006. An All-Subtrees Approach to Unsupervised Parsing. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 865–872.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Eugene Charniak and Glen Carroll. 1992. Two experiments on learning probabilistic dependency grammars from corpora. In *Proceedings of the AAAI Workshop on Statistically-Based NLP Techniques*, page 113.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the Thirty-Second Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI-18)*, volume 2.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 128.

Dan Klein and Christopher D. Manning. 2005. Natural language grammar induction with a generative constituent-context model. *Pattern Recognition*, 38(9):1407–1419.

Jean Maillard, Stephen Clark, and Dani Yogatama. 2017. Jointly learning sentence embeddings and syntax with unsupervised Tree-LSTMs. arXiv preprint 1705.09189.

Yikang Shen, Zhouhan Lin, Chin wei Huang, and Aaron Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *International Conference on Learning Representations*.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Ng, and Chris Manning. 2011. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 129–136.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018a. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics (TACL)*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018b. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to Compose Words into Setences with Reinforcement Learning. *Proceedings of the International Conference on Learning Representations*, pages 1–17.