On the Role of Supervision in Unsupervised Constituency Parsing

Haoyue Shi  Karen Livescu  Kevin Gimpel
Toyota Technological Institute at Chicago, IL, USA, 60637
{freda,klivescu,kgimpel}@ttic.edu

Abstract

We analyze several recent unsupervised constituency parsing models, which are tuned with respect to the parsing $F_1$ score on the Wall Street Journal (WSJ) development set (1,700 sentences). We introduce strong baselines for them, by training an existing supervised parsing model (Kitaev and Klein, 2018) on the same labeled examples they access. When training on the 1,700 examples, or even when using only 50 examples for training and 5 for development, such a few-shot parsing approach can outperform all the unsupervised parsing methods by a significant margin. Few-shot parsing can be further improved by a simple data augmentation method and self-training. This suggests that, in order to arrive at fair conclusions, we should carefully consider the amount of labeled data used for model development. We propose two protocols for future work on unsupervised parsing: (i) use fully unsupervised criteria for hyperparameter tuning and model selection; (ii) use as few labeled examples as possible for model development, and compare to few-shot parsing trained on the same labeled examples.  

1 Introduction

Recent work has considered neural unsupervised constituency parsing (Shen et al., 2018a; Drozdov et al., 2019; Kim et al., 2019b, inter alia), showing that it can achieve much better performance than trivial baselines. However, many of these approaches use the gold parse trees of all sentences in a development set for either early stopping (Shen et al., 2018a, 2019; Drozdov et al., 2019, inter alia) or hyperparameter tuning (Kim et al., 2019a). In contrast, models trained and tuned without any labeled data (Kim et al., 2019b; Peng et al., 2019) are much less competitive.

Are the labeled examples important in order to obtain decent unsupervised parsing performance? How well can we do if we train on these labeled examples rather than merely using them for tuning? In this work, we consider training a supervised constituency parsing model (Kitaev and Klein, 2018) with very few examples as a strong baseline for unsupervised parsing tuned on labeled examples.

We empirically characterize unsupervised and few-shot parsing across the spectrum of labeled data availability, finding that (i) tuning based on a few (as few as 15) labeled examples is sufficient to improve unsupervised parsers over fully unsupervised criteria by a significant margin; (ii) unsupervised parsing with supervised tuning does outperform few-shot parsing with fewer than 15 labeled examples, but few-shot parsing quickly dominates once there are more than 55 examples; and (iii) when few-shot parsing is combined with a simple data augmentation method and self-training (Steedman et al., 2003; Reichart and Rappoport, 2007; McClosky et al., 2006, inter alia), only 15 examples are needed for few-shot parsing to begin to dominate.

Based on these results, we propose the following two protocols for future work on unsupervised parsing:

1. Derive and use fully unsupervised criteria for hyperparameter tuning and model selection.
2. Use as few labeled examples as possible for model development and tuning, and compare to few-shot parsing models trained on the used examples as a strong baseline.

We suggest future work to tune and compare models under each protocol separately.

In addition, we present two side findings on unsupervised parsing: (i) the vocabulary size in unsupervised parsing, which has not been widely considered as a hyperparameter and varies across
prior work, greatly affects the performance of all unsupervised parsing models tested; and (ii) self-training can help improve all investigated unsupervised parsing (Shen et al., 2018a, 2019; Drozdov et al., 2019; Kim et al., 2019a) and few-shot parsing models, and thus can be considered as a post-processing step in future work.

2 Related Work

Unsupervised parsing. During the past two decades, there has been a lot of work on both unsupervised constituency parsing (Klein and Manning, 2002, 2004; Bod, 2006a,b; Segner, 2007; Snyder et al., 2009, inter alia) and unsupervised dependency parsing (Klein and Manning, 2004; Smith and Eisner, 2006; Spitkovsky et al., 2011, 2013, inter alia). Recent work has proposed several effective models for unsupervised or distantly supervised constituency parsing, optimizing either a language modeling objective (Shen et al., 2018a, 2019; Kim et al., 2019b,a, inter alia) or other downstream semantic objectives (Li et al., 2019; Shi et al., 2019). Some of them are tuned with labeled examples in the WSJ development set (Shen et al., 2018a, 2019; Htut et al., 2018; Drozdov et al., 2019; Kim et al., 2019a; Wang et al., 2019) or other labeled examples (Jin et al., 2018, 2019).

Data augmentation. Data augmentation is a strategy for automatically increasing the amount and variety of data for training models, without actually collecting any new data. Such methods have been found helpful on many NLP tasks, including text classification (Kobayashi, 2018; Samanta et al., 2019), relation classification (Xu et al., 2016), and part-of-speech tagging (Sahin and Steedman, 2018). Part of our approach also falls into the category of data augmentation, applied specifically to constituency parsing from very few examples.

Few-shot parsing. Sagae et al. (2008) show that a supervised dependency parsing model trained on 100 examples can work surprisingly well. Recent work has demonstrated the potential of few-shot dependency parsing on multiple languages (Aufrant et al., 2018; Meechan-Maddon and Nivre, 2019; Vania et al., 2019, inter alia). Our approach (§3) can be viewed as few-shot constituency parsing.

3 Few-Shot Constituency Parsing

3.1 Parsing Model

The Benepar parsing model consists of (i) word embeddings, (ii) transformer–based (Vaswani et al., 2017) word-span embeddings, and (iii) a multi-layer perceptron to compute a score for each labeled span. The score of an arbitrary tree is defined as the sum of all of its internal span scores. Given a sentence and its ground-truth parse tree $T^*$, the model is trained to satisfy $\text{score}(T^*) \geq \text{score}(T) + \Delta(T^*, T)$ for any tree $T$ ($T \neq T^*$), where $\Delta$ denotes the Hamming loss on labeled spans. The label-aware CKY algorithm is used to obtain the tree with the highest score. More details can be found in Kitaev and Klein (2018).

3.2 Data Augmentation

We introduce a data augmentation method, subtree substitution (SUB; Figure 1), to automatically improve the diversity of data in the few-shot setting.

We start with a set of sentences with $N$ unlabeled parse trees $S = \{\langle s_i, T_i \rangle\}_{i=1}^N$: $s_i = \langle w_{i1}, w_{i2}, \ldots, w_{iL_i} \rangle$ denotes a sentence with $L_i$ words, where $w_{ijk}$ denotes a word; $T_i = \{b_{ij}, e_{ij}\}_{j=1}^{C_i}$ denotes the unlabeled parse tree of $s_i$ with $C_i$ nonterminal nodes; $b_{ij}$ and $e_{ij}$ denotes the beginning and ending index of a constituent.

The augmented dataset $S'$ is initialized to $S$. At each step, we draw a sentence $s_i$ and its parse tree $T_i$ uniformly from $S'$, and draw a constituent $\langle b_{ij}, e_{ij} \rangle \in T_i$ uniformly from $T_i$. After that, we replace $\langle b_{ij}, e_{ij} \rangle$ with a random $\langle b_{kh}, e_{kh} \rangle \in T_k$; that is, we replace a constituent with another one from the training set. We let $s'_i$ and $T'_i$ denote modified sentence and its parse tree, assign $S' \leftarrow S' \cup \{\langle s'_i, T'_i \rangle\}$, and repeat the above procedure until $S'$ reaches the desired size.

3.3 Self-Training

Steedman et al. (2003), Reichart and Rappoport (2007) and McClosky et al. (2006) have shown that...

---

2In this work, there are only two labels: (i) NT denotes a constituent and (ii) $\emptyset$ denotes non-constituent. The label $\emptyset$ enables the parser output non-binary trees; details can be found in Kitaev and Klein (2018). Almost all existing unsupervised parsing models do not use the nonterminal categories in the development set, so we propose to train such unlabeled constituency parsing models as their baselines.
self-training (ST) on unseen sentences can improve a parsing model. Inspired by this, we apply an iterative self-training strategy after obtaining each supervised or unsupervised parsing model.

Concretely, we start with an arbitrary parsing model $M_0$. At the $i$th step of self-training, we (i) use the trained model from the previous step (i.e., $M_{i-1}$) to predict parse trees for sentences in the WSJ training set and those in the WSJ development set, and (ii) train a supervised parsing model $M_i$ (Kitaev and Klein, 2018) to fit the prediction of $M_{i-1}$. No gold labels are used in self-training.

4 Experiments

4.1 Dataset and Training Details

We use the WSJ portion of the Penn Treebank corpus (Marcus et al., 1993) to train and evaluate the models, replace all number tokens with a special token, and split standard train/dev/test sets following Kim et al. (2019b). For each criterion, we tune the hyperparameters of each model with respect to its performance on the development set. To solve the problem of vocabulary sparsity in the few-shot parsing setting (§3), we initialize the word embeddings of Benepar (Kitaev and Klein, 2018) with the word embeddings from an LSTM–based (Hochreiter and Schmidhuber, 1997) language model trained on the WSJ training set. During training, models are able to access all sentences (without parse trees) in the WSJ training set; for few-shot parsing or unsupervised parsing with supervised tuning, some unlabeled parse trees in the WSJ development set are available as well. We augment the training set to 10,000 examples for few-shot parsing with SUB, and apply 5-step self-training when applicable.

We evaluate the unlabeled $F_1$ score of all models using evalb, discarding punctuation. More details can be found in the supplementary material.

4.2 Models and Tuning Criteria

We investigate four recently proposed models: PRPN (Shen et al., 2018a), ON-LSTM (Shen et al., 2019), DIORA (Drozdov et al., 2019), and Compound PCFG (Kim et al., 2019a).

PRPN and ON-LSTM are left-to-right neural language models, where syntactic distance (Shen et al., 2018b) between consecutive words is computed from the model output and used to infer the constituency parse tree. DIORA learns text-span representations and span-level scores by optimizing a masked language modeling objective. The compound PCFG uses a neural parameterization of a PCFG, as well as a per-sentence latent vector which introduces context sensitivity. Both DIORA and the Compound PCFG use the CKY algorithm to infer the parse tree of a given sentence.

As fully unsupervised tuning criteria, we use perplexity on the development set for PRPN and ON-LSTM, and the upper bound of perplexity for the Compound PCFG, following Shen et al. (2018a, 2019) and Kim et al. (2019a) respectively. For DIORA, we use its reconstruction loss on the development set.5

4.3 Comparison between Unsupervised Parsing and Few-Shot Parsing

We compare unsupervised parsing against few-shot parsing (Table 1 and Figure 2): when there are 55 or more labeled examples available, few-shot
We notice that the result of the Compound PCFG in Table 1 is much worse than that reported by Kim et al. (2019a). The only major difference between their approach and ours is the vocabulary size: instead of keeping all words, they keep the most frequent 10K words in the WSJ corpus and replace others with a special token. To analyze the importance of this choice, we compare the performance of the models with vocabulary size 35K vs. 10K (Figure 2), tuning models separately in the two settings. We find that the vocabulary size, which has not been widely considered a hyperparameter and varies across prior work, greatly affects the performance of all models tested. One possible reason is that a large portion (79.9%) of the low-frequency (i.e., outside the 10K vocabulary) word tokens are nouns or adjectives – some models (e.g., PRPN and Compound PCFG) may benefit from collapsing these tokens to a single form, as it may be a beneficial kind of word clustering. This suggests that we should consider tuning the vocabulary size.

Table 2: Unlabeled $F_1$ scores on the standard WSJ test set. WSJ_train denotes models trained with the 40K sentences in the WSJ training set, and + WSJ_dev SUB denotes models trained with the union of WSJ training sentences and 10K sentences augmented from 1,700 WSJ development sentences. The best number in each row is bolded.

| Model              | WSJ_train + WSJ_dev, SUB |
|--------------------|--------------------------|
| PRPN               | 44.9                     |
| DIORA              | 48.0                     |
| Compound PCFG      | 39.2                     |
| ON-LSTM            | 52.0                     |

Figure 2: Performance of models with vocabulary size 35K (left) and 10K (right) on WSJ Section 24. C-PCFG denotes the Compound PCFG. The $F_1$ scores are averaged over 5 runs with the same hyperparameters, different random seeds, and different sets of labeled examples when applicable.
Table 3: $F_1$ score on WSJ Section 24 of different models, where the base models are those used to report results in Table 1 with $|D_{\text{label}}| = 15$.

| Model       | #ST-steps |
|-------------|-----------|
|             | 0         | 1         | 5         |
| PRPN        | 44.7      | 44.7      | 45.1      |
| Diora       | 46.7      | 48.7      | 49.1      |
| Compound PCFG | 41.1      | 41.8      | 42.2      |
| ON-LSTM     | 50.2      | 51.3      | 52.1      |
| Few-Shot    | 44.3      | 44.5      | 45.0      |
| Few-Shot + SUB | 53.3      | 55.5      | 56.6      |

as a hyperparameter, or fix the vocabulary size for fair comparison in future work.

4.5 Self-Training Improves all Models

Inspired by the fact that self-training boosts the performance of few-shot parsing (Table 1), we apply iterative self-training to the unsupervised parsing models as well, and find that it improves all models (Table 3). It is worth noting that 5-step self-training is better than 1-step self-training for all base models we experimented with. Our results suggest that iterative (e.g., 5-step) self-training may be considered as a standard post-hoc processing step for unsupervised parsing.

5 Discussion

While many state-of-the-art unsupervised parsing models are tuned on all labeled examples in a development set (Drozdov et al., 2019; Kim et al., 2019b; Wang et al., 2019, inter alia), we have demonstrated that, given the same data, few-shot parsing with simple data augmentation and self-training can consistently outperform all of these models by a large margin. We suggest that one possibility for future work is to focus on fully unsupervised criteria, such as language model perplexity (Shen et al., 2018a, 2019; Kim et al., 2019b; Peng et al., 2019; Li et al., 2020) and model stability across different random seeds (Shi et al., 2019), for model selection, as discussed in unsupervised learning work (Smith and Eisner, 2005, 2006; Spitkovsky et al., 2010a,b, inter alia). An alternative is to use as few labeled examples in the development set as possible, and compare to few-shot parsing trained on the used examples as a strong baseline. In addition, we find that self-training is a useful post-processing step for unsupervised parsing.

Our work does not necessarily imply that unsupervised parsers produce poor parses; they may be producing good parses that clash with the conventions of treebanks (Klein, 2005). If this is the case, then extrinsic evaluation of parsers in downstream tasks (Shi et al., 2018), e.g., machine translation (DeNero and Uszkoreit, 2011; Neubig et al., 2012; Gimpel and Smith, 2014), may better show the potential of unsupervised methods.

Acknowledgments

We thank Allyson Ettinger, Yoon Kim, Jiayuan Mao, Shane Settle, and Shubham Toshniwal for helpful discussions, as well as all the knowledgeable anonymous reviewers for their valuable and insightful feedback.

References

Lauriane Aufrant, Guillaume Wisniewski, and François Yvon. 2018. Quantifying training challenges of dependency parsers. In Proceedings of the 27th International Conference on Computational Linguistics, pages 3191–3202, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Rens Bod. 2006a. An all-subtrees approach to unsupervised parsing. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 865–872, Sydney, Australia. Association for Computational Linguistics.

Rens Bod. 2006b. Unsupervised parsing with U-DOP. In Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), pages 85–92, New York City. Association for Computational Linguistics.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 193–203, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Andrew Drozdov, Patrick Verga, Mohit Yadav, Mohit Iyyer, and Andrew McCallum. 2019. Unsupervised latent tree induction with deep inside-outside recursive auto-encoders. 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), pages 1129–1141, Minneapolis, Minnesota. Association for Computational Linguistics.

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
Hao Peng, Roy Schwartz, and Noah A. Smith. 2019. PaLM: A hybrid parser and language model. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3642–3649, Hong Kong, China. Association for Computational Linguistics.

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 Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 616–623, Prague, Czech Republic. Association for Computational Linguistics.

Kenji Sagae, Yusuke Miyao, Rune Saetre, and Jun’ichi Tsujii. 2008. Evaluating the effects of treebank size in a practical application for parsing. In Software Engineering, Testing, and Quality Assurance for Natural Language Processing, pages 14–20, Columbus, Ohio. Association for Computational Linguistics.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Bidisha Samanta, Niloy Ganguly, and Soumen Chakrabarti. 2019. Improved sentiment detection via label transfer from monolingual to synthetic code-switched text. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3528–3537, Florence, Italy. Association for Computational Linguistics.

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 384–391, Prague, Czech Republic. Association for Computational Linguistics.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron Courville. 2018a. Neural language modeling by jointly learning syntax and lexicon. In Proceedings of the International Conference on Learning Representations.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018b. Straight to the tree: Constituency parsing with neural syntactic distance. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1171–1180, Melbourne, Australia. Association for Computational Linguistics.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In Proceedings of the International Conference on Learning Representations.

Haoyue Shi, Jiaoyuan Mao, Kevin Gimpel, and Karen Livescu. 2019. Visually grounded neural syntax acquisition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1842–1861, Florence, Italy. Association for Computational Linguistics.

Haoyue Shi, Hao Zhou, Jiaze Chen, and Lei Li. 2018. On tree-based neural sentence modeling. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4631–4641, Brussels, Belgium. Association for Computational Linguistics.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05), pages 354–362, Ann Arbor, Michigan. Association for Computational Linguistics.

Noah A. Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 569–576, Sydney, Australia. Association for Computational Linguistics.

Benjamin Snyder, Tuhira Naseem, and Regina Barzilay. 2009. Unsupervised multilingual grammar induction. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 73–81, Suntec, Singapore. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, Angel X. Chang, and Daniel Jurafsky. 2011. Unsupervised dependency parsing without gold part-of-speech tags. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1281–1290, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010a. From baby steps to leapfrog: How “less is more” in unsupervised dependency parsing. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 751–759, Los Angeles, California. Association for Computational Linguistics.
Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1983–1995, Seattle, Washington, USA. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010b. Viterbi training improves unsupervised dependency parsing. In Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pages 9–17, Uppsala, Sweden. Association for Computational Linguistics.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary. Association for Computational Linguistics.

Clara Vania, Yova Kementchedjhieva, Anders Søgaard, and Adam Lopez. 2019. A systematic comparison of methods for low-resource dependency parsing on genuinely low-resource languages. arXiv preprint arXiv:1909.02857.

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, pages 5998–6008.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019. Tree transformer: Integrating tree structures into self-attention. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1061–1070, Hong Kong, China. Association for Computational Linguistics.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1461–1470, Osaka, Japan. The COLING 2016 Organizing Committee.