Learning Syntax from Naturally-Occurring Bracketings

Tianze Shi
Cornell University
tianze@cs.cornell.edu

Ozan İrsoy
Bloomberg L.P.
oirsoy@bloomberg.net

Igor Malioutov
Bloomberg L.P.
imalioutov@bloomberg.net

Lillian Lee
Cornell University
llee@cs.cornell.edu

Abstract

Naturally-occurring bracketings, such as answer fragments to natural language questions and hyperlinks on webpages, can reflect human syntactic intuition regarding phrasal boundaries. Their availability and approximate correspondence to syntax make them appealing as distant information sources to incorporate into unsupervised constituency parsing. But they are noisy and incomplete; to address this challenge, we develop a partial-brackets-aware structured ramp loss in learning. Experiments demonstrate that our distantly-supervised models trained on naturally-occurring bracketing data are more accurate in inducing syntactic structures than competing unsupervised systems. On the English WSJ corpus, our models achieve an unlabeled F1 score of 68.9 for constituency parsing.[1]

1 Introduction

Constituency is a foundational building block for phrase-structure grammars. It captures the notion of what tokens can group together and act as a single unit. The motivating insight behind this paper is that constituency may be reflected in markups of bracketings that people provide in doing natural tasks. We term these segments *naturally-occurring bracketings* for their lack of intended syntactic annotation. These include, for example, the segments people pick out from sentences to refer to other Wikipedia pages or to answer semantically-oriented questions; see Figure 1 for an illustration.

Gathering such data requires low annotation expertise and effort. On the other hand, these data are not necessarily suitable for training parsers, as they often contain incomplete, incorrect and sometimes conflicting bracketing information. It is thus an empirical question whether and how much we

---

Wikipedia

Science fiction (sometimes shortened to sci-fi or SF) is a genre of speculative fiction that typically deals with imaginative and futuristic concepts such as advanced science and technology, space exploration, time travel, parallel universes, and extraterrestrial life.

QA-SRL

Context:

Republicans have been imploring the White House to compromise on the wage issue.

Q&As:

Q: Who have been imploring something?
    A: Republicans

Q: What have someone been imploring?
    A: To compromise on the wage issue

Q: What will someone compromise on?
    A: The wage issue

Corresponding bracketings:

[Republicans] have been imploring the White House [to compromise on [the wage issue]].

Figure 1: Two example types of naturally-occurring bracketings. Blue underlined texts in the Wikipedia sentence are hyperlinks. We bracket the QA-SRL sentence in matching colors according to the answers.

could learn syntax from these naturally-occurring bracketing data.

To overcome the challenge of learning from this kind of noisy data, we propose to train discriminative constituency parsers with structured ramp loss (Do et al., 2008), a technique previously adopted in machine translation (Gimpel and Smith, 2012). Specifically, we propose two loss functions to directly penalize predictions in conflict with available partial bracketing data, while allowing the parsers to induce the remaining structures.

We experiment with two types of naturally-occurring bracketing data, as illustrated in Figure 1. First, we consider English question-answer pairs collected for semantic role labeling (QA-SRL; He et al., 2015). The questions are designed for non-

---

experts to specify semantic arguments of predicates in the sentences. We observe that although no syntactic structures are explicitly asked for, humans tend to select constituents in their answers. Second, Wikipedia articles[2] are typically richly annotated with internal links to other articles. These links are marked on phrasal units that refer to standalone concepts, and similar to the QA-SRL data, they frequently coincide with syntactic constituents.

Experiment results show that naturally-occurring bracketings across both data sources indeed help our models induce syntactic constituency structures. Training on the QA-SRL bracketing data achieves an unlabeled F1 score of 68.9 on the English WSJ corpus, an accuracy competitive with state-of-the-art unsupervised constituency parsers that do not utilize such distant supervision data. We find that our proposed two loss functions have slightly different interactions with the two data sources, and that the QA-SRL and Wikipedia data have varying coverage of phrasal types, leading to different error profiles.

In sum, through this work, (1) we demonstrate that naturally-occurring bracketings are helpful for inducing syntactic structures, (2) we incorporate two new cost functions into structured ramp loss to train parsers with noisy bracketings, and (3) our distantly-supervised models achieve results competitive with the state of the art of unsupervised constituency parsing despite training with smaller data size (QA-SRL) or out-of-domain data (Wikipedia).

2   Naturally-Occurring Bracketings

Constituents are naturally reflected in various human cognitive processes, including speech production and perception (Garrett et al., 1966; Gee and Grosjean, 1983), reading behaviors (Hale, 2001; Boston et al., 2008), punctuation marks (Spitkovsky et al., 2011), and keystroke dynamics (Plank, 2016). Conversely, these externalized signals help us gain insight into constituency representations. We consider two such data sources:

a) Answer fragments   When questions are answered with fragments instead of full sentences, those fragments tend to form constituents. This phenomenon corresponds to a well-established constituency test in the linguistics literature (Carnie, 2012, pg. 98, inter alia).

Dataset	QA-SRL	Wikipedia
Number of sentences	1,241	926,077
Brackets/sentence	6.26	0.89
Single word	22.4%	35.8%
Constituent in reference	55.2%	31.1%
Conflicting w/ reference	11.8%	5.3%
SBAR	2.8%	0.07%
NP	36.8%	4.42%
VP	6.3%	0.07%
PP	13.3%	0.04%
ADJP	8.6%	1.48%
ADVP	30.5%	0.39%
Total	21.8%	1.91%

Table 1: Dataset statistics: number of bracketings per sentence (top), percentage of bracketing types (middle), and the reference phrases per label found in the natural bracketings (bottom). *Conflicting* means the bracket crosses some reference span. Reference parses for Wikipedia are generated by a parser trained on PTB.

b) Webpage hyperlinks   Since a hyperlink is a pointer to another location or action (e.g., `mailto:` links), anchor text often represents a conceptual unit related to the link destination. Indeed, Spitkovsky et al. (2010) first give empirical evidence that around half of the anchor text instances in their data respects constituent boundaries and Søgaard (2017) demonstrates that hyperlink data can help boost chunking accuracy in a multi-task learning setup.

Both types of data have been considered in previous work on dependency-grammar induction (Spitkovsky et al., 2010; Naseem and Barzilay, 2011), and in this work, we explore their efficacy for learning constituency structures.

For answer fragments, we use He et al.'s (2015) question-answering-driven semantic role labeling (QA-SRL) dataset, where annotators answer *wh*-questions regarding predicates in sentences drawn from the Wall Street Jounal (WSJ) section of the Penn Treebank (PTB; Marcus et al., 1993). For hyperlinks, we used a 1% sample of 2020-05-01 English Wikipedia, retaining only within-Wikipedia links.[3]

We compare our extracted naturally-occurring bracketings with the reference phrase-structure annotations:[4] Table 1 gives relevant statistics. Our results re-affirm Spitkovsky et al.'s (2010) finding that a large proportion of hyperlinks coin-

---

[2]We worked with articles in English.

[3]See Appendix A for details.

[4]For "ground-truth" structures in the Wikipedia data, we apply a state-of-the-art PTB-trained constituency parser (Kitaev et al., 2019).

cide with syntactic constituents. We also find that 22.4%/35.8% of the natural bracketings are single-word spans, which cannot facilitate parsing decisions, while 11.8%/5.3% of QA-SRL/Wikipedia spans actually conflict with the reference trees and can thus potentially harm training. The QA-SRL data seems more promising for inducing better-quality syntactic structures, as there are more bracketings available across a diverse set of constituent types.

3 Parsing Model

Preliminaries The inputs to our learning algorithm are tuples (w, B), where $w = w_1, \ldots, w_n$ is a length-n sentence and $B = \{(b_k, e_k)\}$ is a set of naturally-occurring bracketings, denoted by the beginning and ending indices b_k and e_k into the sentence w. As a first step, we extract BERT-based contextualized word representations (Devlin et al., 2019) to associate each token w_i with a vector $\mathbf{x}_i$.[5] See Appendix B for details.

Scoring Spans Based on the $\mathbf{x}_i$ vectors, we assign a score s_{ij} to each candidate span (i, j) in the sentence indicating its appropriateness as a constituent in the output structure. We adopt a biaffine scoring function (Dozat and Manning, 2017):

$$s_{ij} = [\mathbf{l}_i; 1]^T W [\mathbf{r}_j; 1],$$

where $[\mathbf{v}; 1]$ appends 1 to the end of vector $\mathbf{v}$, and

$$\mathbf{l}_i = \text{MLP}^{\text{left}}(\mathbf{x}_i) \quad \text{and} \quad \mathbf{r}_j = \text{MLP}^{\text{right}}(\mathbf{x}_j)$$

are the outputs of multi-layer perceptrons (MLPs) that take the vectors at span boundaries as inputs.[6]

Decoding We define the score $s(y)$ of a binary-branching constituency tree y to be the sum of scores of its spans. The best scoring tree among all valid trees $\mathcal{Y}$ can be found using the CKY algorithm (Cocke, 1969; Kasami, 1965; Younger, 1967).

Learning Large-margin training (Taskar et al., 2005) is a typical choice for *supervised* training of constituency parsers. It defines the following

---

loss function to encourage a large margin of at least $\Delta(y, y^*)$ between the gold tree y^* and any predicted tree y:

$$l = \max_{y \in \mathcal{Y}} [s(y) + \Delta(y, y^*)] - s(y^*),$$

where $\Delta(y, y^*)$ is a distance measure between y and y^*. We can reuse the CKY decoder for cost-augmented inference when the distance decomposes into individual spans with some function c:

$$\Delta(y, y^*) = \sum_{\text{span } (i,j) \text{ in } y} c(i, j, y^*).$$

In our setting, we do not have access to the gold-standard y^*, but instead we have a set of bracketings $\tilde{y}$. The scoring $s(\tilde{y})$ is not meaningful since $\tilde{y}$ is not a complete tree, so we adopt structured ramp loss (Do et al., 2008; Gimpel and Smith, 2012) and define

$$
\begin{aligned}
l &= \left(\max_{y \in \mathcal{Y}} [s(y) + \Delta(y, \tilde{y})] - s(\tilde{y}) \right) \\
&\quad + \left(s(\tilde{y}) - \max_{y \in \mathcal{Y}} [s(y) - \Delta(y, \tilde{y})] \right) \\
&= \max_{y \in \mathcal{Y}} [s(y) + \Delta(y, \tilde{y})] \\
&\quad - \max_{y \in \mathcal{Y}} [s(y) - \Delta(y, \tilde{y})],
\end{aligned}
$$

using a combination of cost-augmented and cost-diminished inference. This loss function can be understood as a sum of a convex and a concave large margin loss (Collobert et al., 2006), canceling out the term for directly scoring the gold-standard tree. We consider two methods for incorporating the partial bracketings into the cost functions:

$$
\begin{aligned}
c_{\text{loose}}(i, j, \tilde{y}) &= \mathbb{1}(\text{span } (i, j) \text{ conflicts with } \tilde{y}) \\
c_{\text{strict}}(i, j, \tilde{y}) &= \mathbb{1}(\text{span } (i, j) \text{ not in } \tilde{y}),
\end{aligned}
$$

where $\mathbb{1}$ is an indicator function. c_{loose} is more lenient than c_{strict} as it does not penalize spans that do not conflict with $\tilde{y}$. Both cost definitions promote structures containing bracketings in $\tilde{y}$.[7] In the supervised setting where $\tilde{y}$ refers to a fully-annotated tree y^* without conflicting span boundaries, c_{strict} is equal to c_{loose} and the resulting $\Delta(y, y^*)$ cost functions both correspond to the Hamming distance between y and y^*.

4 Experiments and Results

Data and Implementation We evaluate on the PTB (Marcus et al., 1993) with the standard splits

Model	PLM	Mean	Max
Random Trees		19.2	19.5
Left Branching			8.7
Right Branching			39.5
Upper bound			84.3
URNNG (Kim et al., 2019b)		—	45.4
PRPN (Shen et al., 2018)		47.3	47.9
ON (Shen et al., 2019)		48.1	50.0
DIORA (Drozdov et al., 2019)	○	—	58.9
CPCFG (Kim et al., 2019a)		55.2	60.1
S-DIORA (Drozdov et al., 2020)	○	57.6	64.0
Constituency Tests (Cao et al., 2020)	●	62.8	65.9
+URNNG (Cao et al., 2020)	●	67.9	**71.3**
This work:			
NOB$_{QA-SRL}$, c_{loose}	●	64.5	65.2
NOB$_{QA-SRL}$, c_{strict}	●	**68.9**	70.0
NOB$_{Wikipedia}$, c_{loose}	●	58.2	63.0
NOB$_{Wikipedia}$, c_{strict}	●	56.1	57.0

Table 2: Sentence-level unlabeled F1 scores (%) on the WSJ test set. ● in the PLM column denotes the use of context-sensitive pre-trained language models; ○ uses context-insensitive embedders from PLMs. Methods producing only binary-branching structures (including everything in this table) have an upperbound of 84.3% F1 score, since the gold trees can be non-binary.

Const. Type	Cao et al. (2020)	NOB$_{QA-SRL}$		NOB$_{Wikipedia}$	
		c_{loose}	c_{strict}	c_{loose}	c_{strict}
SBAR	85.3	**89.0**	87.7	66.7	48.3
NP	84.3	**85.5**	85.2	71.8	70.3
VP	**80.8**	52.3	70.9	62.4	49.6
PP	84.4	83.5	**86.5**	67.7	74.8
ADJP	55.6	58.1	57.3	**62.7**	60.1
ADVP	54.6	**76.9**	75.3	66.4	63.9

Table 3: Average recall (%) per consituent type.

(section 23 as the test set). QA-SRL contains 1,241 sentences drawn from the training split (sections 02-21) of the PTB. For Wikipedia, we use a sample of 332,079 sentences that are within 100 tokens long and contain multi-token internal hyperlinks. We fine-tune the pretrained BERT$_{base}$ features with a fixed number of mini-batch updates and report results based on five random runs for each setting. See Appendix B for detailed hyper-parameter settings and optimization procedures.

Evaluation We follow the evaluation setting of Kim et al. (2019a). More specifically, we discard punctuation and trivial spans (single-word and full-sentence spans) during evaluation and report sentence-level F1 scores as our main metrics.

Results Table 2 shows the evaluation results of our models trained on naturally-occurring brack-etings (NOB); Table 3 breaks down the recall

ratios for each constituent type. Our distantly-supervised models trained on QA-SRL are competitive with the state-of-the-art unsupervised results. When comparing our models with Cao et al. (2020), we obtain higher recalls on most constituent types except for VPs. Interestingly, QA-SRL data prefers c_{strict}, while c_{loose} gives better F1 score on Wikipedia; this correlates with the fact that QA-SRL has more bracketings per sentence (Table 1). Finally, our Wikipedia data has a larger relative percentage of ADJP bracketings, which explains the higher ADJP recall of the models trained on Wikipedia, despite their lower overall recalls.

5 Related Work

Unsupervised Parsing Our distantly-supervised setting is similar to unsupervised in the sense that it does not require syntactic annotations. Typically, lack of annotations implies that unsupervised parsers induce grammar from a raw stream of lexical or part-of-speech tokens (Clark, 2001; Klein, 2005) along with carefully designed inductive biases on parameter priors (Liang et al., 2007; Wang and Blunsom, 2013), language universals (Naseem et al., 2010; Martínez Alonso et al., 2017), cross-linguistic (Snyder et al., 2009; Berg-Kirkpatrick and Klein, 2010; Cohen and Smith, 2009; Han et al., 2019) and cross-modal (Shi et al., 2019) signals, structural constraints (Gillenwater et al., 2010; Noji et al., 2016; Jin et al., 2018), etc. The models are usually generative and learn from (re)constructing sentences based on induced structures (Shen et al., 2018, 2019; Drozdov et al., 2019; Kim et al., 2019a,b). Alternatively, one may use reinforcement learning to induce syntactic structures using rewards defined by end tasks (Yogatama et al., 2017; Choi et al., 2018; Havrylov et al., 2019). Our method is related to learning from constituency tests (Cao et al., 2020), but our use of bracketing data permits discriminative parsing models, which focus directly on the syntactic objective.

Learning from Partial Annotations Full syntactic annotations are costly to obtain, so the alternative solution of training parsers from partially-annotated data has attracted considerable research attention, especially within the context of active learning for dependency parsing (Sassano, 2005; Sassano and Kurohashi, 2010; Mirroshandel and Nasr, 2011; Flannery et al., 2011; Flannery and Mori, 2015; Li et al., 2016; Zhang et al., 2017) and grammar induction for constituency parsing

(Pereira and Schabes, 1992; Hwa, 1999; Riezler et al., 2002). These works typically require expert annotators to generate gold-standard, though partial, annotations. In contrast, our work considers the setting and the challenge of learning from noisy bracketing data, which is more comparable to Spreyer and Kuhn (2009) and Spreyer et al. (2010) on transfer learning for dependency parsing.

6 Conclusion and Future Work

We argue that naturally-occurring bracketings are a rich resource for inducing syntactic structures. They reflect human judgment of what constitutes a phrase and what does not. More importantly, they require low annotation expertise and effort; for example, webpage hyperlinks can be extracted essentially for free. Empirically, our models trained on QA-SRL and Wikipedia bracketings achieve competitive results with the state of the art on unsupervised constituency parsing.

Structural probes have been successful in extracting syntactic knowledge from frozen-weight pre-trained language models (e.g., Hewitt and Manning, 2019), but they still require direct syntactic supervision. Our work shows that it is also feasible to retrieve constituency trees from BERT-based models using distant supervision data.

Our models are limited to the unlabeled setting, and we leave it to future work to automatically cluster the naturally-occurring bracketings and to induce phrase labels. Our work also points to potential applications in (semi-)supervised settings including active learning and domain adaptation (Joshi et al., 2018). Future work can also consider other naturally-occurring bracketings induced from sources such as speech production, reading behavior, etc.

Acknowledgements

References

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297, Uppsala, Sweden. Association for Computational Linguistics.

Marisa Ferrara Boston, John Hale, Reinhold Kliegl, Umesh Patil, and Shravan Vasishth. 2008. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research*, 2(1):1–12.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Unsupervised parsing via constituency tests. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4798–4808, Online. Association for Computational Linguistics.

Andrew Carnie. 2012. *Syntax: A Generative Introduction*, third edition. Number 4 in Introducing Linguistics. Wiley-Blackwell, Hoboken, New Jersey.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5094–5101, New Orleans, Louisiana, USA. Association for the Advancement of Artificial Intelligence.

Alexander S. Clark. 2001. *Unsupervised Language Acquisition: Theory and Practice*. Ph.D. thesis, University of Sussex.

John Cocke. 1969. Programming languages and their compilers: Preliminary notes. Technical report, Courant Institute of Mathematical Sciences, New York University.

Shay Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82, Boulder, Colorado, USA. Association for Computational Linguistics.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. 2006. Trading convexity for scalability. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 201–208, Pittsburgh, Pennsylvania, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Chuong B. Do, Quoc V. Le, Choon H. Teo, Olivier Chapelle, and Alex J. Smola. 2008. Tighter bounds for structured estimation. In *Advances in Neural Information Processing Systems 21*, pages 281–288, Vancouver, Canada. Curran Associates, Inc.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France. OpenReview.net.

Andrew Drozdov, Subendhu Rongali, Yi-Pei Chen, Tim O'Gorman, Mohit Iyyer, and Andrew McCallum. 2020. Unsupervised parsing with S-DIORA: Single tree encoding for deep inside-outside recursive autoencoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4832–4845, Online. 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, USA. Association for Computational Linguistics.

Daniel Flannery, Yusuke Miayo, Graham Neubig, and Shinsuke Mori. 2011. Training dependency parsers from partially annotated corpora. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 776–784, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Daniel Flannery and Shinsuke Mori. 2015. Combining active learning and partial annotation for domain adaptation of a Japanese dependency parser. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 11–19, Bilbao, Spain. Association for Computational Linguistics.

Merrill Garrett, Thomas Bever, and Jerry Fodor. 1966. The active use of grammar in speech perception. *Perception & Psychophysics*, 1(1):30–32.

James Paul Gee and François Grosjean. 1983. Performance structures: A psycholinguistic and linguistic appraisal. *Cognitive Psychology*, 15(4):411–458.

Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 194–199, Uppsala, Sweden. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 221–231, Montréal, Canada. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

John Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA, USA. Association for Computational Linguistics.

Wenjuan Han, Ge Wang, Yong Jiang, and Kewei Tu. 2019. Multilingual grammar induction with continuous language identification. 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 5728–5733, Hong Kong, China. Association for Computational Linguistics.

Serhii Havrylov, Germán Kruszewski, and Armand Joulin. 2019. Cooperative learning of disjoint syntax and semantics. 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 1118–1128, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Rebecca Hwa. 1999. Supervised grammar induction using training data with limited constituent information. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 73–79, College Park, Maryland, USA. Association for Computational Linguistics.

Lifeng Jin, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2018. Unsupervised grammar induction with depth-bounded PCFG. *Transactions of the Association for Computational Linguistics*, 6:211–224.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1190–1199, Melbourne, Australia. Association for Computational Linguistics.

Tadao Kasami. 1965. An efficient recognition and syntax-analysis algorithm for context-free language. Technical report, Hawaii University Honolulu Department of Electrical Engineering.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019a. Compound probabilistic context-free grammars for grammar induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.

Yoon Kim, Alexander Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019b. Unsupervised recurrent neural network grammars. 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 1105–1117, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

Zhenghua Li, Min Zhang, Yue Zhang, Zhanyi Liu, Wenliang Chen, Hua Wu, and Haifeng Wang. 2016. Active learning for dependency parsing with partial annotation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 344–354, Berlin, Germany. Association for Computational Linguistics.

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697, Prague, Czech Republic. Association for Computational Linguistics.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *In Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, Atlanta, Georgia, USA.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Héctor Martínez Alonso, Željko Agić, Barbara Plank, and Anders Søgaard. 2017. Parsing Universal Dependencies without training. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 230–240, Valencia, Spain. Association for Computational Linguistics.

Seyed Abolghasem Mirroshandel and Alexis Nasr. 2011. Active learning for dependency parsing using partially annotated sentences. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 140–149, Dublin, Ireland. Association for Computational Linguistics.

Tahira Naseem and Regina Barzilay. 2011. Using semantic cues to learn syntax. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 902–907, San Francisco, California, USA. The AAAI Press.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1234–1244, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Hiroshi Noji, Yusuke Miyao, and Mark Johnson. 2016. Using left-corner parsing to encode universal structural constraints in grammar induction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 33–43, Austin, Texas, USA. Association for Computational Linguistics.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Newark, Delaware, USA. Association for Computational Linguistics.

Barbara Plank. 2016. Keystroke dynamics as signal for shallow syntactic parsing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 609–619, Osaka, Japan. The COLING 2016 Organizing Committee.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 271–278, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Manabu Sassano. 2005. Using a partially annotated corpus to build a dependency parser for Japanese. In *Proceedings of the Second International Joint Conference on Natural Language Processing: Full Papers*, pages 82–92, Jeju Island, Korea. Springer-Verlag Berlin Heidelberg.

Manabu Sassano and Sadao Kurohashi. 2010. Using smaller constituents rather than sentences in active learning for Japanese dependency parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 356–365, Uppsala, Sweden. Association for Computational Linguistics.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, British Columbia, Canada. OpenReview.net.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, LA, USA. OpenReview.net.

Haoyue Shi, Jiayuan 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.

Benjamin Snyder, Tahira 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.

Anders Søgaard. 2017. Using hyperlinks to improve multilingual partial parsers. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 67–71, Pisa, Italy. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011. Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 19–28, Portland, Oregon, USA. Association for Computational Linguistics.

Valentin I. Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. 2010. Profiting from mark-up: Hyper-text annotations for guided parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1278–1287, Uppsala, Sweden. Association for Computational Linguistics.

Kathrin Spreyer and Jonas Kuhn. 2009. Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 12–20, Boulder, Colorado, USA. Association for Computational Linguistics.

Kathrin Spreyer, Lilja Øvrelid, and Jonas Kuhn. 2010. Training parsers on partial trees: A cross-language comparison. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 1978–1985, Valletta, Malta. European Language Resources Association.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Ben Taskar, Vassil Chatalbashev, Daphne Koller, and Carlos Guestrin. 2005. Learning structured prediction models: A large margin approach. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 896–903, Bonn, Germany. Association for Computing Machinery.

Pengyu Wang and Phil Blunsom. 2013. Collapsed variational Bayesian inference for PCFGs. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 173–182, Sofia, Bulgaria. Association for Computational Linguistics.

Wenhui Wang and Baobao Chang. 2016. Graph-based dependency parsing with bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2315, Berlin, Germany. Association for Computational Linguistics.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France. OpenReview.net.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208.

Yue Zhang, Zhenghua Li, Jun Lang, Qingrong Xia, and Min Zhang. 2017. Dependency parsing with partial annotations: An empirical comparison. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 49–58, Taipei, Taiwan. Asian Federation of Natural Language Processing.

A Data

A.1 QA-SRL

He et al.'s (2015) question-answering-driven semantic role labeling dataset (QA-SRL) contains question-answer pairs for 1,241 sentences drawn originally from the training sections of the Penn Treebank (PTB; Marcus et al., 1993). The questions are generated by templates that ask about semantic arguments for all the predicates in a given sentence. Recorded human responses to the questions typically correspond to spans in the sentence. Each question can have multiple answers.

For all question-answer pairs, we first map the answers to consecutive spans in the corresponding sentences. We keep all exact matches when the answer text appears multiple times in the sentence, and we discard any answers that cannot be mapped to a consecutive span in the sentence.

A.2 Wikipedia

We randomly sample 1% of the articles from the 2020-05-01 snapshot of English Wikipedia[8]. We then split the documents into sentences and tokenize with spaCy.[9] This step leads to 926,077 sentences, as reported in Table 1. For ground-truth parse trees, we parse the sentences with Kitaev et al.'s (2019) state-of-the-art constituency parser trained on the PTB. For all the internal hyperlinks in the documents, where there is a hyperlink-tokenization mismatch, we retrieve the smallest span of tokens that covers the hyperlink. To construct the training set in our main experiments, we filter out sentences longer than 100 tokens and sentences without any multiple-token internal hyperlinks. These pre-processing procedures produce 332,079 training sentences.

B Implementation Details

Feature Extractor We use the pretrained $\text{BERT}_{\text{base}}$ model as our feature extractor.[10] For each word in the sentence, we tokenize it with BERT's WordPiece tokenizer, and we take the BERT vector of the last token at the final BERT hidden layer as representation for each word. The feature extractor is fine-tuned along with model training.

Span Scoring MLP[left] and MLP[right] are single-layer MLPs: they both consist of a linear layer projecting BERT representations to 256-dimensional vectors, followed by a leaky ReLU activation function (Maas et al., 2013). The constituent scoring component has parameter $W \in \mathbb{R}^{257 \times 257}$. All the parameters are randomly initialized (Glorot and Bengio, 2010).

Training and Optimization We optimize the neural networks using the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-12}$. For each batch, we sample 8 sentences from the training set and average the loss collected for each sentence. The gradients are clipped at 1.0 before back propagation. The learning rate linearly increases from zero to 1×10^{-5} in 2,000 training steps. After warmup, we keep training the model until we reach 20,000 training steps. We do not perform early stopping, since in the unsupervised parsing setting, we do not look at validation accuracies until we finish training. We leave it as future work to explore other model selection strategies.

Hyperparameter Selection We use the default recommended β_1, β_2, and ϵ values for the Adam optimizer, and we use a typical fine-tuning learning rate for the pre-trained BERT model (Devlin et al., 2019). The number of training steps is based on our preliminary observation of the convergence of the training loss, and the batch size is limited by our computing hardware. We fix the initial values we set for the size of the biaffine matrix (257×257) and the number of warmup steps (2,000) throughout our experiments. A better hyperparameter selection strategy may lead to improved results.

Speed For a length-n sentence, the time complexity for the CKY decoder is $O(n^3)$. On a RTX 2080 GPU, our model parses 409 sentences per second on average and the training process for each model finishes within 2 hours.