Unsupervised Chunking as Syntactic Structure Induction with a Knowledge-Transfer Approach

Anup Anand Deshmukh[*1], **Qianqiu Zhang**[2], **Ming Li**[1], **Jimmy Lin**[1], **and Lili Mou**[2]
[1]David R. Cheriton School of Computer Science, University of Waterloo
[2]Dept. Computing Science & Alberta Machine Intelligence Institute (Amii), University of Alberta
[1]{aa2deshmukh, mli, jimmylin}@uwaterloo.ca
[2]qianqiu@ualberta.ca doublepower.mou@gmail.com

Abstract

In this paper, we address *unsupervised chunking* as a new task of syntactic structure induction, which is helpful for understanding the linguistic structures of human languages as well as processing low-resource languages. We propose a knowledge-transfer approach that heuristically induces chunk labels from state-of-the-art unsupervised parsing models; a hierarchical recurrent neural network (HRNN) learns from such induced chunk labels to smooth out the noise of the heuristics. Experiments show that our approach largely bridges the gap between supervised and unsupervised chunking.[1]

1 Introduction

Understanding the linguistic structure of language (e.g., parsing and chunking) is an important research topic in NLP. Most previous work employs supervised machine learning methods to predict linguistic structures. While these methods achieve high performance, they need massive data labeled with linguistic structures, such as treebanks (Marcus et al., 1993). Existing resources are mainly constructed for widely used languages (e.g., English); further constructing new treebanks for low-resource languages is cumbersome and expensive.

Unsupervised syntactic structure induction has been attracting increasing interest in recent years (Kim et al., 2019a; Shen et al., 2018a,b). This task concerns discovering linguistic structures of text without using labeled data. It is important to NLP research because it can be potentially used for low-resource languages and also be a first pass in annotating large treebanks for them. Moreover, grammar learned by these unsupervised methods shed light on linguistic theories.

Previous unsupervised syntactic structure mainly focuses on the task of constituency parsing which organize words in a hierarchical manner (Kim et al., 2019a,b; Shen et al., 2018a). Recently, Shen et al. (2021) propose to jointly induce constituency and dependency structures from text.

In this work, we address *unsupervised chunking*, another meaningful task of linguistic structure discovery. The chunking task aims to group the words of a sentence into chunks (roughly speaking, phrases) in a non-hierarchical fashion (Sang and Buchholz, 2000; Kudo and Matsumoto, 2001), and our setting is to detect chunks without the supervision of annotated linguistic structures.

In fact, unsupervised chunking has real-world applications, as understanding text fundamentally requires finding spans like noun phrases and verb phrases. It would benefit various downstream tasks, such as keywords extraction (Firoozeh et al., 2020), named entity recognition (Sano et al., 2017), and open information extraction (Niklaus et al., 2018).

In our paper, we propose a knowledge-transfer approach to unsupervised chunking by hierarchical recurrent neural networks (HRNN). We utilize the recent advances of unsupervised parsers, and propose a maximal left-branching heuristic to induce chunk labels from unsupervised parsing. Without any supervision of annotated grammars, such heuristic leads to reasonable (albeit noisy and imperfect) chunks. We further design an HRNN model that learns from the heuristic chunk labels. Our HRNN involves a trainable chunking gate that switches between a lower word-level RNN and a upper phrase-level RNN. This explicitly models the composition of words into chunks and chunks into the sentence. Results on three datasets show that our HRNN can indeed smooth out the noise of heuristically induced chunk labels, with a considerable improvement in terms of the phrase-F1 score; such observations are consistent in different domains and languages.

[*] Work partially done as a co-op intern at the University of Alberta.

[1]Our code and output are released at https://github.com/Anup-Deshmukh/Unsupervised-Chunking

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3626–3634
November 7–11, 2021.

Related Work. Unsupervised syntactic structure detection has attracted much attention in early NLP research because of its use in low-resource scenarios (Clark, 2001; Klein, 2005). Klein and Manning (2002) propose to model constituency and context for each spans with an Expectation–Maximization (EM) algorithm. Early work also focuses on unsupervised dependency parsing for syntactic structure induction (Seginer, 2007; Paskin, 2001). Klein and Manning (2004) combine constituency and dependency models via co-training to further boost their performance.

To learn the syntactic structures, Haghighi and Klein (2006) propose a probabilistic context-free grammar (PCFG), augmented with manually designed features. Reichart and Rappoport (2008) perform clustering by syntactic features to obtain labeled parse trees. Clark (2001) clusters sequences of tags based on their local mutual information to build parse trees. Such early studies typically used heuristics, linguistic knowledge, and manually designed features for unsupervised syntactic structure induction (Wolff, 1988; Klein and Manning, 2002; Clark, 2001).

In the deep learning era, unsupervised parsing has revived the interest. Socher et al. (2011) propose the recursive autoencoder, where a binary tree is built by greedily minimizing the reconstruction loss. Such recursive tree structures can also be learned in an unsupervised way by CYK-style marginalization (Maillard et al., 2019) and Gumbel-softmax (Choi et al., 2018). Yogatama et al. (2017) learn a shift–reduce parser by reinforcement learning towards a downstream task. However, evidence shows the above approaches do not yield linguistically plausible trees (Williams et al., 2018).

Shen et al. propose to model the syntactic distance (2018a) or syntactic ordering (2018b) to build parse trees. Kim et al. (2019b) propose a Compound PCFG for unsupervised parsing. The trees given by these approaches are more correlated with constituency trees.

Li et al. (2019) propose to transfer knowledge among several unsupervised parsers and obtain better performance. Our work is inspired by such knowledge transfer, but we propose insightful heuristics that induces chunk labels from unsupervised parsers. We also design Hierarchical RNN to learn from induced chunk labels.

Previous studies address unsupervised chunking as an important task in speech processing; they use acoustic information to determine the chunks (Pate and Goldwater, 2011; Barrett et al., 2018). Our work only considers textual information, and views unsupervised chunking as a new task of syntactic structure induction.

2 Model

In this section, we will first induce chunking labels from state-of-the-art unsupervised parsing. Then, we will train a hierarchical RNN to learn from induced labels to smooth out the noise.

2.1 Inducing Chunk Labels from Unsupervised Parsing

We propose to induce chunk labels from state-of-the-art unsupervised parsers. The intuition is that the chunking structure can be thought of as a flattened parse tree, and thus agree with the parsing structure to some extent. Our knowledge-transfer approach is able to take advantage of recent advances in unsupervised parsing (Kim et al., 2019a,b).

Specifically, we adopt the Compound PCFG which is a 5-tuple grammar $\mathcal{G} = (\mathcal{S}, \mathcal{N}, \mathcal{P}, \Sigma, \mathcal{R})$, where $\mathcal{S}$ is a start symbol; $\mathcal{N}$, $\mathcal{P}$, and Σ are finite sets of nonterminal, preterminal, and terminal symbols, respectively. $\mathcal{R}$ is a finite set of rules taking one of the following forms:

$$S \rightarrow A \qquad A \in \mathcal{N} \tag{1}$$

$$A \rightarrow B\ C \qquad B, C \in \mathcal{N} \cup \mathcal{P} \tag{2}$$

$$T \rightarrow w \qquad T \in \mathcal{P}, w \in \Sigma \tag{3}$$

where $S \rightarrow A$ is the start of a sentence and $T \rightarrow w$ indicates the generation of a word. $A \rightarrow B\,C$ models the bifurcations of a binary constituency tree, where a constituent node is not explicitly associated with a type (e.g., noun phrase).

In addition, the model maintains a sentence-level continuous random vector, serving as the prior of PCFG. The Compound PCFG is trained by maximum likelihood of text, where the PCFG is marginalized by the Viterbi-like algorithm and the continuous distribution is treated by amortized variational inference. We refer readers to Kim et al. (2019b) for details.

We would like to induce chunk labels from Compound PCFG, which is a state-of-the-art unsupervised parser. Given a sentence, we obtain its parse tree by applying the Viterbi-like CYK algorithm to Compound PCFG.

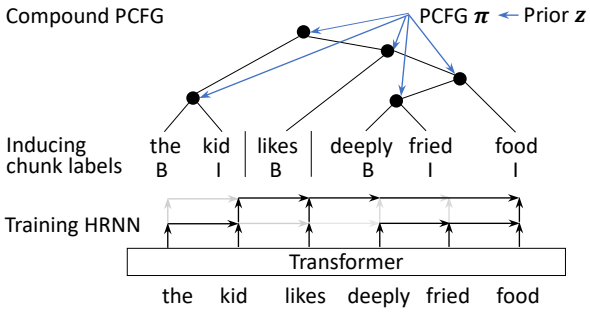

Figure 1: Overview of our approach.

We propose a simple yet effective heuristic that extracts maximal left-branching subtrees as chunks. As known, the English language is strongly biased to right-branching structures (Williams et al., 2018; Li et al., 2019). We observe, on the other hand, that a left-branching structure typically indicates closely related words. Here, a *left-branching* subtree means that the words are grouped in the form of $((\cdots((\mathrm{x}_i\mathrm{x}_{i+1})\mathrm{x}_{i+2})\cdots)\mathrm{x}_{i+n-1})$. A left-branching subtree for words $\mathrm{x}_i\cdots\mathrm{x}_{i+n-1}$ is *maximal* if neither $\mathrm{x}_{i-1}\mathrm{x}_i\cdots\mathrm{x}_{i+n-1}$ nor $\mathrm{x}_i\cdots\mathrm{x}_{i+n-1}\mathrm{x}_{i+n}$ is left-branching. We extract all maximal left-branching subtrees as chunks.

In Figure 1, for example, "deeply fried food" is a three-word maximal left-branching subtree $\wedge$, whereas "the kid" and "likes" are also maximal left-branching subtrees (although degenerated). Our heuristic treats them as chunks. The following theorem shows that our heuristic can unambiguously give chunk labels for any sentence with any parse tree. (See Appendix A for proof.)

Theorem 1. *Given any binary parse tree, every word will belong to one and only one chunk by the maximal left-branching heuristic.*

Our simple heuristic achieves reasonable chunking performance, although it is noisy. Then, HRNN learning (discussed in next part) will smooth out such noise and yield more meaningful chunks.

2.2 Training Hierarchical RNN

We would like to train a machine learning model to learn from the Compound PCFG-induced chunk labels. Our intuition is that a learning machine pools the knowledge of different samples into a parametric model and thus may smooth out the noise of our heuristics.

Specifically, we run Compound PCFG on an unlabeled corpus to obtain chunk labels in the BI schema (Ramshaw and Marcus, 1995), where "B" refers to the beginning of a chunk, and "I" refers to the inside of a chunk. Then, a machine learning model (e.g., a neural network) will learn from the pseudo-groundtruth labels.

We observe that a classic RNN or Transformer may not be suitable for the chunking task, because the prediction at a time step is unaware of previous predicted chunks, thus lacking autoregressiveness. Feeding predicted chunk labels like a sequence-to-sequence model is not adequate, because a BI label only contains one bit information and cannot provide useful autoregressive information either.

To this end, we design a hierarchical RNN to model the autoregressivenss of predicted chunks by altering the neural structure. Our HRNN contains a lower word-level RNN and an upper chunk-level RNN. We also design a gating mechanism that switches between the two RNNs in a soft manner, also serving as the predicted probability of the chunk label.

Let $\mathrm{x}^{(1)},\cdots,\mathrm{x}^{(n)}$ be the words in a sentence. We first apply the pretrained language model BERT (Kenton et al., 2019) to obtain the contextual representations of the words, denoted by $\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(n)}$. This helps our model to understand the global context of the sentence. For a step t, we first predict a switching gate $m^{(t)}\in(0,1)$ as the chunking decision.[2]

$$m^{(t)} = \sigma(W[\underline{\boldsymbol{h}}^{(t-1)};\overline{\boldsymbol{h}}^{(t-1)};\boldsymbol{x}^{(t)}]) \quad (4)$$

where $\underline{\boldsymbol{h}}^{(t-1)}$ is the hidden state of the lower RNN and $\overline{\boldsymbol{h}}^{(t-1)}$ is that of the upper RNN. Semicolon represents vector concatenation, and σ represents the sigmoid function.

Such a switching gate is also used to control the information flow by altering the network architecture, shown in Figure 1. In this way, it provides meaningful autoregressive information, as it makes HRNN aware of previously detected chunks.

Suppose our model predicts that the tth word is the beginning of a chunk. This essentially "cuts" the sequence into two parts at this step. The lower RNN and upper RNN are updated by

$$\underline{\boldsymbol{h}}^{(t)}_{\text{cut}} = \underline{f}\,(\boldsymbol{x}^{(t)},\,\underline{\boldsymbol{h}}^{(\text{sos})}) \quad (5)$$

$$\overline{\boldsymbol{h}}^{(t)}_{\text{cut}} = \overline{f}\,(\underline{\boldsymbol{h}}^{(t-1)},\,\overline{\boldsymbol{h}}^{(t-1)}) \quad (6)$$

where $\underline{f}$ and $\overline{f}$ are the transition functions of the two RNNs, respectively.

[2] $m^{(t)}=1$ corresponds to "B," i.e., a new chunk, and $m^{(t)}=0$ corresponds to "I," i.e., inside of a chunk.

Method	CoNLL-2000 English (Newswire)		CoNLL-2003 German (Newswire)		English Web Treebank (Reviews)	
	Phrase F1	Tag Acc.	Phrase F1	Tag Acc.	Phrase F1	Tag Acc.
Supervised Methods						
NLTK-tagger-chunker	83.71	89.51	87.82	93.59	-	-
Supervised HMM	87.68	93.99	90.16	94.77	98.62	99.44
Unsupervised Methods						
PMI Chunker	35.64	64.5	42.19	64.42	32.28	65.34
Baum–Welch HMM	25.04	58.93	27.01	58.52	24.17	58.02
LM Chunker	42.05	68.74	45.06	68.62	31.23	62.55
Compound PCFG Chunker	62.89	81.64	55.94	75.54	58.17	79.33
LM → HRNN	47.99	73.10	48.40	70.10	39.43	70.5
Compound PCFG → HRNN	**68.12**	**83.90**	**57.14**	**75.81**	**64.32**	**83.25**

Table 1: Chunking performance on the CoNLL-2000, CoNLL-2003, and English Web Treebank. For both CoNLL datasets, the Phrasal F1 and Tag accuracy scores are calculated against groundtruth chunk labels. For the English Web Treebank, we treat the chunks generated by (NLTK-tagger, Bird, 2006)) as groundtruth labels. → refers to our knowledge-transfer approaches.

In Equation (5), the lower RNN ignores its previous hidden state but restarts from a learnable initial state $\underline{\boldsymbol{h}}^{(\text{sos})}$, due to the prediction of a new phrase. In Equation (6), the upper RNN picks the newly formed phrase with representation $\underline{\boldsymbol{h}}^{(t-1)}$ captured by the lower RNN, and fuses it with the previous chunk's representation in the upper RNN $\overline{\boldsymbol{h}}^{(t-1)}$.

Suppose our model predicts that the tth word is not the beginning of a chunk, i.e., "no cut" is performed at this step. The RNNs are updated by

$$\underline{\boldsymbol{h}}^{(t)}_{\text{nocut}} = \underline{f}\left(\boldsymbol{x}^{(t)}, \underline{\boldsymbol{h}}^{(t-1)}\right) \tag{7}$$

$$\overline{\boldsymbol{h}}^{(t)}_{\text{nocut}} = \overline{\boldsymbol{h}}^{(t-1)} \tag{8}$$

Here, the lower RNN updates its hidden state with the input $\boldsymbol{x}^{(t)}$ as a normal RNN, whereas the upper RNN is idle because no phrase is formed.

The "cut" and "nocut" cases can be unified by

$$\overline{\boldsymbol{h}}^{(t)} = m^{(t)}\, \overline{\boldsymbol{h}}^{(t)}_{\text{cut}} + (1 - m^{(t)})\, \overline{\boldsymbol{h}}^{(t)}_{\text{nocut}} \tag{9}$$

$$\underline{\boldsymbol{h}}^{(t)} = m^{(t)}\, \underline{\boldsymbol{h}}^{(t)}_{\text{cut}} + (1 - m^{(t)})\, \underline{\boldsymbol{h}}^{(t)}_{\text{nocut}} \tag{10}$$

In fact, we keep $m^{(t)}$ as a real number and fuse the lower RNN and upper RNN in a soft manner. This is because chunking by its nature may be ambiguous, and our soft gating mechanism is able to better preserve the information.

3 Experiments

Setup. We used the CoNLL-2000 (Sang and Buchholz, 2000), CoNLL-2003 (Sang and De Meulder, 2003), and English Web Treebank (Bies et al., 2012) for evaluation. We compare the model output with groundtruth chunks in terms of phrase F1 and tag accuracy. Dataset details and our experimental settings are presented in Appendix B.

Main Results. Table 1 presents main results of our knowledge-transfer approach. In addition to Compound PCFG, we also adopt another state-of-the-art unsupervised parser (Kim et al., 2019a) based on the features of a pretrained language model (LM). Specifically, we threshold the BERT (Kenton et al., 2019) similarity of consecutive words for chunking. We observe that the LM-based unsupervised chunker is worse than the Compound PCFG. Therefore, our main model variant uses Compound PCFG as the "teacher" model, i.e., the source of knowledge transfer. We train our student HRNN model to learn from the heuristically induced chunk labels. Results show that we achieve an improvement of more than 5 percentage points in phrase F1 based on either the LM-based chunker or Compound PCFG (42.05 vs. 47.99; 62.89 vs. 68.12) on the CoNLL-2000 dataset. The large margins imply that our HRNN can indeed smooth out the noise of heuristics and capture the chunking patterns.

We evaluate our knowledge-transfer approach on a different language (German) and a different domain (English Web Treebank). The results show a similar trend as the CoNLL-2000 dataset. This highlights the generality of our approach in different languages and domains.

We also tested traditional unsupervised methods for chunking, such as thresholding point-wise mutual information (PMI, de Cruys and Tim, 2011) and the Baum–Welch algorithm for the hidden Markov model (HMM, Rabiner, 1989). These methods perform significantly worse than recent advances in unsupervised syntactic structure discovery. In general, our knowledge transfer approach with HRNN largely bridges the gap between super-

Method	Phrase F1	Tag Acc.	Time (Sec.)
CoNLL-2000 (English)			
Compund PCFG	62.89	81.64	1803.90
Our HRNN model	**68.12**	**83.90**	**364.71**
CoNLL-2003 (German)			
Compund PCFG	55.94	75.54	163.04
Our HRNN model	**57.14**	**75.81**	**71.38**
English Web Treebank			
Compund PCFG	58.17	79.33	311.38
Our HRNN model	**64.32**	**83.25**	**167.29**

Table 2: Comparing the chunking quality and inference efficiency of the teacher Compound PCFG and our student HRNN. The inference time (in second) is obtained on NVIDIA Quadro RTX 6000 GPU with 25 GB RAM.

Chunking Heuristics	Phrase F1	Tag Acc.
1-word & 2-word chunks	55.72	75.14
Maximal right branching	40.83	69.28
Maximal left branching	**62.89**	**81.64**

Table 3: Analysis of chunking heuristics. HRNN is not applied in this comparison.

#	Method	Phrase F1	Tag Acc.
1	Teacher: Compound PCFG	62.89	81.64
2	→ HRNN only	65.01	82.22
3	→ BERT+1-layer RNN	67.19	83.86
4	→ BERT+2-layer RNN	66.53	83.34
5	→ BERT+HRNN (hard)	67.90	83.80
6	→ BERT+HRNN	**68.12**	**83.90**

Table 4: Ablation study of the student model.

vised and unsupervised chunking.

We compare the inference efficiency of our student HRNN and the teacher Compound PCFG in Table 2. We observe that Compound PCFG is slow in inference, as it requires Monte Carlo sampling to marginalize the latent variable and dynamic programming to marginalize the PCFG. Our HRNN not only yields higher-quality chunks, but also is 2-5x faster. Compound PCFG uses the Viterbi-like CYK algorithm for building parse trees, which has the worst case running time of $\mathcal{O}(n^3)$, where n is the length of the sentence. Thus, efficiency improvement is larger on the CoNLL-2000 dataset, as it contains longer sentences (shown in Table 5, Appendix B).

Analysis. We provide detailed analyses of our maximal left-branching chunking heuristic and student HRNN model to better understand their contribution. We chose the CoNLL-2000 dataset as our testbed, due to constraints of time and space.

Table 3 compares the heuristics that induce chunks from parse trees. We observe that our maximal left-branching heuristic outperforms right-branching by 20 points in Phrase F1. We also introduce a thresholding approach that extracts one-word and two-word chunks only, since most groundtruth chunks contain one or two words. The performance of such heuristic is higher than right-branching, but worse than our left-branching. The results are consistent with our conjecture that right-branching is a common structure of English and does not suggest meaningful chunks. On the contrary, left-branching indicates closely related words and is an effective heuristic for inducing chunks from parse trees.

Table 4 presents an ablation study on the student model. As seen, all student models outperform the teacher model, showing that the imperfection of chunk heuristics can indeed be smoothed out by a machine learning model.

However, a classic RNN or the Transformer predicts chunk labels individually, which does not provide autoregressive information. The performance is worse than HRNN even if the number of layers is controlled (Rows 4 vs. 6). The HRNN using soft gates outperforms a hard HRNN (Rows 5 vs. 6). This verifies that our soft HRNN can better handle the ambiguity of chunks and provide better autoregressive information. Building HRNN on top of BERT is also helpful (Rows 2 vs. 6), as BERT can capture global contextual information.

4 Conclusion

In this paper, we address a new task of syntactic structure discovery, namely, unsupervised chunking. We propose a hierarchical RNN with soft gates to learn from the chunk labels inducted by a state-of-the-art unsupervised parser, Compound PCFG. Results show that our approach largely bridges the gap between supervised and unsupervised chunking. We also show rigorous analysis on our chunk heuristics and the student model's architecture.

Acknowledgments

The work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant Nos. RGPIN2020-04465 and OGP0046506. Lili Mou is supported by the Amii Fellow Program, the Canada CIFAR AI Chair Program, and a donation from DeepMind. This research is also supported by Compute Canada (www.computecanada.ca).

References

Maria Barrett, Ana Valeria González-Garduño, Lea Frermann, and Anders Søgaard. 2018. Unsupervised induction of linguistic categories with records of reading, speaking, and writing. 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 2028–2038.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. *Linguistic Data Consortium*.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5094–5101.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning*.

Van de Cruys and Tim. 2011. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 16–20.

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.

Nazanin Firoozeh, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. Keyword extraction: Issues and methods. *Natural Language Engineering*, 26(3):259–291.

Aria Haghighi and Dan Klein. 2006. Prototype-driven grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 881–888.

Jacob Kenton, Devlin Ming-Wei Chang, and Lee Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2019a. Are pre-trained language models aware of phrases? Simple but strong baselines for grammar induction. In *International Conference on Learning Representations*.

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

Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Stanford University.

Dan Klein and Christopher D Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135.

Dan Klein and Christopher D 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*, pages 478–485.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Bowen Li, Lili Mou, and Frank Keller. 2019. An imitation learning approach to unsupervised parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3485–3492.

Jean Maillard, Stephen Clark, and Dani Yogatama. 2019. Jointly learning sentence embeddings and syntax with unsupervised tree-LSTMs. *Natural Language Engineering*, 25(4):433–449.

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

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878.

Mark A Paskin. 2001. Grammatical bigrams. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, pages 91–97.

John K Pate and Sharon Goldwater. 2011. Unsupervised syntactic chunking with acoustic cues: Computational models for prosodic bootstrapping. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 20–29.

Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Lance A Ramshaw and Mitchell P Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*, pages 157–176.

Roi Reichart and Ari Rappoport. 2008. Unsupervised induction of labeled parse trees by clustering with syntactic features. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 721–728.

Erik Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, pages 127–132.

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

Motoki Sano, Hiroyuki Shindo, Ikuya Yamada, and Yuji Matsumoto. 2017. Segment-level neural conditional random fields for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 97–102.

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

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

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018b. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. 2021. StructFormer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 7196–7209.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics*, 6:253–267.

J Gerard Wolff. 1988. Learning syntax and meanings through optimization and distributional analysis. In *Categories and Processes in Language Acquisition*, pages 179–215.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. In *International Conference on Learning Representations*.

A Proof of Theorem 1.

Theorem 1. *Given any binary parse tree, every word will belong to one and only one chunk by the maximal left-branching heuristic.*

Proof. [Existence] A single word itself is a left-branching subtree, which belongs to some maximal left-branching subtree.

[Uniqueness] We will show that two different maximal left-branching subtrees $\mathbf{s}_1$ and $\mathbf{s}_2$ cannot overlap. Assume by way of contradiction that there exists a word x_i in both $\mathbf{s}_1$ and $\mathbf{s}_2$. Then, $\mathbf{s}_1$ must be a substructure of $\mathbf{s}_2$ or vice versa; otherwise, the paths root–$\mathbf{s}_1$–x_i and root–$\mathbf{s}_2$–x_i violate the acyclic nature of a tree. But $\mathbf{s}_1$ being a subtree of $\mathbf{s}_2$ (or vice versa) contradicts with the maximality of $\mathbf{s}_1$ and $\mathbf{s}_2$. □

This easy theorem shows our maximal left-branching heuristic can unambiguously give chunk labels for any sentence with any binary parse tree.

B Experimental Setup

Datasets. We used the CoNLL-2000 (Sang and Buchholz, 2000), CoNLL-2003 (Sang and De Meulder, 2003), and English Web Treebank (Bies et al., 2012) for evaluation. CoNLL-2000 is widely used for the task of chunking and contains groundtruth chunk labels. CoNLL-2003 (German) dataset was developed for language-independent named entity recognition (Sang and De Meulder, 2003) which also contains groundtruth chunk labels for the entities. Both CoNLL-2000 and CoNLL-2003 contain sentences from the newswire domain. To evaluate the performance on a different domain, we make use of the English Web Treebank (Bies et al., 2012). It consists of online review sentences and their manually annotated parse trees. We use state-of-the-art supervised chunker (NLTK-tagger, Bird, 2006) to generate chunk labels for these sentences. Table 5 summarizes dataset statistics.

Dataset	#Train	#Val	#Test	Avg. len
CoNLL-2000 (English)	7929	950	2003	20.7
CoNLL-2003 (German)	7000	2000	1000	11
English Web Treebank	6496	1856	936	13.7

Table 5: Dataset statistics.

Our work is for unsupervised chunking, and thus we did not use the chunk labels of the training set. Instead, the training sentences were used for unsupervised parser to perform knowledge transfer, i.e., we predicted pseudo-chunk labels by Compound PCFG to train the Hierarchical RNN.

CoNLL-2000 (English) and CoNLL-2003 (German) datasets are labeled with the BIO schema, where "O" indicates outside a chunk (mainly punctuation). We followed the BI schema and ignored the "O" tokens.

We adopted the standard evaluation script from the CoNLL-2000 shared task to evaluate our chunk labels (Sang and Buchholz, 2000). It calculates the phrase F1 score and the tag accuracy of the predicted chunks against groundtruth labels from the dataset.

Model Settings. We employed the pretrained BERT (Kenton et al., 2019) to capture global contextual sentence information. The HRNN uses vanilla transition with 100 dimensions. In our preliminary experiments, we tried 300 dimension and achieve very close performance, suggesting that the model capacity is already enough for chunking. This is also evidenced by Rows 3–4, Table 4. We did not tune hyperparameters much, as our work focuses scientific questions of unsupervised chunking and knowledge transfer, instead of hyperparameter engineering.

We used the Adam optimizer to train the student model during knowledge transfer. We picked the best model by validation for early stopping, following most work on unsupervised parsing (Drozdov et al., 2019; Li et al., 2019). Roughly, such fine-tuning did not exceed 15 epochs.

C Case Study

In Figure 2, we present a few examples of chunking structures generated by both HRNN and Compound PCFG (teacher model) along with the groundtruth.

Our method is able to detect longer noun phrases, such as *small cable-television systems* (Example 1) and *white house press secretary marlin fitzwater* (Example 3), which agree more with the groundtruth chunks.

HRNN is also able to correct nonsensical chunks produced by Compound PCFG. In Example 2, the two words *of survival* is split into two chunks *of* and *survival* as they are not in a same semantic unit. In Example 3, (*bush*) (*aids lawmakers*) is corrected to (*bush aids*) (*lawmakers*).

In general, HRNN not only effectively learns

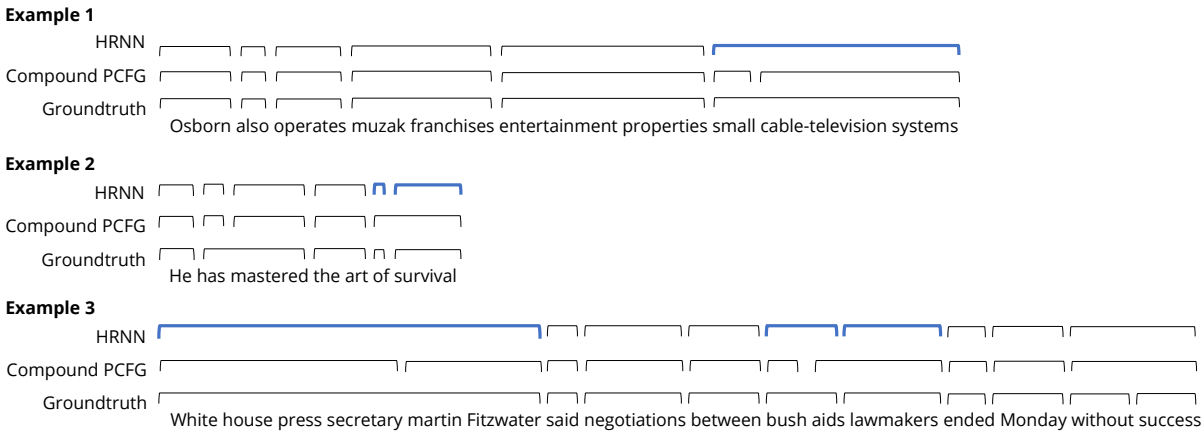

Figure 2: Examples of chunking structures produced by HRNN and Compound PCFG. The difference is highlighted in thick blue. We also show groundtruth chunks for reference.

the chunking patterns from Compound PCFG, but also can smooth out its noise and achieve higher performance for unsupervised chunking.