Depth-bounding is effective: Improvements and evaluation of unsupervised PCFG induction

Lifeng Jin
Department of Linguistics
The Ohio State University
jin.544@osu.edu

Finale Doshi-Velez
Harvard University
finale@seas.harvard.edu

Timothy Miller
Boston Children's Hospital &
Harvard Medical School
timothy.miller@childrens.harvard.edu

William Schuler
Department of Linguistics
The Ohio State University
schuler@ling.osu.edu

Lane Schwartz
Department of Linguistics
University of Illinois at Urbana-Champaign
lanes@illinois.edu

Abstract

There have been several recent attempts to improve the accuracy of grammar induction systems by bounding the recursive complexity of the induction model (Ponvert et al., 2011; Noji and Johnson, 2016; Shain et al., 2016; Jin et al., 2018). Modern depth-bounded grammar inducers have been shown to be more accurate than early unbounded PCFG inducers, but this technique has never been compared against unbounded induction within the same system, in part because most previous depth-bounding models are built around sequence models, the complexity of which grows exponentially with the maximum allowed depth. The present work instead applies depth bounds within a chart-based Bayesian PCFG inducer (Johnson et al., 2007b), where bounding can be switched on and off, and then samples trees with and without bounding.[1] Results show that depth-bounding is indeed significantly effective in limiting the search space of the inducer and thereby increasing the accuracy of the resulting parsing model. Moreover, parsing results on English, Chinese and German show that this bounded model with a new inference technique is able to produce parse trees more accurately than or competitively with state-of-the-art constituency-based grammar induction models.

1 Introduction

Unsupervised grammar inducers hypothesize hierarchical structures for strings of words. Using context-free grammars (CFGs) to define these structures, previous attempts at either CFG parameter estimation (Carroll and Charniak, 1992; Schabes and Pereira, 1992; Johnson et al., 2007b) or directly inducing a CFG as well as its probabilities (Liang et al., 2009; Tu, 2012) have not achieved as much success as experiments with other kinds of formalisms (Klein and Manning, 2004; Seginer, 2007; Ponvert et al., 2011). The assumption has been made that the space of grammars is so big that constraints must be applied to the learning process to reduce the burden of the learner (Gold, 1967; Cramer, 2007; Liang et al., 2009).

One constraint that has been applied is recursion depth (Schuler et al., 2010; Ponvert et al., 2011; Shain et al., 2016; Noji and Johnson, 2016; Jin et al., 2018), motivated by human cognitive constraints on memory capacity (Chomsky and Miller, 1963). Recursion depth can be defined in a left-corner parsing paradigm (Rosenkrantz and Lewis, 1970; Johnson-Laird, 1983). Left-corner parsers require only minimal stack memory to process left-branching and right-branching structures, but require an extra stack element to process each center embedding in a structure. For example, a left-corner parser must add a stack element for each of the first three words in the sentence, *For parts the plant built to fail was awful,* shown in Figure 1. These kinds of depth bounds in sentence processing have been used to explain the relative difficulty of center-embedded sentences compared to more right-branching paraphrases like *It was awful for the plant's parts to fail*.

However, depth-bounded grammar induction has never been compared against unbounded induction in the same system, in part because most previous depth-bounding models are built around sequence models, the complexity of which grows exponentially with the maximum allowed depth. In order to compare the effects of depth-bounding more directly, this work extends a chart-based Bayesian PCFG induction model (Johnson et al., 2007b) to include depth bounding, which allows both bounded and unbounded PCFGs to be induced from unannotated text.

Experiments reported in this paper confirm that

[1] The public repository can be found at https://github.com/lifengjin/dimi_emnlp18.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2721–2731
Brussels, Belgium, October 31 - November 4, 2018.

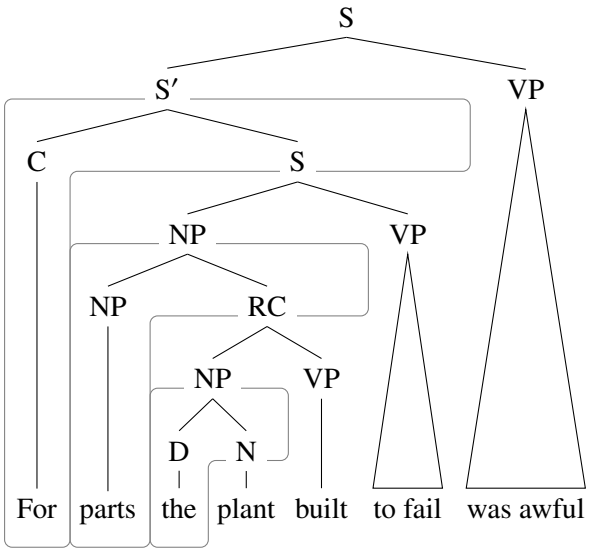

Figure 1: Stack elements after the word *the* in a left-corner parse of the sentence *For parts the plant built to fail was awful.*

depth-bounding does empirically have the effect of significantly limiting the search space of the inducer. Analyses of this model also show that the posterior samples are indicative of implicit depth limits in the data. This work also shows for the first time that it is possible to induce an accurate unbounded PCFG from raw text with no strong linguistic constraints. With a novel grammar-level marginalization in posterior inference, comparisons of the accuracy of bounded grammar induction using this model against other recent constituency grammar inducers show that this model is able to achieve state-of-the-art or competitive results on datasets in multiple languages.

2 Related work

Induction of PCFGs has long been considered a difficult problem (Carroll and Charniak, 1992; Johnson et al., 2007b; Liang et al., 2009; Tu, 2012). Lack of success for direct estimation was attributed either to a lack of correlation between the linguistic accuracy and the optimization objective (Johnson et al., 2007b), or the likelihood function or the posterior being filled with weak local optima (Smith, 2006; Liang et al., 2009). Much of this grammar induction work used strong linguistically motivated constraints or direct linguistic annotation to help the inducer eliminate some local optima. Schabes and Pereira (1992) use bracketed corpora to provide extra structural information to the inducer. Use of part-of-speech (POS) sequences in place of word strings is popular in the dependency grammar induction literature (Klein and Manning, 2002, 2004; Berg-Kirkpatrick et al., 2010; Jiang et al., 2016; Noji and Johnson, 2016). Combinatory Categorial Grammar (CCG) induction also relies on POS tags to assign basic categories to words (Bisk and Hockenmaier, 2012, 2013), among other constraints such as CCG combinators. Other linguistic constraints such as constraints of root nodes (Noji and Johnson, 2016), attachment rules (Naseem et al., 2010) or acoustic cues (Pate, 2013) have also been used in induction.

Depth-like constraints have been applied in work by Seginer (2007) and Ponvert et al. (2011) to help with the search. Both of these systems are successful in inducing phrase structure trees from only words, but only generate unlabeled constituents.

Depth-bounds are directly used by induction models in work by Noji and Johnson (2016), Shain et al. (2016) and Jin et al. (2018), and are shown to be beneficial to induction. Noji and Johnson (2016) apply depth-bounding to dependency grammar induction with POS tags. However the constituency parsing evaluation scores they report are low compared to other induction systems. The model in Shain et al. (2016) is a hierarchical sequence model instead of a PCFG. Although depth-bounding limits the search space, the sequence model has more parameters than a PCFG, therefore benefits brought by depth-bounding may be offset by this larger parameter space.

Jin et al. (2018) also apply depth-bounding to a grammar inducer and induce depth-bounded PCFGs and show that the depth-bounded grammar inducer can learn labeled PCFGs competitive with state-of-the-art grammar inducers that only produce unlabeled trees. However, because of the cognitively motivated left-corner HMM sampler used in the model, its state space grows exponentially with the maximum depth and polynomially with the number of categories. This renders the transition matrix and the trellis of the inducer too big to be practical in exploring models with higher depth limits, let alone unbounded models. By using Gibbs sampling for PCFGs (Goodman, 1998; Johnson et al., 2007b), here described as the inside-sampling algorithm, the state space of the model proposed in this work grows only polynomially with both the maximum depth and the number of categories. This allows experiments with

more complex models and also achieves a faster processing speed due to an overall smaller state space.

3 Proposed model

The model described in this paper follows Jin et al. (2018) to induce a depth-bounded PCFG by first inducing an unbounded PCFG and then deterministically deriving the parameters of a depth-bounded PCFG from it. The main difference between this model and the model in Jin et al. (2018) is that they use the bounded PCFG to derive parameters for a factored HMM sequence model, where a forward-filtering backward-sampling algorithm (Carter and Kohn, 1996) can be used in inference. In contrast, the model described in this paper transforms the unbounded PCFG into a bounded PCFG, and then uses the inside-sampling algorithm (Goodman, 1998) to sample from the posterior of the parse trees given the bounded PCFG in inference. This section first gives an overview of the model, then briefly reviews the depth-bounding algorithm for PCFGs (van Schijndel et al., 2013; Jin et al., 2018), and finally describes the inference.

As defined in Jin et al. (2018), a Chomsky normal form (CNF) unbounded PCFG is a matrix $\mathbf{G}$ of binary rule probabilities with one row for each of C parent symbols c and one column for each of C^2+W combinations of left and right child symbols a and b, which can be pairs of nonterminals or observed words from vocabulary W followed by null symbols $\perp$:

$$\mathbf{G} = \sum_{a,b,c} \mathsf{P}(c \rightarrow a\ b \mid c)\ \delta_c\ (\delta_a \otimes \delta_b)^\top \qquad (1)$$

where δ_c is a Kronecker delta (a vector with value one at index c and zeros elsewhere) and $\otimes$ is a Kronecker product (multiplying two matrices[2] of dimension $m \times n$ and $o \times p$ into a matrix of dimension $mo \times np$ composed of products of all pairs of elements in the operands). A deterministic depth-bounding transform ϕ is then applied to $\mathbf{G}$ to create a depth-bounded version $\mathbf{G}_D$. A depth-bounded grammar is composed of a set of side- and depth-specific distributions $\mathbf{G}_{s,d}$:

$$\mathbf{G}_D = \sum_{s\in\{1,2\}} \sum_{d\in\{1..D\}} \mathbf{D}_{s,d}\, \mathbf{G}_{s,d}\, \mathbf{E}_{s,d}{}^\top \qquad (2)$$

[2]or vectors in case n and p equal one

where side $s \in \{1, 2\}$ indicates left (1) or right (2) child. Categories in $\mathbf{G}_D$ are made to be side- and depth-specific using transforms $\mathbf{D}_{s,d}$ and $\mathbf{E}_{s,d}$:[3]

$$\mathbf{D}_{s,d} = \delta_s \otimes \delta_d \otimes \mathbf{I} \qquad (3a)$$

$$\mathbf{E}_{1,d} = \delta_1 \otimes \delta_d \otimes \mathbf{I} \otimes \delta_2 \otimes \delta_d \otimes \mathbf{I} \qquad (3b)$$

$$\mathbf{E}_{2,d} = \delta_1 \otimes \delta_{d+1} \otimes \mathbf{I} \otimes \delta_2 \otimes \delta_d \otimes \mathbf{I} \qquad (3c)$$

The generative story of this model is as follows. The model first generates an unbounded grammar $\mathbf{G}$ from the Dirichlet prior. Distributions over expansions $\mathsf{P}(c \rightarrow a\ b \mid c)$ of each category c in this model are drawn from a Dirichlet with symmetric parameter β:

$$\mathbf{G} \sim \text{Dirichlet}(\beta) \qquad (4)$$

Trees for sentences $1..N$ are each drawn from a PCFG given parameters $\mathbf{G}_D = \phi(\mathbf{G})$:

$$\tau_{1..N} \sim \text{PCFG}(\mathbf{G}_D) \qquad (5)$$

Each tree τ is a set $\{\tau_\epsilon, \tau_1, \tau_2, \tau_{11}, \tau_{12}, \tau_{21}, ...\}$ of category labels τ_η where $\eta \in \{1, 2\}^*$ is a Gorn address specifying a path of left or right branches from the root. Categories of every pair of left and right children $\tau_{\eta 1}, \tau_{\eta 2}$ are drawn from a multinomial defined by the grammar $\mathbf{G}_D$ and the category of the parent τ_η:

$$\tau_{\eta 1}, \tau_{\eta 2} \sim \text{Multinomial}(\delta_{\tau_\eta}{}^\top \mathbf{G}_D) \qquad (6)$$

where $\mathsf{P}_{\mathbf{G}_D}(a\ b \mid w) = \mathsf{P}_{\mathbf{G}_D}(a\ b \mid \perp) = [\![a, b{=}\perp, \perp]\!]$ for $w \in W$, and $[\![\cdot]\!]$ is an indicator function.

In inference, a Gibbs sampler can be used to iteratively draw samples from the conditional posteriors of the unbounded grammar and the parse trees. For example, at iteration t:

$$\mathbf{G}^t \sim \mathsf{P}(\mathbf{G}^t \mid \tau^{t-1}_{1..N}, \sigma_{\tau^{t-1}_{1..N}}, \beta) \qquad (7)$$

$$\tau^t_{1...N} \sim \mathsf{P}(\tau^t_{1..N} \mid \mathbf{G}^t_D, \sigma_{\tau^t_{1..N}}) \qquad (8)$$

where σ_τ denotes the terminals in τ. These distributions will be defined in Section 3.2.

3.1 Depth-bounding a PCFG

This section summarizes the depth-bounding function ϕ for PCFGs described in van Schijndel et al. (2013) and Jin et al. (2018). Depth-bounding essentially creates a set of PCFGs with depth- and side-specific categories where no tree that exceeds

[3]Note that this correctly stipulates depth increases for left children of right children.

its depth bound can be generated by the bounded grammar. Because depth increases when a left child of a right child of some parent category performs non-terminal expansion, the probability of such expansions at the maximum depth limit as well as non-depth-increasing expansions beyond the maximum depth limit must be removed from the unbounded grammar. Following van Schijndel et al. (2013) and Jin et al. (2018), this can be done by iteratively defining a side- and depth-specific containment likelihood $\mathbf{h}^{(i)}_{s,d}$ for left- or right-side siblings $s \in \{1, 2\}$ at depth $d \in \{1..D\}$ at each iteration $i \in \{1..I\}$, as a vector with one row for each nonterminal or terminal symbol (or null symbol $\perp$) in $\mathbf{G}$, containing the probability of each symbol generating a complete yield within depth d as an s-side sibling:

$$\mathbf{h}^{(0)}_{s,d} = \mathbf{0} \tag{9a}$$

$$\mathbf{h}^{(i)}_{1,d} = \begin{cases} \mathbf{G}\,(\mathbf{1} \otimes \delta_{\perp} \;+\; \mathbf{h}^{(i-1)}_{1,d} \otimes \mathbf{h}^{(i-1)}_{2,d}) & \text{if } d \leq D+1 \\ \mathbf{0} & \text{if } d > D+1 \end{cases} \tag{9b}$$

$$\mathbf{h}^{(i)}_{2,d} = \begin{cases} \delta_{\mathrm{T}} & \text{if } d = 0 \\ \mathbf{G}\,(\mathbf{1} \otimes \delta_{\perp} \;+\; \mathbf{h}^{(i-1)}_{1,d+1} \otimes \mathbf{h}^{(i-1)}_{2,d}) & \text{if } 0 < d \leq D \\ \mathbf{0} & \text{if } d > D \end{cases} \tag{9c}$$

where 'T' is a top-level category label at depth zero. Following previous work, experiments described in this paper use $I = 20$.

A depth-bounded grammar $\mathbf{G}_{s,d}$ can then be defined to be the original grammar $\mathbf{G}$ reweighted and renormalized by this containment likelihood:

$$\mathbf{G}_{1,d} = \frac{\mathbf{G}\,\mathrm{diag}(\mathbf{1} \otimes \delta_{\perp} \;+\; \mathbf{h}^{(I)}_{1,d} \otimes \mathbf{h}^{(I)}_{2,d})}{\mathbf{h}^{(I)}_{1,d}} \tag{10a}$$

$$\mathbf{G}_{2,d} = \frac{\mathbf{G}\,\mathrm{diag}(\mathbf{1} \otimes \delta_{\perp} \;+\; \mathbf{h}^{(I)}_{1,d+1} \otimes \mathbf{h}^{(I)}_{2,d})}{\mathbf{h}^{(I)}_{2,d}} \tag{10b}$$

3.2 Gibbs sampling of unbounded grammars and bounded trees

As defined above, this model samples iteratively from the conditional posteriors of $\mathsf{P}(\mathbf{G} \mid \tau_{0..N}, \sigma_{\tau_{0..N}}, \beta)$ and $\mathsf{P}(\tau_{0..N} \mid \mathbf{G}_D, \sigma_{\tau_{0..N}})$ in inference, extending the Gibbs sampling algorithm for PCFG induction introduced in Johnson et al. (2007b) to depth-bounded grammars. The below equations will omit the superscript t for the iteration number of inference for clarity.

To sample from the conditional posterior of $\mathbf{G}$, it is necessary to first sum over all rule applications in all sampled trees:

$$\mathbf{C}_D = \sum_{\tau \in \tau_{1..N}} \sum_{\tau_\eta \in \tau} \delta_{\tau_\eta}\,(\delta_{\tau_{\eta 1}} \otimes \delta_{\tau_{\eta 2}})^{\top} \tag{11}$$

then remove side- and depth-specificity from category labels:

$$\mathbf{C} = \sum_{s} \sum_{d} \mathbf{D}_{s,d}{}^{\top} \mathbf{C}_D\, \mathbf{E}_{s,d} \tag{12}$$

A side- and depth-independent grammar is then sampled from these counts, plus the pseudo-count β:

$$\mathbf{G} \sim \mathrm{Dirichlet}(\beta + \mathbf{C}) \tag{13}$$

Inside-sampling (Goodman, 1998; Johnson et al., 2007b) is then used to sample from the posterior of trees $\mathsf{P}(\tau_{0..N} \mid \mathbf{G}_D, \sigma_{\tau_{0...N}})$. Given a depth-bounded grammar and a sentence, this algorithm first constructs the inside chart $\mathbf{V} \in \mathbb{R}^{L \times L \times C}$, where L is the length of the sentence. A chart vector $\mathbf{V}_{[i,j,1..C]}$ for the span i, j where $i < j \leq L$ in some sentence $w_{1..L}$ is the likelihood $\mathsf{P}_{\mathbf{G}_D}(w_{i..j} \mid c)$ of the span for all side- and depth-specific categories c:

$$\mathbf{V}_{[i,j,1..C]} = \begin{cases} \mathbf{G}_D\,(\delta_{w_i} \otimes \delta_{\perp}) & \text{if } j{-}i = 1 \\ \sum_k \mathbf{G}_D\,(\mathbf{V}_{[i,k,1..C]} \otimes \mathbf{V}_{[k,j,1..C]}) & \text{if } j{-}i > 1 \end{cases} \tag{14}$$

Trees are sampled iteratively from the top down by first choosing a split point $k_{i,j}$ for the current span i, j such that $i < k_{i,j} < j$:

$$k_{i,j} \sim \mathrm{Mul}\left(\sum_k \delta_k\, \delta^{\top}_{c_{i,j}} \mathbf{G}_D\,(\mathbf{V}_{[i,k,1..C]} \otimes \mathbf{V}_{[k,j,1..C]})\right) \tag{15}$$

The algorithm then samples pairs of category labels $c_{i,k_{i,j}}$ and $c_{k_{i,j},j}$ adjacent at this split point $k_{i,j}$:

$$c_{i,k}, c_{k,j} \sim \mathrm{Mul}\left(\delta^{\top}_{c_{i,j}} \mathbf{G}_D\,\mathrm{diag}(\mathbf{V}_{[i,k,1..C]} \otimes \mathbf{V}_{[k,j,1..C]})\right) \tag{16}$$

Empirically the sampler spends most of its time constructing the inside chart. The model described in this paper therefore efficiently computes the inside chart using matrix multiplication, which is able to exploit GPU optimization.

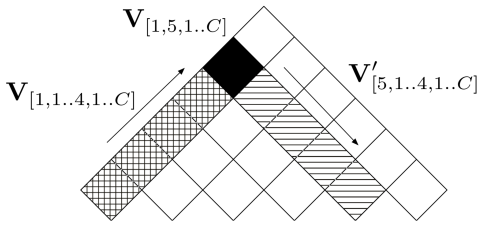

Figure 2: Example of matrix multiplication in place of looping over break points for the span (0,5). Each chart cell represents a likelihood vector for the span between i and j where i is the leftmost delimiting index of the span and j the rightmost. The arrows represent the order in which the cells are stored in the chart matrices $\mathbf{V}$ and $\mathbf{V}'$.

Efficient inside score calculation

The complexity of the inside algorithm is cubic on the length of the sentence because it has to iterate over all start points i, all end points j and all split points k of a span. For a dense PCFG with a large number of states, the explicit looping is undesirable, especially when it can be formulated as matrix multiplication. The split point loop is therefore replaced with a matrix multiplication in order to take advantage of highly optimized GPU linear algebra packages like cuBLAS and cuSPARSE, whereas previous work explores how to parse efficiently on GPUs (Johnson, 2011; Canny et al., 2013; Hall et al., 2014).

Inside likelihoods are propagated using a copy $\mathbf{V}'$ of the inside likelihood tensor $\mathbf{V}$ with the first and second indices reversed:

$$\mathbf{V}'_{[j,i,c]} = \mathbf{V}_{[i,j,c]} \tag{17}$$

This reversal allows the sum over split points $k \in \{i{+}1, ..., j{-}1\}$ to be calculated as a product of contiguous matrices, which can be efficiently implemented on a GPU:

$$\mathbf{V}_{[i,j,1..C]} = \mathbf{G}_D \, \mathrm{vec}(\mathbf{V}_{[i,i+1..j-1,1..C]}{}^{\top} \mathbf{V}'_{[j,i+1..j-1,1..C]}) \tag{18}$$

where $\mathrm{vec}(\mathbf{M})$ flattens a matrix $\mathbf{M}$ into a vector.

3.3 Posterior inference on constituents

Prior work (Johnson et al., 2007a) shows that using EM-like algorithms, which seek to maximize the likelihood of data marginalizing out the latent trees, does not yield good performance. Because trees are the main target for evaluation, it may be preferable to find the most probable tree structures given the marginal posterior of tree structures compared to finding the most probable grammar. Some recent work (McClosky and Charniak, 2015; Keith et al., 2018) explores how to use marginal distributions of tree structures from supervised parsers to create more accurate parse trees. Based on these arguments, this model performs maximum a posteriori (MAP) inference on constituents (PIoC) using approximate conditional posteriors of spans to create final parses for evaluation.

Formally, let $\sigma^{\star}_{i,j}$ be an MAP unlabeled span of words in a sentence from a corpus σ, with start point i and end point j, and $\sigma_{i,k}, \sigma_{k,j}$ its possible children. This algorithm iteratively looks for the best pair of children $\sigma^{\star}_{i,k}, \sigma^{\star}_{k,j}$ according to the posterior of the children, using all posterior samples. The spans are sentence-specific, but the below equations omit the sentence index for brevity:

$$\begin{aligned}
\sigma^{\star}_{i,k}, \sigma^{\star}_{k,j} &= \arg\max_{\sigma_{i,k},\sigma_{k,j}} \mathsf{P}(\sigma_{i,k}, \sigma_{k,j} \mid \sigma^{\star}_{i,j}, \sigma) \\
&= \arg\max_{\sigma_{i,k},\sigma_{k,j}} \int \mathsf{P}(\sigma_{i,k}, \sigma_{k,j}, \mathbf{G} \mid \sigma^{\star}_{i,j}, \sigma)\, d\mathbf{G} \\
&\approx \arg\max_{\sigma_{i,k},\sigma_{k,j}} \sum_{\hat{\mathbf{G}} \sim \mathsf{P}(\mathbf{G}|\sigma)} \mathsf{P}(\sigma_{i,k}, \sigma_{k,j}, \hat{\mathbf{G}} \mid \sigma^{\star}_{i,j}, \sigma)
\end{aligned} \tag{19}$$

where σ is the training corpus. Starting from the whole sentence $\sigma_{0,N}$, this algorithm finds the best children for a span from the Monte Carlo estimation of the marginal posterior distribution of children for the span, and then continues to split the found children spans. Because samples from different runs at different iterations can be used to approximate the span posteriors, the process marginalizes out sampled grammars, whole-sentence parse trees and constituent labels to only consider split points for spans. In terms of input and output, the PIoC algorithm takes in posterior samples of trees for a sentence, and outputs an unlabeled binary-branching tree.

There are a few benefits of doing posterior inference on constituents. First, the distribution $\mathsf{P}(\sigma_{i,k}, \sigma_{k,j} \mid \sigma^{\star}_{i,j}, \sigma)$ quantifies how much uncertainty there is in splitting a span $\sigma_{i,j}$ at all possible k's. One way of using this uncertainty information is to merge spans where uncertainty is high, effectively weakening or removing the constraint of binary-branching from the grammar inducer. Second, this algorithm produces trees that may not be seen in the samples, potentially helping aggregate evidence across different iterations within a run and across runs. Third, the multimodal na-

ture of the joint posterior of grammars and trees often makes the sampler get stuck at local modes, but doing MAP on constituents may allow information about trees from different modes to come together. If different grammars all consider certain children for a span to be highly likely, then these children should be in the final parse output. Finally, it is a nonparametric way of doing model selection. As will be shown, model selection relies on the log likelihood of the data, but the log likelihood of the data is only weakly correlated with parsing accuracy. Performing PIoC with multiple runs can increase accuracy without depending too heavily on log likelihood for model selection.

4 Model analysis and evaluation

The model described above has hyperparameters for maximum depth D, number of categories C and the symmetric Dirichlet prior β. Following Jin et al. (2018), this evaluation uses the first half of the WSJ20 corpus as the development set (WSJ20dev) for all experiments. However instead of using the development set only to set the hyperparameters of the model, this evaluation also uses it to explore interactions among parsing accuracy, model fit, depth limit and category domain. The first set of experiments explores various settings of D in the hope of acquiring a better picture of how depth-bounding affects the inducer. The second set of experiments uses the value of D tuned in the first experiments, and does PIoC on different sets of samples to examine the effect it has on parse quality. Optimal parameter values from these first two experiments are then applied in experiments on English (The Penn Treebank; Marcus et al., 1993), Chinese (The Chinese Treebank 5.0; Xia et al., 2000) and German (NEGRA 2.0; Skut et al., 1998) data to show how the model performs compared with competing systems.

Each run in evaluation uses one sample of parse trees from the posterior samples after convergence. Preliminary experiments show that the samples after convergence are very similar within a run and their parsing accuracies differ very little. This evaluation follows Seginer (2007) by running unlabeled PARSEVAL on parse trees collected from each run. Punctuation is retained in the raw text in induction, and removed in evaluation, also following Seginer (2007).

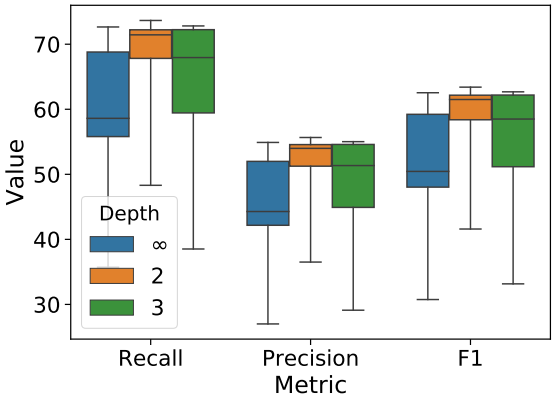

Figure 3: PARSEVAL scores for runs with different depth limits. The difference of all PARSEVAL scores between depth ∞ and depth 2 is significant (p=0.017, Student's t test).

4.1 Analysis of model behavior

The first experiment explores the effects of depth-bounding on linguistic grammar quality. The hypothesis is that depth-bounding limits the search space of possible grammars, so the inducer will be less likely to find low-quality local optima where cognitively implausible parse trees are assigned non-zero probabilities, because such local optima would be removed from the posterior by limiting the maximum depth of parse trees to a small number d.

The effect of depth-bounding

Figure 3 shows the effect of depth bounding using 60 data points of unlabeled PARSEVAL scores from 20 different runs for each of three different depth bounds: 2, 3, and ∞ (unbounded). The range of possible parsing accuracy scores is very wide, as mapped out by the runs. Although the unbounded model is able to reach the performance upper bound seen from the figure, most of the time its results are in the middle of the range. By bounding the maximum depth to 2, the sampler is able to stay in the region of high parsing accuracy. This may be because the majority of the modes in the region of low parsing accuracy require higher depth limits, and humans who produce the sentences do not have access to those higher depth limits. The difference between depth ∞ and depth 2 is significant (p=0.017, Student's t test), showing that depth-bounding does have a positive effect on the linguistic grammar quality of the induced grammars. Data from depth 3 also shows a positive trend of inducing better grammars than

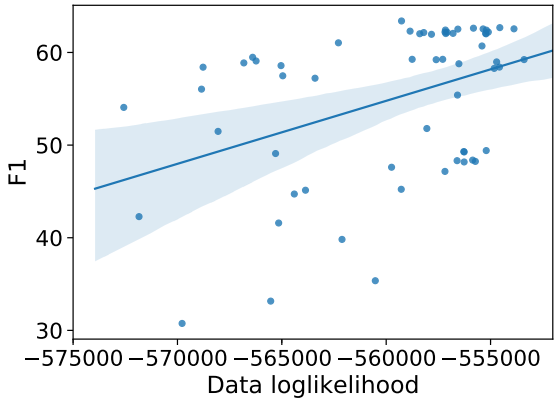

Figure 4: The correlation between data likelihood and parsing accuracy of all 60 runs. Calculations show that there is a significant ($p = 0.007$) positive correlation (Pearson's r=0.39) between data likelihood and parsing accuracy at convergence for our model.

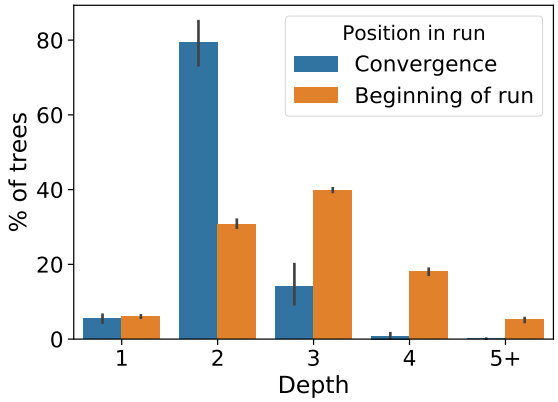

Figure 5: The usage of different depths for parse trees in the samples from 20 runs with the unbounded grammar.

unbounded.

A purely right-branching baseline achieves an F1 score of 48 on the WSJ20 development dataset. A majority of induction runs perform better than this baseline, which indicates that the PCFG induction model with the inside-sampling algorithm is able to find good solutions, most of the time much better than the right-branching baseline. This is especially interesting when the grammar is unbounded with almost no other constraint, which had previously been shown to converge to weak local optima.

Correlation of model fit and parsing accuracy

Model fit, or data likelihood, has been reported not to be correlated or to be correlated only weakly with parsing accuracy for some unsupervised grammar induction models (Smith, 2006; Johnson et al., 2007b; Liang et al., 2009) when the model has converged to a local maximum. Figure 4 shows the correlation between data likelihood and parsing accuracy at convergence for all the runs. There is a significant ($p = 0.007$) positive correlation (Pearson's r=0.39) between data likelihood and parsing accuracy at convergence for our model. This indicates that although noisy and unreliable, the data likelihood can be used as a metric to do preliminary model selection.

The bounded unbounded PCFG

We also examine the distribution of tree depths in unbounded runs. For a run, we compute the percentage of parse trees with a certain depth, and then examine how these percentages vary across different runs. Theoretically the possible maximum depth of a parse for a sentence is the sentence length divided by 2. For example, a 20-word sentence can have a parse of depth 10 because at least two words are needed to create a new depth with a center embedded phrase, but under most PCFGs this maximally center embedded configuration is not very likely. Figure 5 shows the percentage of tree depths from samples in the beginning of each unbounded run and at convergence. It shows that at the beginning of the sampling process with a random model sampled from the prior, the distribution of parse tree depths seems to be centered around depth 2 and 3, with non-negligible probability mass at other depth levels. At convergence, the distribution of parse tree depths is very peaked with a large portion of the probability mass concentrated at depth 2. Given that an unbounded PCFG has no constraint on depth, this convergence of the marginal posterior distribution of parse tree depth shows that the depth limit seems to be a natural tendency in the data, rather than an arbitrary preference of corpus annotators.

4.2 Posterior uncertainty of constituents

Experiments were also conducted to determine whether posterior inference on constituents (PIoC) has any effect on parsing accuracy. These experiments use 10 runs on WSJ20dev with depth 2 that have the highest log-likelihoods for exploration. In this data, some spans have a strikingly higher degree of uncertainty than other spans. For example, the posterior probability of splitting the phrase *the old story*, into *the old* and *story* is 0.55, and the

System	Rec	Prec	F1
Best	73.65	55.66	63.40
Best w/ PIoC	73.59	56.41	63.87
All w/ PIoC	72.99	59.21	65.38
All w/ PIoC w/o best	73.00	59.06	65.29

Table 1: Development results for different systems using posterior inference on constituents (PIoC).

probability of splitting it into *the* and *old story* is 0.45. Some other spans like *use old tools* have virtually no uncertainty in how the inducer evaluates the splits. Many such spans with high uncertainty are noun phrases, which are not annotated with subconstituents in the Penn Treebank annotation. The parser can therefore avoid precision losses by not splitting constituents with 3 or 4 words if there is large uncertainty in this posterior.[4] This experiment only merges spans that would cover 3 or 4 words and leave merging spans with larger coverage to future work.

Table 1 shows parsing results on the WSJ20dev dataset. The *Best* result is from an arbitrary sample at convergence of the oracle best run. The *Best with PIoC* is the same run, but with PIoC to aggregate 100 posterior samples at convergence. *All with PIoC* uses 100 posterior samples from all of the 10 chosen runs, and finally *All with PIoC without best* excludes the best run in PIoC calculation.

There is almost a point of gain in precision going from *Best* to *Best with PIoC* with virtually no recall loss, showing that the posterior uncertainty is helpful in flattening binary trees. As more samples from the posterior are collected, as shown in *All with PIoC without best*, the precision gain is even more substantial. This shows that with PIoC there is no need to know which sample from which run is the best. Model selection in this case is only needed to weed out the runs with very low likelihood.

4.3 Multilingual PARSEVAL

A final set of experiments compare the proposed model with several state-of-the-art constituency grammar induction systems on three different languages. The competing systems are CCL (Seginer, 2007)[5] and UPPARSE (Ponvert et al., 2011).[6] We also include the published results of DB-PCFG (Jin et al., 2018) on English for comparison.[7] The corpora used are the WSJ20test dataset used in Jin et al. (2018), the CTB20 (sentences with 20 words or fewer from the Chinese Treebank) and NEGRA20 (sentences with 20 words or fewer from the German NEGRA Treebank) datasets used in Seginer (2007). All systems are trained and evaluated on the same datasets to ensure fair and direct comparison. Five different induction runs were run on each dataset with the same hyperparameters D=2, C=15, β=0.2 as tuned on the development set, and three runs with the highest likelihood at convergence were chosen for comparison with other models. Parse trees were then calculated using PIoC as previously described, removing punctuation to calculate the unlabeled PARSEVAL scores with EVALB. Multiple runs of CCL and UPPARSE on the same data yield the same results.

Table 2 shows the unlabeled PARSEVAL scores for the competing systems. The model described in this paper shows strong performance in all languages. On English and Chinese, this model achieves the new state-of-the-art recall and F1 numbers. On German, this model also achieves the best recall scores among all models, showing that more constituents found in the gold annotation are discovered. It is worth noting that the CCL and UPPARSE models do take advantage of additional linguistic constraints, e.g. using punctuation as delimiters of constituents. Experiments described in this paper show that this system can perform better than or competitive with these existing models without similar heuristics and constraints.

The model described in this paper performs relatively poorly on precision due to the fact that trees produced by this system are mostly binary-branching with some constituents flattened by PIoC. This issue is most evident on Negra, where fully binary-branching trees have nearly twice as many constituents as are annotated in gold. This puts any system that produces binary-branching trees under a precision celling of 0.51, and F1 celling of 0.675.

5 Conclusion

Experiments in this work confirm that depth-bounding does empirically have the effect of limiting the search space of an unsupervised PCFG in-

[4] I.e. if the difference between the first and the second highest posterior probabilities is smaller than 0.3.

[5] https://github.com/DrDub/cclparser

[6] https://github.com/eponvert/upparse

[7] We are not able to run DB-PCFG on the other languages due to its substantial resource requirements.

System	WSJ20test			CTB20			NEGRA20		
	Rec	Prec	F1	Rec	Prec	F1	Rec	Prec	F1
CCL	61.7	**60.1**	60.9	35.3	39.2	37.1	44.4	27.2	33.7
UPPARSE	40.5	47.8	43.9	33.8	**44.0**	38.2	55.5	**41.9**	**47.7**
DB-PCFG	70.5	53.0	60.5	-	-	-	-	-	-
this work	**73.1**	55.6	**63.1**	**43.8**	35.1	**38.9**	**59.1**	31.2	40.8

Table 2: PARSEVAL scores for different constituency grammar induction systems.

ducer. Analysis of a depth-bounded model demonstrates desirable engineering properties, including a significant correlation between parsing accuracy and data likelihood, and interesting linguistic properties such as implicit bounding for unbounded grammars. This paper also introduces the Posterior Inference on Constituents technique for model selection and shows for the first time that it is possible to accurately induce a PCFG with no strong universal linguistic constraints. Comparisons of the proposed model with other state-of-the-art constituency grammar inducers show that this model is able to achieve state-of-the-art or competitive results on datasets in multiple languages.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments. Computations for this project were partly run on the Ohio Supercomputer Center (1987). This research was funded by the Defense Advanced Research Projects Agency award HR0011-15-2-0022. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 582–590.

Yonatan Bisk and Julia Hockenmaier. 2012. Simple Robust Grammar Induction with Combinatory Categorial Grammars. *AAAI*, pages 1643–1649.

Yonatan Bisk and Julia Hockenmaier. 2013. An HDP Model for Inducing Combinatory Categorial Grammars. In *Transactions Of The Association For Computational Linguistics*, pages 75–88.

John Canny, David Hall, and Dan Klein. 2013. A multi-Teraflop Constituency Parser using GPUs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1898–1907.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. *Working Notes of the Workshop on Statistically-Based NLP Techniques*, (March):1–13.

C. K. Carter and R. Kohn. 1996. Markov chain Monte Carlo in conditionally Gaussian state space models. *Biometrika*, 83(3):589–601.

Noam Chomsky and George A Miller. 1963. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology*, pages 269–321. Wiley, New York, NY.

Bart Cramer. 2007. Limitations of current grammar induction algorithms. *Proceedings of the 45th Annual Meeting of the ACL: Student Research Workshop on - ACL '07*, (June):43.

Mark E. Gold. 1967. Language Identification in the Limit. *Information and Control*, (10):447–474.

Joshua Goodman. 1998. Parsing Inside-Out.

David Hall, Taylor Berg-Kirkpatrick, Canny John, and Dan Klein. 2014. Sparser, Better, Faster GPU Parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2007, page 208217.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2016. Unsupervised neural dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 61503248, pages 763–771.

Lifeng Jin, William Schuler, Finale Doshi-Velez, Timothy A Miller, and Lane Schwartz. 2018. Unsupervised Grammar Induction with Depth-bounded PCFG. *Transactions of the Association for Computational Linguistics*.

Mark Johnson. 2011. Parsing in parallel on multiple cores and gpus. In *In Proceedings of Australasian Language Technology Association Workshop*, pages 29–37.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007a. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems*, volume 19, page 641.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007b. Bayesian Inference for PCFGs via Markov chain Monte Carlo. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference (ACL)*, pages 139–146.

Philip N Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press, Cambridge, MA, USA.

Katherine A Keith, Su Lin Blodgett, and Brendan O'Connor. 2018. Monte Carlo Syntax Marginals for Exploring and Using Dependency Parses. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 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 Annual Meeting on Association for Computational Linguistics*, volume 1, pages 478–485.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Probabilistic Grammars and Hierarchical Dirichlet Processes. *The Handbook of Applied Bayesian Analysis*.

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.

David McClosky and Eugene Charniak. 2015. Syntactic Parse Fusion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, September, pages 1360–1366.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using Universal Linguistic Knowledge to Guide Grammar Induction. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, (October):1234–1244.

Hiroshi Noji and Mark Johnson. 2016. Using Left-corner Parsing to Encode Universal Structural Constraints in Grammar Induction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 33–43.

The Ohio Supercomputer Center. 1987. Ohio Supercomputer Center. \url{http://osc.edu/ark:/19495/f5s1ph73}.

John K Pate. 2013. Unsupervised Dependency Parsing with Acoustic Cues. *Transactions of the Association for Computational Linguistics*, 1:63–74.

Elias Ponvert, Jason Baldridge, and Katrin Erk. 2011. Simple unsupervised grammar induction from raw text with cascaded finite state models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1077–1086.

D J Rosenkrantz and P M Lewis. 1970. Deterministic left corner parsing. In *11th Annual Symposium on Switching and Automata Theory (swat 1970)*, pages 139–152.

Fernando Schabes and Yves Pereira. 1992. Inside-Outside Reestimation From Partially Bracketed Corpora. *Proceedings of the 30th annual meeting on Association for Computational Linguistics*, pages 128–135.

Marten van Schijndel, Andy Exley, and William Schuler. 2013. A Model of language processing as hierarchic sequential prediction. *Topics in Cognitive Science*, 5(3):522–540.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2010. Broad-coverage parsing using human-Like memory constraints. *Computational Linguistics*, 36(1):1–30.

Yoav Seginer. 2007. Fast Unsupervised Incremental Parsing. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 384–391.

Cory Shain, William Bryce, Lifeng Jin, Victoria Krakovna, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2016. Memory-bounded left-corner unsupervised grammar induction on child-directed input. In *Proceedings of the International Conference on Computational Linguistics*, pages 964–975.

Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. In *Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation.*, page 7.

Noah Ashton Smith. 2006. Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text. *PhD Thesis*, pages 1–228.

Kewei Tu. 2012. *Unsupervised learning of probabilistic grammars*. Ph.D. thesis.

Fei Xia, Martha Palmer, Nianwen Xue, Mary Ellen Ocurowski, John Kovarik, Fu-Dong Chiou, Shizhe Huang, Tony Kroch, and Mitch Marcus. 2000. Developing Guidelines and Ensuring Consistency for

Chinese Text Annotation. *Proceedings of the Second Language Resources and Evaluation Conference*, (Section 3).