Unsupervised Grammar Induction with Depth-bounded PCFG

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 has been recent interest in applying cognitively- or empirically-motivated bounds on recursion depth to limit the search space of grammar induction models (Ponvert et al., 2011; Noji and Johnson, 2016; Shain et al., 2016). This work extends this depth-bounding approach to probabilistic context-free grammar induction (DB-PCFG), which has a smaller parameter space than hierarchical sequence models, and therefore more fully exploits the space reductions of depth-bounding. Results for this model on grammar acquisition from transcribed child-directed speech and newswire text exceed or are competitive with those of other models when evaluated on parse accuracy. Moreover, grammars acquired from this model demonstrate a consistent use of category labels, something which has not been demonstrated by other acquisition models.

1 Introduction

Grammar acquisition or grammar induction (Carroll and Charniak, 1992) has been of interest to linguists and cognitive scientists for decades. This task is interesting because a well-performing acquisition model can serve as a good baseline for examining factors of grounding (Zettlemoyer and Collins, 2005; Kwiatkowski et al., 2010), or as a piece of evidence (Clark, 2001; Zuidema, 2003) about the Distributional Hypothesis (Harris, 1954) against the poverty of the stimulus (Chomsky, 1965). Unfortunately, previous attempts at inducing unbounded context-free grammars (Johnson et al., 2007; Liang et al., 2009) converged to weak modes of a very multimodal distribution of grammars. There has been recent interest in applying cognitively- or empirically-motivated bounds on recursion depth to limit the search space of grammar induction models (Ponvert et al., 2011; Noji and Johnson, 2016; Shain et al., 2016). Ponvert et al. (2011) and Shain et al. (2016) in particular report benefits for depth bounds on grammar acquisition using hierarchical sequence models, but either without the capacity to learn full grammar rules (e.g. that a noun phrase may consist of a noun phrase followed by a prepositional phrase), or with a very large parameter space that may offset the gains of depth-bounding. This work extends the depth-bounding approach to directly induce probabilistic context-free grammars,[1] which have a smaller parameter space than hierarchical sequence models, and therefore arguably make better use of the space reductions of depth-bounding. This approach employs a procedure for deriving a sequence model from a PCFG (van Schijndel et al., 2013), developed in the context of a supervised learning model, and adapts it to an unsupervised setting.

Results for this model on grammar acquisition from transcribed child-directed speech and newswire text exceed or are competitive with those of other models when evaluated on parse accuracy. Moreover, grammars acquired from this model demonstrate a consistent use of category labels, as shown in a noun phrase discovery task, something which has not been demonstrated by other acquisition models.

[1] https://github.com/lifengjin/db-pcfg

Transactions of the Association for Computational Linguistics, vol. 6, pp. 211–224, 2018. Action Editor: Xavier Carreras.
Submission batch: 9/2017; Revision batch: 12/2017; Published 4/2018.

2 Related work

This paper describes a Bayesian Dirichlet model of depth-bounded probabilistic context-free grammar (PCFG) induction. Bayesian Dirichlet models have been applied to the related area of latent variable PCFG induction (Johnson et al., 2007; Liang et al., 2009), in which subtypes of categories like noun phrases and verb phrases are induced on a given tree structure. The model described in this paper is given only words and not only induces categories for constituents but also tree structures.

There are a wide variety of approaches to grammar induction outside the Bayesian modeling paradigm. The CCL system (Seginer, 2007a) uses deterministic scoring systems to generate bracketed output of raw text. UPPARSE (Ponvert et al., 2011) uses a cascade of HMM chunkers to produce syntactic structures. BMMM+DMV (Christodoulopoulos et al., 2012) combines an unsupervised part-of-speech (POS) tagger BMMM and an unsupervised dependency grammar inducer DMV (Klein and Manning, 2004). The BMMM+DMV system alternates between phases of inducing POS tags and inducing dependency structures. A large amount work (Klein and Manning, 2002; Klein and Manning, 2004; Bod, 2006; Berg-kirkpatrick et al., 2010; Gillenwater et al., 2011; Headden et al., 2009; Bisk and Hockenmaier, 2013; Scicluna and de la Higuera, 2014; Jiang et al., 2016; Han et al., 2017) has been on grammar induction with input annotated with POS tags, mostly for dependency grammar induction. Although POS tags can also be induced, this separate induction has been criticized (Pate and Johnson, 2016) for missing an opportunity to leverage information learned in grammar induction to estimate POS tags. Moreover, most of these models explore a search space that includes syntactic analyses that may be extensively center embedded and therefore are unlikely to be produced by human speakers. Unlike most of these approaches, the model described in this paper uses cognitively motivated bounds on the depth of human recursive processing to constrain its search of possible trees for input sentences.

Some previous work uses depth bounds in the form of sequence models (Ponvert et al., 2011; Shain et al., 2016), but these either do not produce complete phrase structure grammars (Ponvert et al., 2011) or do so at the expense of large parameter sets (Shain et al., 2016). Other work implements depth bounds on left-corner configurations of dependency grammars (Noji and Johnson, 2016), but the use of a dependency grammar makes the system impractical for addressing questions of how category types such as noun phrases may be learned. Unlike these, the model described in this paper induces a PCFG directly and then bounds it with a model-to-model transform, which yields a smaller space of learnable parameters and directly models the acquisition of category types as labels.

Some induction models learn semantic grammars from text annotated with semantic predicates (Zettlemoyer and Collins, 2005; Kwiatkowski et al., 2012). There is evidence humans use semantic bootstrapping during grammar acquisition (Naigles, 1990), but these models typically rely on a set of pre-defined universals, such as combinators (Steedman, 2000), which simplify the induction task. In order to help address the question of whether such universals are indeed necessary for grammar induction, the model described in this paper does not assume any strong universals except independently motivated limits on working memory.

3 Background

Like Noji and Johnson (2016) and Shain et al. (2016), the model described in this paper defines bounding depth in terms of memory elements required in a left-corner parse. A left-corner parser (Rosenkrantz and Lewis, 1970; Johnson-Laird, 1983; Abney and Johnson, 1991; Resnik, 1992) uses a stack of memory elements to store derivation fragments during incremental processing. Each derivation fragment represents a disjoint connected component of phrase structure a/b consisting of a top sign a lacking a bottom sign b yet to come. For example, Figure 1 shows the derivation fragments in a traversal of a phrase structure tree for the sentence *The cart the horse the man bought pulled broke.* Immediately before processing the word *man*, the traversal has recognized three fragments of tree structure: two from category NP to category RC (covering *the cart* and *the horse*) and one from category NP to category N (cover-

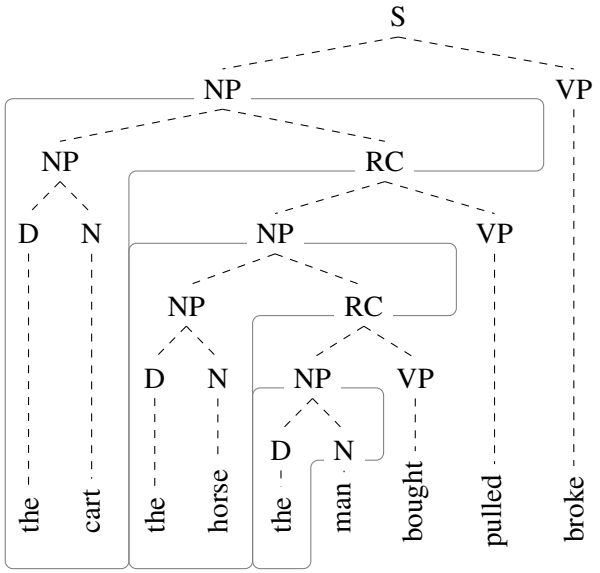

Figure 1: Derivation fragments before the word *man* in a left-corner traversal of the sentence *The cart the horse the man bought pulled broke.*

ing *the*). Derivation fragments at every time step are numbered top-down by depth d to a maximum depth of D. A left-corner parser requires more derivation fragments — and thus more memory — to process center-embedded constructions than to process left- or right-embedded constructions, consistent with observations that center embedding is more difficult for humans to process (Chomsky and Miller, 1963; Miller and Isard, 1964). Grammar acquisition models (Noji and Johnson, 2016; Shain et al., 2016) then restrict this memory to some low bound, e.g. two derivation fragments.

For sequences of observed word tokens w_t for time steps $t \in \{1..T\}$, sequence models like Ponvert et al. (2011) and Shain et al. (2016) hypothesize sequences of hidden states q_t. Models like Shain et al. (2016) implement bounded grammar rules as depth bounds on a hierarchical sequence model implementation of a left-corner parser, using random variables within each hidden state q_t for:

1. preterminal labels p_t and labels of top and bottom signs, a_t^d and b_t^d, of derivation fragments at each depth level d (which correspond to left and right children in tree structure), and

2. Boolean variables for decisions to 'fork out' f_t and 'join in' j_t derivation fragments (in Johnson-Laird (1983) terms, to *shift* with or without *match* and to *predict* with or without *match*).

Probabilities from these distributions are then multiplied together to define a transition model $\mathbf{M}$ over hidden states:

$$\begin{aligned}
\mathbf{M}_{[q_{t-1},q_t]} &= \mathsf{P}(q_t \mid q_{t-1}) && \text{(1a)}\\
&\stackrel{\text{def}}{=} \mathsf{P}(f_t\, p_t\, j_t\, a_t^{1..D}\, b_t^{1..D} \mid q_{t-1}) && \text{(1b)}\\
&= \mathsf{P}(f_t \mid q_{t-1})\\
&\quad \cdot \mathsf{P}(p_t \mid q_{t-1}\, f_t)\\
&\quad \cdot \mathsf{P}(j_t \mid q_{t-1}\, f_t\, p_t)\\
&\quad \cdot \mathsf{P}(a_t^{1..D} \mid q_{t-1}\, f_t\, p_t\, j_t)\\
&\quad \cdot \mathsf{P}(b_t^{1..D} \mid q_{t-1}\, f_t\, p_t\, j_t\, a_t^{1..D}). && \text{(1c)}
\end{aligned}$$

For example, just after the word *horse* is recognized in Figure 1, the parser store contains two derivation fragments yielding *the cart* and *the horse*, both with top category NP and bottom category RC. The parser then decides to fork out the next word *the* based on the bottom category RC of the last derivation fragment on the store. Then the parser generates a preterminal category D for this word based on this fork decision and the bottom category of the last derivation fragment on the store. Then the parser decides *not* to join the resulting D directly to the RC above it, based on these fork and preterminal decisions and the bottom category of the store. Finally the parser generates NP and N as the top and bottom categories of a new derivation fragment yielding just the new word *the* based on all these previous decisions, resulting in the store state shown in the figure.

The model over the fork decision (*shift* with or without *match*) is defined in terms of a depth-specific sub-model $\theta_{\mathrm{F},\bar{d}}$, where $\perp$ is an empty derivation fragment and $\bar{d}$ is the depth of the deepest non-empty derivation fragment at time step $t-1$:

$$\mathsf{P}(f_t \mid q_{t-1}) \stackrel{\text{def}}{=} \mathsf{P}_{\theta_{\mathrm{F},\bar{d}}}(f_t \mid b_{t-1}^{\bar{d}});\quad \bar{d} = \max_d \{b_{t-1}^d \neq \perp\} \quad (2)$$

The model over the preterminal category label is then conditioned on this fork decision. When there is no fork, the preterminal category label is deterministically linked to the category label of the bottom sign of the deepest derivation fragment at the previous time step (using $[\![\phi]\!]$ as a deterministic indicator function, equal to one when ϕ is true and zero

otherwise). When there is a fork, the preterminal category label is defined in terms of a depth-specific sub-model $\theta_{\mathrm{P},\bar{d}}$:[2]

$$\mathsf{P}(p_t \mid q_{t-1}\ f_t) \stackrel{\text{def}}{=} \begin{cases} [\![p_t{=}b^{\bar{d}}_{t-1}]\!] & \text{if } f_t = 0 \\ \mathsf{P}_{\theta_{\mathrm{P},\bar{d}}}(p_t \mid b^{\bar{d}}_{t-1}) & \text{if } f_t = 1. \end{cases} \tag{3}$$

The model over the join decision (*predict* with or without *match*) is also defined in terms of a depth-specific sub-model $\theta_{\mathrm{J},\bar{d}}$ with parameters depending on the outcome of the fork decision:[3]

$$\mathsf{P}(j_t \mid q_{t-1}\ f_t\ p_t) \stackrel{\text{def}}{=} \begin{cases} \mathsf{P}_{\theta_{\mathrm{J},\bar{d}}}(j_t \mid b^{\bar{d}-1}_{t-1}\ a^{\bar{d}}_{t-1}) & \text{if } f_t{=}0 \\ \mathsf{P}_{\theta_{\mathrm{J},\bar{d}+1}}(j_t \mid b^{\bar{d}}_{t-1}\ p_t) & \text{if } f_t{=}1. \end{cases} \tag{4}$$

Decisions about the top categories of derivation fragments $a^{1..D}_t$ (which correspond to left siblings in tree structures) are decomposed into fork- and join-specific cases. When there is a join, the top category of the deepest derivation fragment deterministically depends on the corresponding value at the previous time step. When there is no join, the top category is defined in terms of a depth-specific sub-model:[4]

$$\mathsf{P}_{\theta_{\mathrm{A}}}(a^{1..D}_t \mid q_{t-1}\ f_t\ p_t\ j_t) \stackrel{\text{def}}{=}$$

$$\begin{cases} \phi_{\bar{d}-2} \cdot [\![a^{\bar{d}-1}_t = a^{\bar{d}-1}_{t-1}]\!] & \cdot \psi_{\bar{d}+0} \quad \text{if } f_t, j_t = 0, 1 \\ \phi_{\bar{d}-1} \cdot \mathsf{P}_{\theta_{\mathrm{A},\bar{d}}}(a^{\bar{d}}_t \mid b^{\bar{d}-1}_{t-1}\ a^{\bar{d}}_{t-1}) & \cdot \psi_{\bar{d}+1} \quad \text{if } f_t, j_t = 0, 0 \\ \phi_{\bar{d}-1} \cdot [\![a^{\bar{d}}_t = a^{\bar{d}}_{t-1}]\!] & \cdot \psi_{\bar{d}+1} \quad \text{if } f_t, j_t = 1, 1 \\ \phi_{\bar{d}-0} \cdot \mathsf{P}_{\theta_{\mathrm{A},\bar{d}+1}}(a^{\bar{d}+1}_t \mid b^{\bar{d}}_{t-1}\ p_t) & \cdot \psi_{\bar{d}+2} \quad \text{if } f_t, j_t = 1, 0. \end{cases} \tag{5}$$

Decisions about the bottom categories $b^{1..D}_t$ (which correspond to right children in tree structures) also depend on the outcome of the fork and join variables, but are defined in terms of a side- and depth-specific sub-model in every case:[5]

$$\mathsf{P}_{\theta_{\mathrm{B}}}(b^{1..D}_t \mid q_{t-1}\ f_t\ p_t\ j_t\ a^{1..D}_t) \stackrel{\text{def}}{=}$$

$$\begin{cases} \phi_{\bar{d}-2} \cdot \mathsf{P}_{\theta_{\mathrm{B,R},\bar{d}-1}}(b^{\bar{d}-1}_t \mid b^{\bar{d}-1}_{t-1}\ a^{\bar{d}}_{t-1}) \cdot \psi_{\bar{d}+0} & \text{if } f_t, j_t = 0, 1 \\ \phi_{\bar{d}-1} \cdot \mathsf{P}_{\theta_{\mathrm{B,L},\bar{d}}}(b^{\bar{d}}_t \mid a^{\bar{d}}_t\ a^{\bar{d}}_{t-1}) \cdot \psi_{\bar{d}+1} & \text{if } f_t, j_t = 0, 0 \\ \phi_{\bar{d}-1} \cdot \mathsf{P}_{\theta_{\mathrm{B,R},\bar{d}}}(b^{\bar{d}}_t \mid b^{\bar{d}}_{t-1}\ p_t) \cdot \psi_{\bar{d}+1} & \text{if } f_t, j_t = 1, 1 \\ \phi_{\bar{d}-0} \cdot \mathsf{P}_{\theta_{\mathrm{B,L},\bar{d}+1}}(b^{\bar{d}+1}_t \mid a^{\bar{d}+1}_t\ p_t) \cdot \psi_{\bar{d}+2} & \text{if } f_t, j_t = 1, 0. \end{cases} \tag{6}$$

[2] Here, again, $\bar{d} = \max_d\{b^d_{t-1} \neq \perp\}$.

[3] Again, $\bar{d} = \max_d\{b^d_{t-1} \neq \perp\}$.

[4] Here $\phi_{\bar{d}} = [\![a^{1..\bar{d}}_t = a^{1..\bar{d}}_{t-1}]\!]$, $\psi_{\bar{d}} = [\![a^{\bar{d}+1..D}_t = \perp]\!]$, and again, $\bar{d} = \max_d\{b^d_{t-1} \neq \perp\}$.

[5] Here $\phi_{\bar{d}} = [\![b^{1..\bar{d}}_t = b^{1..\bar{d}}_{t-1}]\!]$, $\psi_{\bar{d}} = [\![b^{\bar{d}+1..D}_t = \perp]\!]$, and again, $\bar{d} = \max_d\{b^d_{t-1} \neq \perp\}$.

In a sequence model inducer like Shain et al. (2016), these depth-specific models are assumed to be independent of each other and fit with a Gibbs sampler, backward sampling hidden variable sequences from forward distributions using this compiled transition model $\mathbf{M}$ (Carter and Kohn, 1996), then counting individual sub-model outcomes from sampled hidden variable sequences, then resampling each sub-model using these counts with Dirichlet priors over a, b, and p models and Beta priors over f and j models, then re-compiling these resampled models into a new $\mathbf{M}$.

However, note that with K category labels this model contains $DK^2 + 3DK^3$ separate parameters for preterminal categories and top and bottom categories of derivation fragments at every depth level, each of which can be independently learned by the Gibbs sampler. Although this allows the hierarchical sequence model to learn grammars that are more expressive than PCFGs, the search space is several times larger than the K^3 space of PCFG nonterminal expansions. The model described in this paper instead induces a PCFG and derives sequence model distributions from the PCFG, which has fewer parameters, and thus strictly reduces the search space of the model.

4 The DB-PCFG Model

The depth-bounded probabilistic context-free grammar (DB-PCFG) model described in this paper directly induces a PCFG and then deterministically derives the parameters of a probabilistic left-corner parser from this single source. This derivation is based on an existing derivation of probabilistic left-corner parser models from PCFGs (van Schijndel et al., 2013), which was developed in a supervised parsing model, adapted here to run more efficiently within a larger unsupervised grammar induction model.[6]

A PCFG can be defined in Chomsky normal form as a matrix $\mathbf{G}$ of binary rule probabilities with one row for each of K parent symbols c and one column for each of $K^2{+}W$ combinations of left and

[6] More specifically, the derivation differs from that of van Schijndel et al. (2013) in that it removes terminal symbols from conditional dependencies of models over fork and join decisions and top and bottom category labels, substantially reducing the size of the derived model that must be run during induction.

right child symbols a and b, which can be pairs of nonterminals or observed words from vocabulary W followed by null symbols $\perp$:[7]

$$\mathbf{G} = \sum_{a,b,c} \mathsf{P}(c \rightarrow a\ b \mid c)\ \delta_c\ (\delta_a \otimes \delta_b)^\top. \quad (7)$$

A depth-bounded grammar is a set of side- and depth-specific distributions:

$$\mathbf{G}_D = \{\mathbf{G}_{s,d} \mid s \in \{\mathrm{L}, \mathrm{R}\}, d \in \{1..D\}\}. \quad (8)$$

The posterior probability of a depth-bounded model $\mathbf{G}_D$ given a corpus (sequence) of words $w_{1..T}$ is proportional to the product of a likelihood and a prior:

$$\mathsf{P}(\mathbf{G}_D \mid w_{1..T}) \propto \mathsf{P}(w_{1..T} \mid \mathbf{G}_D) \cdot \mathsf{P}(\mathbf{G}_D). \quad (9)$$

The likelihood is defined as a marginal over bounded PCFG trees τ of the probability of that tree given the grammar times the product of the probability of the word at each time step or token index t given this tree:[8]

$$\mathsf{P}(w_{1..T} \mid \mathbf{G}_D) = \sum_{\tau} \mathsf{P}(\tau \mid \mathbf{G}_D) \cdot \prod_{t} \mathsf{P}(w_t \mid \tau). \quad (10)$$

The probability of each tree is defined to be the product of the probabilities of each of its branches:[9]

$$\mathsf{P}(\tau \mid \mathbf{G}_D) = \prod_{\tau_\eta \in \tau} \mathsf{P}_{\mathbf{G}_D}(\tau_\eta \rightarrow \tau_{\eta 0}\ \tau_{\eta 1} \mid \tau_\eta). \quad (11)$$

[7] This definition assumes a Kronecker delta function δ_i, defined as a vector with value one at index i and zeros everywhere else, and a Kronecker product $\mathbf{M} \otimes \mathbf{N}$ over matrices $\mathbf{M}$ and $\mathbf{N}$, which tiles copies of $\mathbf{N}$ weighted by values in $\mathbf{M}$ as follows:

$$\mathbf{M} \otimes \mathbf{N} = \begin{bmatrix} \mathbf{M}_{[1,1]}\,\mathbf{N} & \mathbf{M}_{[1,2]}\,\mathbf{N} & \cdots \\ \mathbf{M}_{[2,1]}\,\mathbf{N} & \mathbf{M}_{[2,2]}\,\mathbf{N} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}. \quad (1')$$

The Kronecker product specializes to vectors as single-column matrices, generating vectors that contain the products of all combinations of elements in the operand vectors.

[8] This notation assumes the observed data $w_{1..T}$ is a single long sequence of words, and the hidden variable τ is a single large but depth-bounded tree structure (e.g. a right-branching discourse structure). Since the implementation is incremental, segmentation decisions may indeed be treated as hidden variables in τ, but the experiments described in Section 5 are run on sentence-segmented input.

[9] Here, η is a node address, with left child $\eta 0$ and right child $\eta 1$, or with right child equal to $\perp$ if unary.

The probability $\mathsf{P}(\mathbf{G}_D)$ is itself an integral over the product of a deterministic transform ϕ from an unbounded grammar to a bounded grammar $\mathsf{P}(\mathbf{G}_D \mid \mathbf{G}) = [\![\mathbf{G}_D = \phi(\mathbf{G})]\!]$ and a prior over unbounded grammars $\mathsf{P}(\mathbf{G})$:

$$\mathsf{P}(\mathbf{G}_D) = \int \mathsf{P}(\mathbf{G}_D \mid \mathbf{G}) \cdot \mathsf{P}(\mathbf{G}) \cdot d\mathbf{G}. \quad (12)$$

Distributions $\mathsf{P}(\mathbf{G})$ for each nonterminal symbol (rows) within this unbounded grammar can then be sampled from a Dirichlet distribution with a symmetric parameter β:

$$\mathbf{G} \sim \mathrm{Dirichlet}(\beta), \quad (13)$$

which then yields a corresponding transformed sample in $\mathsf{P}(\mathbf{G}_D)$ for corresponding nonterminals. Note that this model is different than that of Shain et al. (2016), who induce a hierarchical HMM directly.

A depth-specific grammar $\mathbf{G}_D$ is (deterministically) derived from $\mathbf{G}$ via transform ϕ with probabilities for expansions constrained to and renormalized over only those outcomes that yield terminals within a particular depth bound D. This depth-bounded grammar is then used to derive left-corner expectations (anticipated counts of categories appearing as left descendants of other categories), and ultimately the parameters of the depth-bounded left-corner parser defined in Section 3. Counts for $\mathbf{G}$ are then obtained from sampled hidden state sequences, and rows of $\mathbf{G}$ are then directly sampled from the posterior updated by these counts.

4.1 Depth-bounded grammar

In order to ensure the bounded version of $\mathbf{G}$ is a consistent probability model, it must be renormalized in transform ϕ to assign a probability of zero to any derivation that exceeds its depth bound D. For example, if $D = 2$, then it is not possible to expand a left sibling at depth 2 to anything other than a lexical item, so the probability of any non-lexical expansion must be removed from the depth-bounded model, and the probabilities of all remaining outcomes must be renormalized to a new total without this probability. Following van Schijndel et al. (2013), 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 $s \in \{\mathrm{L}, \mathrm{R}\}$ at depth $d \in \{1..D\}$ at each it-

eration $i \in \{1..I\}$,[10] 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:

$$\mathbf{h}^{(0)}_{s,d} = \mathbf{0} \tag{14a}$$

$$\mathbf{h}^{(i)}_{\mathrm{L},d} = \begin{cases} \mathbf{G}\,(\mathbf{1} \otimes \delta_\perp \;+\; \mathbf{h}^{(i-1)}_{\mathrm{L},d} \otimes \mathbf{h}^{(i-1)}_{\mathrm{R},d}) & \text{if } d \leq D+1 \\ \mathbf{0} & \text{if } d > D+1 \end{cases} \tag{14b}$$

$$\mathbf{h}^{(i)}_{\mathrm{R},d} = \begin{cases} \delta_{\mathrm{T}} & \text{if } d = 0 \\ \mathbf{G}\,(\mathbf{1} \otimes \delta_\perp \;+\; \mathbf{h}^{(i-1)}_{\mathrm{L},d+1} \otimes \mathbf{h}^{(i-1)}_{\mathrm{R},d}) & \text{if } 0 < d \leq D \\ \mathbf{0} & \text{if } d > D. \end{cases} \tag{14c}$$

where 'T' is a top-level category label at depth zero.

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:[11]

$$\mathbf{G}_{\mathrm{L},d} = \frac{\mathbf{G}\,\mathrm{diag}(\mathbf{1} \otimes \delta_\perp \;+\; \mathbf{h}^{(I)}_{\mathrm{L},d} \otimes \mathbf{h}^{(I)}_{\mathrm{R},d})}{\mathbf{h}^{(I)}_{\mathrm{L},d}} \tag{15a}$$

$$\mathbf{G}_{\mathrm{R},d} = \frac{\mathbf{G}\,\mathrm{diag}(\mathbf{1} \otimes \delta_\perp \;+\; \mathbf{h}^{(I)}_{\mathrm{L},d+1} \otimes \mathbf{h}^{(I)}_{\mathrm{R},d})}{\mathbf{h}^{(I)}_{\mathrm{R},d}}. \tag{15b}$$

This renormalization ensures the depth-bounded model is consistent. Moreover, this distinction between a learned *unbounded* grammar $\mathbf{G}$ and a derived *bounded* grammar $\mathbf{G}_{s,d}$ which is used to derive a parsing model may be regarded as an instance of Chomsky's (1965) distinction between linguistic competence and performance.

The side- and depth-specific grammar can then be used to define expected counts of categories occurring as left descendants (or 'left corners') of right-sibling ancestors:

$$\mathbf{E}^{(1)}_{d} = \mathbf{G}_{\mathrm{R},d}\,(\mathrm{diag}(\mathbf{1}) \otimes \mathbf{1}) \tag{16a}$$

$$\mathbf{E}^{(i)}_{d} = \mathbf{E}^{(i-1)}_{d}\,\mathbf{G}_{\mathrm{L},d}\,(\mathrm{diag}(\mathbf{1}) \otimes \mathbf{1}) \tag{16b}$$

$$\mathbf{E}^{+}_{d} = \textstyle\sum_{i=1}^{I} \mathbf{E}^{(i)}_{d}. \tag{16c}$$

This left-corner expectation will be used to estimate the marginalized probability over all grammar rule expansions between derivation fragments, which must traverse an unknown number of left children of some right-sibling ancestor.

4.2 Depth-bounded parsing

Again following van Schijndel et al. (2013), the fork and join decision, and the preterminal, top and bottom category label sub-models described in Section 3 can now be defined in terms of these side- and depth-specific grammars $\mathbf{G}_{s,d}$ and depth-specific left-corner expectations $\mathbf{E}^{+}_{d}$.

First, probabilities for no-fork and yes-fork outcomes below some bottom sign of category b at depth d are defined as the normalized probabilities, respectively, of any lexical expansion of a right sibling b at depth d, and of any lexical expansion following any number of left child expansions from b at depth d:

$$\mathsf{P}_{\theta_{\mathrm{F},d}}(0 \mid b) = \frac{{\delta_b}^{\top}\mathbf{G}_{\mathrm{R},d}\,(\mathbf{1} \otimes \delta_\perp)}{{\delta_b}^{\top}(\mathbf{G}_{\mathrm{R},d} + \mathbf{E}^{+}_{d}\,\mathbf{G}_{\mathrm{L},d})\,(\mathbf{1} \otimes \delta_\perp)} \tag{17a}$$

$$\mathsf{P}_{\theta_{\mathrm{F},d}}(1 \mid b) = \frac{{\delta_b}^{\top}\mathbf{E}^{+}_{d}\,\mathbf{G}_{\mathrm{L},d}\,(\mathbf{1} \otimes \delta_\perp)}{{\delta_b}^{\top}(\mathbf{G}_{\mathrm{R},d} + \mathbf{E}^{+}_{d}\,\mathbf{G}_{\mathrm{L},d})\,(\mathbf{1} \otimes \delta_\perp)}. \tag{17b}$$

The probability of a preterminal p given a bottom category b is simply a normalized left-corner expected count of p under b:

$$\mathsf{P}_{\theta_{\mathrm{P},d}}(p \mid b) \stackrel{\mathrm{def}}{=} \frac{{\delta_b}^{\top}\mathbf{E}^{+}_{d}\,\delta_p}{{\delta_b}^{\top}\mathbf{E}^{+}_{d}\,\mathbf{1}}. \tag{18}$$

Yes-join and no-join probabilities below bottom sign b and above top sign a at depth d are then defined similarly to fork probabilities, as the normalized probabilities, respectively, of an expansion to left child a of a right sibling b at depth d, and of an expansion to left child a following any number of

[10] Experiments described in this article use $I = 20$ following observations of convergence at this point in supervised parsing.

[11] where $\mathrm{diag}(\mathbf{v})$ is a diagonalization of a vector $\mathbf{v}$:

$$\mathrm{diag}(\mathbf{v}) = \begin{bmatrix} \mathbf{v}_{[1]} & 0 & \cdots \\ 0 & \mathbf{v}_{[2]} & \\ \vdots & & \ddots \end{bmatrix}. \tag{2'}$$

left child expansions from b at depth d:

$$\mathsf{P}_{\theta_{\mathrm{J},d}}(1 \mid b\, a) = \frac{\delta_b{}^\top \mathbf{G}_{\mathrm{R},d}\,(\delta_a \otimes \mathbf{1})}{\delta_b{}^\top (\mathbf{G}_{\mathrm{R},d} + \mathbf{E}_d^+\,\mathbf{G}_{\mathrm{L},d})\,(\delta_a \otimes \mathbf{1})} \quad (19a)$$

$$\mathsf{P}_{\theta_{\mathrm{J},d}}(0 \mid b\, a) = \frac{\delta_b{}^\top \mathbf{E}_d^+\,\mathbf{G}_{\mathrm{L},d}\,(\delta_a \otimes \mathbf{1})}{\delta_b{}^\top (\mathbf{G}_{\mathrm{R},d} + \mathbf{E}_d^+\,\mathbf{G}_{\mathrm{L},d})\,(\delta_a \otimes \mathbf{1})}. \quad (19b)$$

The distribution over category labels for top signs a above some top sign of category c and below a bottom sign of category b at depth d is defined as the normalized distribution over category labels following a chain of left children expanding from b which then expands to have a left child of category c:

$$\mathsf{P}_{\theta_{\mathrm{A},d}}(a \mid b\, c) = \frac{\delta_b{}^\top \mathbf{E}_d^+ \,\mathrm{diag}(\delta_a)\,\mathbf{G}_{\mathrm{L},d}\,(\delta_c \otimes \mathbf{1})}{\delta_b{}^\top \mathbf{E}_d^+ \,\mathrm{diag}(\mathbf{1})\,\mathbf{G}_{\mathrm{L},d}\,(\delta_c \otimes \mathbf{1})}. \quad (20)$$

The distribution over category labels for bottom signs b below some sign a and sibling of top sign c is then defined as the normalized distribution over right children of grammar rules expanding from a to c followed by b:

$$\mathsf{P}_{\theta_{\mathrm{B},s,d}}(b \mid a\, c) = \frac{\delta_a{}^\top \mathbf{G}_{\mathrm{s},d}\,(\delta_c \otimes \delta_b)}{\delta_a{}^\top \mathbf{G}_{\mathrm{s},d}\,(\delta_c \otimes \mathbf{1})}. \quad (21)$$

Finally, a lexical observation model $\mathbf{L}$ is defined as a matrix of unary rule probabilities with one row for each combination of store state and preterminal symbol and one column for each observation symbol:

$$\mathbf{L} = \mathbf{1} \otimes \mathbf{G}\,(\mathrm{diag}(\mathbf{1}) \otimes \delta_\perp). \quad (22)$$

4.3 Gibbs sampling

Grammar induction in this model then follows a forward-filtering backward-sampling algorithm (Carter and Kohn, 1996). This algorithm first computes a forward distribution $\mathbf{v}_t$ over hidden states at each time step t from an initial value $\perp$:

$$\mathbf{v}_0{}^\top = \delta_\perp{}^\top \quad (23a)$$

$$\mathbf{v}_t{}^\top = \mathbf{v}_{t-1}{}^\top\, \mathbf{M}\,\mathrm{diag}(\mathbf{L}\,\delta_{w_t}). \quad (23b)$$

The algorithm then samples hidden states backward from a multinomial distribution given the previously sampled state q_{t+1} at time step $t{+}1$ (assuming input parameters to the multinomial function are normalized):

$$q_t \sim \mathrm{Multinom}(\,\mathrm{diag}(\mathbf{v}_t)\,\mathbf{M}\,\mathrm{diag}(\mathbf{L}\,\delta_{w_{t+1}})\,\delta_{q_{t+1}}\,). \quad (24)$$

Grammar rule applications $\mathbf{C}$ are then counted from these sampled sequences:[12]

$$\mathbf{C} = \sum_t \begin{cases} \delta_{b_{t-1}^{\bar{d}-1}}\,(\delta_{a_{t-1}^{\bar{d}}} \otimes \delta_{b_t^{\bar{d}-1}})^\top & \text{if } f_t, j_t = 0, 1 \\ \delta_{a_t^{\bar{d}}}\,(\delta_{a_{t-1}^{\bar{d}}} \otimes \delta_{b_t^{\bar{d}}})^\top & \text{if } f_t, j_t = 0, 0 \\ \delta_{b_{t-1}^{\bar{d}}}\,(\delta_{p_t} \otimes \delta_{b_t^{\bar{d}}})^\top & \text{if } f_t, j_t = 1, 1 \\ \delta_{a_t^{\bar{d}+1}}\,(\delta_{p_t} \otimes \delta_{b_t^{\bar{d}+1}})^\top & \text{if } f_t, j_t = 1, 0 \end{cases} + \sum_t \delta_{p_t}\,(\delta_{w_t} \otimes \delta_\perp)^\top, \quad (25)$$

and a new grammar $\mathbf{G}$ is sampled from a Dirichlet distribution with counts $\mathbf{C}$ and a symmetric hyper-parameter β as parameters:

$$\mathbf{G} \sim \mathrm{Dirichlet}(\,\mathbf{C} + \beta\,). \quad (26)$$

This grammar is then used to define transition and lexical models $\mathbf{M}$ and $\mathbf{L}$ as defined in Sections 3 through 4.2 to complete the cycle.

4.4 Model hyper-parameters and priors

There are three hyper-parameters in the model. K is the number of non-terminal categories in the grammar $\mathbf{G}$, D is the maximum depth, and β is the parameter for the symmetric Dirichlet prior over multinomial distributions in the grammar $\mathbf{G}$.

As seen from the previous subsection, the prior is over all possible rules in an unbounded PCFG grammar. Because the number of non-terminal categories of the unbounded PCFG grammar is given as a hyper-parameter, the number of rules in the grammar is always known. It is possible to use non-parametric priors over the number of non-terminal categories, however due to the need to dynamically mitigate the computational complexity of filtering and sampling using arbitrarily large category sets, this is left for future work.

5 Evaluation

The DB-PCFG model described in Section 4 is evaluated first on synthetic data to determine whether it can reliably learn a recursive grammar from data with a known optimum solution, and to determine the hyper-parameter value for β for doing so. Two experiments on natural data are then carried out. First, the model is run on natural data from the Adam

[12] Again, $\bar{d} = \max_d\{a_{t-1}^d \neq \perp\}$.

and Eve parts of the CHILDES corpus (Macwhinney, 1992) to compare with other grammar induction systems on a human-like acquisition task. Then data from the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993) is used for further comparison in a domain for which competing systems are optimized. The competing systems include UPPARSE (Ponvert et al., 2011)[13], CCL (Seginer, 2007a)[14], BMMM+DMV with undirected dependency features (Christodoulopoulos et al., 2012)[15] and UHHMM (Shain et al., 2016).[16]

For the natural language datasets, the variously parametrized DB-PCFG systems[17] are first validated on a development set, and the optimal system is then run until convergence with the chosen hyperparameters on the test set. In development experiments, the log-likelihood of the dataset plateaus usually after 500 iterations. The system is therefore run at least 500 iterations in all test set experiments, with one iteration being a full cycle of Gibbs sampling. The system is then checked to see whether the log-likelihood has plateaued, and halted if it has.

The DB-PCFG model assigns trees sampled from conditional posteriors to all sentences in a dataset in every iteration as part of the inference. The system is further allowed to run at least 250 iterations after convergence and proposed parses are chosen from the iteration with the greatest log-likelihood after convergence. However, once the system reaches convergence, the evaluation scores of parses from different iterations post-convergence appear to differ very little.

5.1 Synthetic data

Following Liang et al. (2009) and Scicluna and de la Higuera (2014), an initial set of experiments on synthetic data are used to investigate basic properties of the model—in particular:

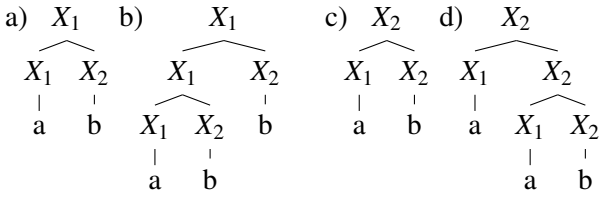

Figure 2: Synthetic left-branching (a,b) and right-branching (c,d) datasets.

1. whether the model is balanced or biased in favor of left- or right-branching solutions,

2. whether the model is able to posit recursive structure in appropriate places, and

3. what hyper-parameters enable the model to find optimal modes more quickly.

The risk of bias in branching structure is important because it might unfairly inflate induction results on languages like English, which are heavily right branching. In order to assess its bias, the model is evaluated on two synthetic datasets, each consisting of 200 sentences. The first dataset is a left-branching corpus, which consists of 100 sentences of the form $a\,b$ and 100 sentences of the form $a\,b\,b$, with optimal tree structures as shown in Figure 2 (a) and (b). The second dataset is a right-branching corpus, which consists of 100 sentences of the form $a\,b$ and 100 sentences of the form $a\,a\,b$, with optimal tree structures as shown in Figure 2 (c) and (d). Results show both structures (and both corresponding grammars) are learnable by the model, and result in approximately the same log likelihood. These synthetic datasets are also used to tune the β hyperparameter of the model (as defined in Section 4) to enable it to find optimal modes more quickly. The resulting β setting of 0.2 is then used in induction on the CHILDES and Penn Treebank corpora.

After validating that the model is not biased, the model is also evaluated on a synthetic center-embedding corpus consisting of 50 sentences each of the form $a\,b\,c$; $a\,b\,b\,c$; $a\,b\,a\,b\,c$; and $a\,b\,b\,a\,b\,b\,c$, which has optimal tree structures as shown in Figure 3.[18] Note that the (b) and (d) trees have depth 2

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

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

[15] BMMM:https://github.com/christos-c/bmmm DMV:https://code.google.com/archive/p/pr-toolkit/

[16] https://github.com/tmills/uhhmm/tree/coling16

[17] The most complex configuration that would run on available GPUs was D=2, K=15. Analysis of full WSJ (Schuler et al., 2010) shows 47.38% of sentences require depth 2, 38.32% require depth 3 and 6.26% require depth 4.

[18] Here, in order to more closely resemble natural language input, tokens a, b, and c are randomly chosen uniformly from $\{a_1, \ldots, a_{50}\}$, $\{b_1, \ldots, b_{50}\}$ and $\{c_1, \ldots, c_{50}\}$, respectively.

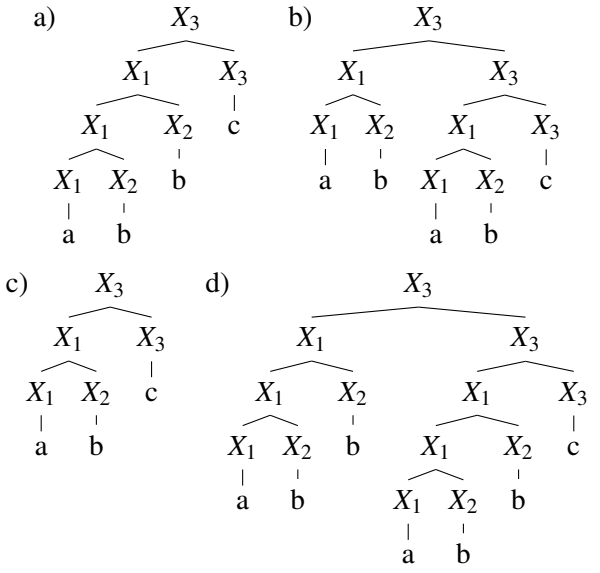

Figure 3: Synthetic center-embedding structure. Note that tree structures (b) and (d) have depth 2 because they have complex sub-trees spanning $a\,b$ and $a\,b\,b$, respectively, embedded in the center of the yield of their roots.

because they each have a complex sub-tree spanning $a\,b$ and $a\,b\,b$ embedded in the center of the yield of the root. Results show the model is capable of learning depth 2 (recursive) grammars.

Finally, as a gauge of the complexity of this task, results of the model described in this paper are compared with those of other grammar induction models on the center-embedding dataset. In this experiment, all models are assigned hyper-parameters matching the optimal solution. The DB-PCFG is run with K=5 and D=2 and β=0.2 for all priors, the BMMM+DMV (Christodoulopoulos et al., 2012) is run with 3 preterminal categories, and the UHHMM model is run with 2 active states, 4 awaited states and 3 parts of speech.[19] Table 1 shows the PARSEVAL scores for parsed trees using the learned grammar from each unsupervised system. Only the DB-PCFG model is able to recognize the correct tree structures and the correct category labels on this dataset, showing the task is indeed a robust challenge. This suggests that hyper-parameters optimized on this dataset may be portable to natural data.

[19]It is not possible to use just 2 awaited states, which is the gold setting, since the UHHMM system errors out when the number of categories is small.

System	Precision	Recall	F1
CCL	83.2	71.1	76.7
UPPARSE	91.4	80.7	85.7
UHHMM	37.7	37.7	37.7
BMMM+DMV	99.2	83.2	90.5
DB-PCFG	**100.0**	**100.0**	**100.0**

Table 1: The performance scores of unlabeled parse evaluation of different systems on synthetic data.

Hyperparameters	Precision	Recall	F1
D1K15	**57.1**	**70.7**	**63.2**
D1K30	52.8	65.4	58.5
D1K45	44.4	54.9	49.1
D2K15	44.0	54.5	48.7

Table 2: PARSEVAL results of different hyperparameter settings for the DB-PCFG system on the Adam dataset. Hyperparameter D is the number of possible depths, and K is the number of non-terminals.

5.2 Child-directed speech corpus

After setting the β hyperparameter on synthetic datasets, the DB-PCFG model is evaluated on 14,251 sentences of transcribed child-directed speech from the Eve section of the Brown corpus of CHILDES (Macwhinney, 1992). Hyperparameters D and K are set to optimize performance on the Adam section of the Brown Corpus of CHILDES, which is about twice as long as Eve. Following previous work, these experiments leave all punctuation in the input for learning, then remove it in all evaluations on development and test data.

Model performance is evaluated against Penn Treebank style annotations of both Adam and Eve corpora (Pearl and Sprouse, 2013). Table 2 shows the PARSEVAL scores of the DB-PCFG system with different hyperparameters on the Adam corpus for development.The simplest configuration, D1K15 (depth 1 only with 15 non-terminal categories), obtains the best score, so this setting is applied to the test corpus, Eve. Results of the D=1, K=15 DB-PCFG model on Eve are then compared against those of other grammar induction systems which use only raw text as input on the same corpus. Following Shain et al. (2016) the BMMM+DMV system is run for 10 iterations with 45 categories and its output is converted from dependency graphs to constituent

System	Precision	Recall	F1
CCL	50.5	53.5	51.9
UPPARSE	60.5	51.9	55.9
UHHMM	55.5	69.3	61.7
BMMM+DMV	63.5	63.3	63.4
UHHMM-F	62.9	68.4	65.6
DB-PCFG	**64.5**	**80.5**	**71.6****
Right-branching	**68.7**	**85.8**	**76.3**

Table 3: PARSEVAL scores on Eve dataset for all competing systems. These are unlabeled precision, recall and F1 scores on constituent trees without punctuation. Both the right-branching baseline and the best performing system are in bold. (**: $p < 0.0001$, permutation test)

System	NP Recall	NP agg F1
CCL	35.5	-
UPPARSE	69.1	-
UHHMM	61.4	27.4
BMMM+DMV	71.3	61.2
DB-PCFG (D1K15)	75.7	28.7
DB-PCFG (D1K30)	78.6	60.7
DB-PCFG (D1K45)	76.9	64.0
DB-PCFG (D2K15)	**85.1**	**65.9**
Right-branching	64.2	-

Table 4: Performances of different systems for noun phrase recall and aggregated F1 scores on the Eve dataset.

trees (Collins et al., 1999). The UHHMM system is run on the Eve corpus using settings in Shain et al. (2016), which also includes a post-process option to flatten trees (reported here as UHHMM-F).

Table 3 shows the PARSEVAL scores for all the competing systems on the Eve dataset. The right-branching baseline is still the most accurate in terms of PARSEVAL scores, presumably because of the highly right-branching structure of child-directed speech in English. The DB-PCFG system with only one memory depth and 15 non-terminal categories achieves the best performance in terms of F1 score and recall among all the competing systems, significantly outperforming other systems ($p < 0.0001$, permutation test).[20]

The Eve corpus has about 5,000 sentences with more than one depth level, therefore one might expect a depth-two model to perform better than a depth-one model, but this is not true if only PARSEVAL scores are considered. This issue will be revisited in the following section with the noun phrase discovery task.

5.3 NP discovery on child-directed speech

When humans acquire grammar, they do not only learn tree structures, they also learn category types: noun phrases, verb phrases, prepositional phrases, and where each type can and cannot occur.

Some of these category types — in particular, noun phrases — are fairly universal across languages, and may be useful in downstream tasks such as (unsupervised) named entity recognition. The DB-PCFG and other models that can be made to produce category types are therefore evaluated on a noun phrase discovery task.

Two metrics are used for this evaluation. First, the evaluation counts all constituents proposed by the candidate systems, and calculates recall against the gold annotation of noun phrases. This metric is not affected by which branching paradigm the system is using and reveals more about the systems' performances. This metric differs from that used by Ponvert et al. (2011) in that this metric takes NPs at all levels in gold annotation into account, not just base NPs.[21]

The second metric, for systems that produce category labels, calculates F1 scores of induced categories that can be mapped to noun phrases. The first 4,000 sentences are used as the development set for learning mappings from induced category labels to phrase types. The evaluation calculates precision, recall and F1 of all spans of proposed categories against the gold annotations of noun phrases in the development set, and aggregates the categories ranked by their precision scores so that the F1 score of the aggregated category is the highest on the development set. The evaluation then calculates the F1 score of this aggregated category on the remainder of the dataset, excluding this development set.

[20] Resulting scores are better when applying Shain et al. (2016) flattening to output binary-branching trees. For the D=1, K=15 model, precision and F1 can be raised to 70.31% and 74.33%. However, since the flattening is a heuristic which may not apply in all cases, these scores are not considered to be comparable results.

[21] Ponvert et al. (2011) define base NPs as NPs with no NP descendants, a restriction motivated by their particular task (chunking).

System	WSJ10test			WSJ20test		
	Precision	Recall	F1	Precision	Recall	F1
CCL	63.4	71.9	67.4	**60.1**	61.7	**60.9****
UPPARSE	54.7	48.3	51.3	47.8	40.5	43.9
UHHMM	49.1	63.4	55.3	-	-	-
BMMM+DMV(K10)	36.2	40.6	38.2	25.3	29.0	27.0
UHHMM-F	57.1	54.4	55.7	-	-	-
DB-PCFG (D2K15)	**64.5**	**82.6**	**72.4****	53.0	**70.5**	60.5
Right-branching	55.1	70.5	61.8	41.5	55.3	47.4

Table 5: PARSEVAL scores for all competing systems on WSJ10 and WSJ20 test sets. These are unlabeled precision, recall and F1 scores on constituent trees without punctuation (**: p <0.0001, permutation test).

The UHHMM system is the only competing system that is natively able to produce labels for proposed constituents. BMMM+DMV does not produce constituents with labels by default, but can be evaluated using this metric by converting dependency graphs into constituent trees, then labeling each constituent with the part-of-speech tag of the head. For CCL and UPPARSE, the NP agg F1 scores are not reported because they do not produce labeled constituents.

Table 4 shows the scores for all systems on the Eve dataset and four runs of the DB-PCFG system on these two evaluation metrics. Surprisingly the D=2, K=15 model which has the lowest PARSEVAL scores is most accurate at discovering noun phrases. It has the highest scores on both evaluation metrics. The best model in terms of PARSEVAL scores, the D=1, K=15 DB-PCFG model, performs poorly among the DB-PCFG models, despite the fact that its NP recall is higher than the competing systems. The low score of NP agg F1 of DB-PCFG at D1K15 shows a diffusion of induced syntactic categories when the model is trying to find a balance among labeling and branching decisions. The UPPARSE system, which is proposed as a base NP chunker, is relatively poor at NP recall by this definition.

The right-branching baseline does not perform well in terms of NP recall. This is mainly because noun phrases are often left children of some other constituent and the right branching model is unable to incorporate them into the syntactic structures of whole sentences. Therefore although the right-branching model is the best model in terms of PARSEVAL scores, it is not helpful in terms of finding noun phrases.

5.4 Penn Treebank

To further facilitate direct comparison to previous work, we run experiments on sentences from the Penn Treebank (Marcus et al., 1993). The first experiment uses the sentences from Wall Street Journal part of the Penn Treebank with at most 20 words (WSJ20). The first half of the WSJ20 dataset is used as a development set (WSJ20dev) and the second half is used as a test set (WSJ20test). We also extract sentences in WSJ20test with at most 10 words from the proposed parses from all systems and report results on them (WSJ10test). WSJ20dev is used for finding the optimal hyperparameters for both DB-PCFG and BMMM-DMV systems.[22]

Table 5 shows the PARSEVAL scores of all systems. The right-branching baseline is relatively weak on these two datasets, mainly because formal writing is more complex and uses more non-right-branching structures (e.g., subjects with modifiers or parentheticals) than child-directed speech. For WSJ10test, both the DB-PCFG system and CCL are able to outperform the right branching baseline. The F1 difference between the best-performing previous-work system, CCL, and DB-PCFG is highly significant. For WSJ20test, again both CCL and DB-

[22] Although UHHMM also needs tuning, in practice we find that this system is too inefficient to be tuned on a development set, and it requires too many resources when the hyperparameters become larger than used in previous work. We believe that further increasing the hyperparameters of UHHMM may lead to performance increase, but the released version is not scalable to larger values of these settings. We also do not report UHHMM on WSJ20test for the same scalabilty reason. The results of WSJ10test of UHHMM is induced with all WSJ10 sentences.

System	WSJ10			WSJ40		
	Precision	Recall	F1	Precision	Recall	F1
CCL	75.3	76.1	75.7	58.7	55.9	57.2
UPPARSE	74.6	66.7	70.5	60.0	49.4	54.2
DB-PCFG (D2K15)	65.5	83.6	73.4	47.0	63.6	54.1
Right-branching	55.2	70.0	61.7	35.4	47.4	40.5

Table 6: Published PARSEVAL results for competing systems. Please see text for details as the systems are trained and evaluated differently.

PCFG are above the right-branching baseline. The difference between the F scores of CCL and DB-PCFG is very small compared to WSJ10, however it is also significant.

It is possible that the DB-PCFG is being penalized for inducing fully binarized parse trees. The accuracy of the DB-PCFG model is dominated by recall rather than precision, whereas CCL and other systems are more balanced. This is an important distinction if it is assumed that phrase structure is binary (Kayne, 1981; Larson, 1988), in which case precision merely scores non-linguistic decisions about whether to suppress annotation of non-maximal projections. However, since other systems are not optimized for recall, it would not be fair to use only recall as a comparison metric in this study.

Finally, Table 6 shows the published results of different systems on WSJ. The CCL results come from Seginer (2007b), where the CCL system is trained with all sentences from WSJ, and evaluated on sentences with 40 words or fewer from WSJ (WSJ40) and WSJ10. The UPPARSE results come from Ponvert et al. (2011), where the UPPARSE system is trained using 00-21 sections of WSJ, and evaluated on section 23 and the WSJ10 subset of section 23. The DB-PCFG system uses hyperparameters optimized on the WSJ20dev set, and is evaluated on WSJ40 and WSJ10, both excluding WSJ20dev. The results are not directly comparable, but the results from the DB-PCFG system is competitive with the other systems, and numerically have the best recall scores.

6 Conclusion

This paper describes a Bayesian Dirichlet model of depth-bounded PCFG induction. Unlike earlier work this model implements depth bounds directly on PCFGs by derivation, reducing the search space of possible trees for input words without exploding the search space of parameters with multiple side- and depth-specific copies of each rule. Results for this model on grammar acquisition from transcribed child-directed speech and newswire text exceed or are competitive with those of other models when evaluated on parse accuracy. Moreover, grammars acquired from this model demonstrate a consistent use of category labels, something which has not been demonstrated by other acquisition models.

In addition to its practical merits, this model may offer some theoretical insight for linguists and other cognitive scientists. First, the model does not assume any universals except independently motivated limits on working memory, which may help address the question of whether universals are indeed necessary for grammar induction. Second, the distinction this model draws between its learned *unbounded* grammar $\mathbf{G}$ and its derived *bounded* grammar $\mathbf{G}_D$ seems to align with Chomsky's (1965) distinction between competence and performance, and has the potential to offer some formal guidance to linguistic inquiry about both kinds of models.

Acknowledgments

The authors would like to thank Cory Shain and William Bryce for their valuable input. We would like also to thank the Action Editor Xavier Carreras and the anonymous reviewers for insightful 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

Steven P. Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *J. Psycholinguistic Research*, 20(3):233–250.

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

Rens Bod. 2006. Unsupervised parsing with U-DOP. In *Proceedings of the Conference on Computational Natural Language Learning*, pages 85–92.

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.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2012. Turning the pipeline into a loop: iterated unsupervised dependency parsing and POS induction. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; Workshop on the Induction of Linguistic Structure*, pages 96–99.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the Workshop on Computational Natural Language Learning*, volume 7, pages 1–8.

Michael Collins, Lance Ramshaw, Jan Hajič, and Christoph Tillmann. 1999. A Statistical Parser for Czech. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 505–512.

Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2011. Posterior sparsity in unsupervised dependency parsing. *Journal of Machine Learning Research*, 12:455–490.

Wenjuan Han, Yong Jiang, and Kewei Tu. 2017. Dependency grammar induction with neural lexicalization and big training data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1684–1689.

Zellig Harris. 1954. Distributional structure. In Jerry A. Fodor and Jerrold J. Katz, editors, *The Structure of Language: Readings in the Philosophy of Language*, volume 10, pages 33–49. Prentice-Hall.

William P. Headden, III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 101–109.

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

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

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

Richard Kayne. 1981. Unambiguous Paths. In R. May and J. Koster, editors, *Levels of Syntactic Representation*, pages 143–183. Foris Publishers.

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.

Tom Kwiatkowski, Luke S. Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233.

Tom Kwiatkowski, Sharon Goldwater, Luke S. Zettlemoyer, and Mark Steedman. 2012. A probabilistic model of syntactic and semantic acquisition from

child-directed utterances and their meanings. In *Proceedings of the Annual Meeting of European Chapter of Association for Computational Linguistics*.

Richard K. Larson. 1988. On the double object construction. *Linguistic Inquiry*, 19(3):335–391.

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

Brian Macwhinney. 1992. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Elrbaum Associates, Mahwah, NJ, third edition.

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.

George A. Miller and Stephen Isard. 1964. Free recall of self-embedded English sentences. *Information and Control*, 7:292–303.

Letitia R. Naigles. 1990. Children use syntax to learn verb meanings. *The Journal of Child Language*, 17:357–374.

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 and Mark Johnson. 2016. Grammar induction from (lots of) words alone. In *Proceedings of the International Conference on Computational Linguistics*, pages 23–32.

Lisa Pearl and Jon Sprouse. 2013. Syntactic islands and learning biases: Combining experimental syntax and computational modeling to investigate the language acquisition problem. *Language Acquisition*, 20(1):23–68, 1.

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.

Philip Resnik. 1992. Left-corner parsing and psychological plausibility. In *Proceedings of the International Conference on Computational Linguistics*, pages 191–197.

Stanley J. Rosenkrantz and Philip M. Lewis, II. 1970. Deterministic left corner parser. In *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata*, pages 139–152.

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

James Scicluna and Colin de la Higuera. 2014. PCFG induction for unsupervised parsing and language modelling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1353–1362.

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

Yoav Seginer. 2007b. *Learning Syntactic Structure*. Ph.D. thesis, University of Amsterdam.

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.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press/Bradford Books, Cambridge, MA.

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.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Proceedings of the Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 658–666, Arlington, Virginia. AUAI Press.

Willem Zuidema. 2003. How the poverty of the stimulus solves the poverty of the stimulus. In *Advances in Neural Information Processing Systems*, volume 15, page 51.