Improving Unsupervised Dependency Parsing with Richer Contexts and Smoothing

William P. Headden III, Mark Johnson, David McClosky

Brown Laboratory for Linguistic Information Processing (BLLIP)

Brown University

Providence, RI 02912

{headdenw,mj,dmcc}@cs.brown.edu

Abstract

Unsupervised grammar induction models tend to employ relatively simple models of syntax when compared to their supervised counterparts. Traditionally, the unsupervised models have been kept simple due to tractability and data sparsity concerns. In this paper, we introduce basic valence frames and lexical information into an unsupervised dependency grammar inducer and show how this additional information can be leveraged via smoothing. Our model produces state-of-the-art results on the task of unsupervised grammar induction, improving over the best previous work by almost 10 percentage points.

1 Introduction

The last decade has seen great strides in statistical natural language parsing. Supervised and semi-supervised methods now provide highly accurate parsers for a number of languages, but require training from corpora hand-annotated with parse trees. Unfortunately, manually annotating corpora with parse trees is expensive and time consuming so for languages and domains with minimal resources it is valuable to study methods for parsing without requiring annotated sentences.

In this work, we focus on unsupervised dependency parsing. Our goal is to produce a directed graph of dependency relations (e.g. Figure 1) where each edge indicates a head-argument relation. Since the task is unsupervised, we are not given any examples of correct dependency graphs and only take words and their parts of speech as input. Most of the recent work in this area (Smith, 2006; Cohen et al., 2008) has focused on variants of the

Figure 1: Example dependency parse.

Dependency Model with Valence (DMV) by Klein and Manning (2004). DMV was the first unsupervised dependency grammar induction system to achieve accuracy above a right-branching baseline. However, DMV is not able to capture some of the more complex aspects of language. Borrowing some ideas from the supervised parsing literature, we present two new models: Extended Valence Grammar (EVG) and its lexicalized extension (L-EVG). The primary difference between EVG and DMV is that DMV uses valence information to determine the number of arguments a head takes but not their categories. In contrast, EVG allows different distributions over arguments for different valence slots. L-EVG extends EVG by conditioning on lexical information as well. This allows L-EVG to potentially capture subcategorizations. The downside of adding additional conditioning events is that we introduce data sparsity problems. Incorporating more valence and lexical information increases the number of parameters to estimate. A common solution to data sparsity in supervised parsing is to add smoothing. We show that smoothing can be employed in an unsupervised fashion as well, and show that mixing DMV, EVG, and L-EVG together produces state-of-the-art results on this task. To our knowledge, this is the first time that grammars with differing levels of detail have been successfully combined for unsupervised dependency parsing.

A brief overview of the paper follows. In Section 2, we discuss the relevant background. Section 3 presents how we will extend DMV with additional

features. We describe smoothing in an unsupervised context in Section 4. In Section 5, we discuss search issues. We present our experiments in Section 6 and conclude in Section 7.

2 Background

In this paper, the observed variables will be a corpus of n sentences of text $\mathbf{s} = s_1 \dots s_n$, and for each word s_{ij} an associated part-of-speech τ_{ij}. We denote the set of all words as V_w and the set of all parts-of-speech as V_τ. The hidden variables are parse trees $\mathbf{t} = t_1 \dots t_n$ and parameters $\bar{\theta}$ which specify a distribution over $\mathbf{t}$. A dependency tree t_i is a directed acyclic graph whose nodes are the words in s_i. The graph has a single incoming edge for each word in each sentence, except one called the *root* of t_i. An edge from word i to word j means that word j is an *argument* of word i or alternatively, word i is the *head* of word j. Note that each word token may be the argument of at most one head, but a head may have several arguments.

If parse tree t_i can be drawn on a plane above the sentence with no crossing edges, it is called *projective*. Otherwise it is *nonprojective*. As in previous work, we restrict ourselves to projective dependency trees. The dependency models in this paper will be formulated as a particular kind of Probabilistic Context Free Grammar (PCFG), described below.

2.1 Tied Probabilistic Context Free Grammars

In order to perform smoothing, we will find useful a class of PCFGs in which the probabilities of certain rules are required to be the same. This will allow us to make independence assumptions for smoothing purposes without losing information, by giving analogous rules the same probability.

Let $G = (\mathcal{N}, \mathcal{T}, S, \mathcal{R}, \theta)$ be a Probabilistic Context Free Grammar with nonterminal symbols $\mathcal{N}$, terminal symbols $\mathcal{T}$, start symbol $S \in \mathcal{N}$, set of productions $\mathcal{R}$ of the form $N \to \beta$, $N \in \mathcal{N}, \beta \in (\mathcal{N} \cup \mathcal{T})^*$. Let $\mathcal{R}_N$ indicate the subset of $\mathcal{R}$ whose left-hand sides are N. θ is a vector of length $|\mathcal{R}|$, indexed by productions $N \to \beta \in \mathcal{R}$. $\theta_{N \to \beta}$ specifies the probability that N rewrites to β. We will let θ_N indicate the subvector of θ corresponding to $\mathcal{R}_N$.

A tied PCFG constrains a PCFG G with a tying relation, which is an equivalence relation over rules that satisfies the following properties:

1. Tied rules have the same probability.
2. Rules expanding the same nonterminal are never tied.
3. If $N_1 \to \beta_1$ and $N_2 \to \beta_2$ are tied then the tying relation defines a one-to-one mapping between rules in $\mathcal{R}_{N_1}$ and $\mathcal{R}_{N_2}$, and we say that N_1 and N_2 are tied nonterminals.

As we see below, we can estimate tied PCFGs using standard techniques. Clearly, the tying relation also defines an equivalence class over nonterminals. The tying relation allows us to formulate the distributions over trees in terms of rule equivalence classes and nonterminal equivalence classes. Suppose $\bar{\mathcal{R}}$ is the set of rule equivalence classes and $\bar{\mathcal{N}}$ is the set of nonterminal equivalence classes. Since all rules in an equivalence class $\bar{r}$ have the same probability (condition 1), and since all the nonterminals in an equivalence class $\bar{N} \in \bar{\mathcal{N}}$ have the same distribution over rule equivalence classes (condition 1 and 3), we can define the set of rule equivalence classes $\bar{\mathcal{R}}_{\bar{N}}$ associated with a nonterminal equivalence class $\bar{N}$, and a vector $\bar{\theta}$ of probabilities, indexed by rule equivalence classes $\bar{r} \in \bar{\mathcal{R}}$. $\bar{\theta}_{\bar{N}}$ refers to the subvector of $\bar{\theta}$ associated with nonterminal equivalence class $\bar{N}$, indexed by $\bar{r} \in \bar{\mathcal{R}}_{\bar{N}}$. Since rules in the same equivalence class have the same probability, we have that for each $r \in \bar{r}$, $\theta_r = \bar{\theta}_{\bar{r}}$.

Let $f(\mathbf{t}, r)$ denote the number of times rule r appears in tree $\mathbf{t}$, and let $f(\mathbf{t}, \bar{r}) = \sum_{r \in \bar{r}} f(\mathbf{t}, r)$. We see that the complete data likelihood is

$$P(\mathbf{s}, \mathbf{t}|\theta) = \prod_{\bar{r} \in \bar{\mathcal{R}}} \prod_{r \in \bar{r}} \theta_r^{f(\mathbf{t},r)} = \prod_{\bar{r} \in \bar{\mathcal{R}}} \bar{\theta}_{\bar{r}}^{f(\mathbf{t},\bar{r})}$$

That is, the likelihood is a product of multinomials, one for each nonterminal equivalence class, and there are no constraints placed on the parameters of these multinomials besides being positive and summing to one. This means that all the standard estimation methods (e.g. Expectation Maximization, Variational Bayes) extend directly to tied PCFGs.

Maximum likelihood estimation provides a point estimate of $\bar{\theta}$. However, often we want to incorporate information about $\bar{\theta}$ by modeling its *prior* distribution. As a prior, for each $\bar{N} \in \bar{\mathcal{N}}$ we will specify a

Dirichlet distribution over $\bar{\theta}_{\bar{N}}$ with hyperparameters $\alpha_{\bar{N}}$. The Dirichlet has the density function:

$$P(\bar{\theta}_{\bar{N}}|\alpha_{\bar{N}}) = \frac{\Gamma(\sum_{\bar{r}\in\bar{\mathcal{R}}_{\bar{N}}}\alpha_{\bar{r}})}{\prod_{\bar{r}\in\bar{\mathcal{R}}_{\bar{N}}}\Gamma(\alpha_{\bar{r}})}\prod_{\bar{r}\in\bar{\mathcal{R}}_{\bar{N}}}\bar{\theta}_{\bar{r}}^{\alpha_{\bar{r}}-1},$$

Thus the prior over $\bar{\theta}$ is a product of Dirichlets,which is *conjugate* to the PCFG likelihood function (Johnson et al., 2007). That is, the posterior $P(\bar{\theta}|\mathbf{s},\mathbf{t},\alpha)$ is also a product of Dirichlets, also factoring into a Dirichlet for each nonterminal $\bar{N}$, where the parameters $\alpha_{\bar{r}}$ are augmented by the number of times rule $\bar{r}$ is observed in tree $\mathbf{t}$:

$$\begin{aligned} P(\bar{\theta}|\mathbf{s},\mathbf{t},\alpha) &\propto P(\mathbf{s},\mathbf{t}|\bar{\theta})P(\bar{\theta}|\alpha) \\ &\propto \prod_{\bar{r}\in\bar{\mathcal{R}}}\bar{\theta}_{\bar{r}}^{f(\mathbf{t},\bar{r})+\alpha_{\bar{r}}-1} \end{aligned}$$

We can see that $\alpha_{\bar{r}}$ acts as a pseudocount of the number of times $\bar{r}$ is observed prior to $\mathbf{t}$.

To make use of this prior, we use the Variational Bayes (VB) technique for PCFGs with Dirichlet Priors presented by Kurihara and Sato (2004). VB estimates a distribution over $\bar{\theta}$. In contrast, Expectation Maximization estimates merely a point estimate of $\bar{\theta}$. In VB, one estimates $Q(\mathbf{t},\bar{\theta})$, called the variational distribution, which approximates the posterior distribution $P(\mathbf{t},\bar{\theta}|\mathbf{s},\alpha)$ by minimizing the KL divergence of P from Q. Minimizing the KL divergence, it turns out, is equivalent to maximizing a lower bound $\mathcal{F}$ of the log marginal likelihood $\log P(\mathbf{s}|\alpha)$.

$$\log P(\mathbf{s}|\alpha) \geq \sum_{\mathbf{t}}\int_{\bar{\theta}} Q(\mathbf{t},\bar{\theta})\log\frac{P(\mathbf{s},\mathbf{t},\bar{\theta}|\alpha)}{Q(\mathbf{t},\bar{\theta})} = \mathcal{F}$$

The negative of the lower bound, $-\mathcal{F}$, is sometimes called the *free energy*.

As is typical in variational approaches, Kurihara and Sato (2004) make certain independence assumptions about the hidden variables in the variational posterior, which will make estimating it simpler. It factors $Q(\mathbf{t},\bar{\theta}) = Q(\mathbf{t})Q(\bar{\theta}) = \prod_{i=1}^{n}Q_i(t_i)\prod_{\bar{N}\in\bar{\mathcal{N}}}Q(\bar{\theta}_{\bar{N}})$. The goal is to recover $Q(\bar{\theta})$, the estimate of the posterior distribution over parameters and $Q(t)$, the estimate of the posterior distribution over trees. Finding a local maximum of $\mathcal{F}$ is done via an alternating maximization of $Q(\bar{\theta})$ and $Q(\mathbf{t})$. Kurihara and Sato (2004) show that each $Q(\bar{\theta}_{\bar{N}})$ is a Dirichlet distribution with parameters $\hat{\alpha}_r = \alpha_r + E_{Q(\mathbf{t})}f(\mathbf{t},r)$.

2.2 Split-head Bilexical CFGs

In the sections that follow, we frame various dependency models as a particular variety of CFGs known as split-head bilexical CFGs (Eisner and Satta, 1999). These allow us to use the fast Eisner and Satta (1999) parsing algorithm to compute the expectations required by VB in $O(m^3)$ time (Eisner and Blatz, 2007; Johnson, 2007) where m is the length of the sentence.[1]

In the split-head bilexical CFG framework, each nonterminal in the grammar is annotated with a terminal symbol. For dependency grammars, these annotations correspond to words and/or parts-of-speech. Additionally, split-head bilexical CFGs require that each word s_{ij} in sentence s_i is represented in a split form by two terminals called its left part s_{ijL} and right part s_{ijR}. The set of these parts constitutes the terminal symbols of the grammar. This split-head property relates to a particular type of dependency grammar in which the left and right dependents of a head are generated independently. Note that like CFGs, split-head bilexical CFGs can be made probabilistic.

2.3 Dependency Model with Valence

The most successful recent work on dependency induction has focused on the Dependency Model with Valence (DMV) by Klein and Manning (2004). DMV is a generative model in which the head of the sentence is generated and then each head recursively generates its left and right dependents. The arguments of head H in direction d are generated by repeatedly deciding whether to generate another new argument or to stop and then generating the argument if required. The probability of deciding whether to generate another argument is conditioned on H, d and whether this would be the first argument (this is the sense in which it models valence). When DMV generates an argument, the part-of-speech of that argument A is generated given H and d.

[1] Efficiently parsable versions of split-head bilexical CFGs for the models described in this paper can be derived using the fold-unfold grammar transform (Eisner and Blatz, 2007; Johnson, 2007).

Rule	Description
$S \rightarrow Y_H$	Select H as root
$Y_H \rightarrow L_H\ R_H$	Move to split-head representation
$L_H \rightarrow H_L$	STOP $\mid dir = L, head = H, val = 0$
$L_H \rightarrow L^1_H$	CONT $\mid dir = L, head = H, val = 0$
$L'_H \rightarrow H_L$	STOP $\mid dir = L, head = H, val = 1$
$L'_H \rightarrow L^1_H$	CONT $\mid dir = L, head = H, val = 1$
$L^1_H \rightarrow Y_A\ L'_H$	Arg $A \mid dir = L, head = H$

Figure 2: Rule schema for DMV. For brevity, we omit the portion of the grammar that handles the right arguments since they are symmetric to the left (all rules are the same except for the attachment rule where the RHS is reversed). $val \in \{0, 1\}$ indicates whether we have made any attachments.

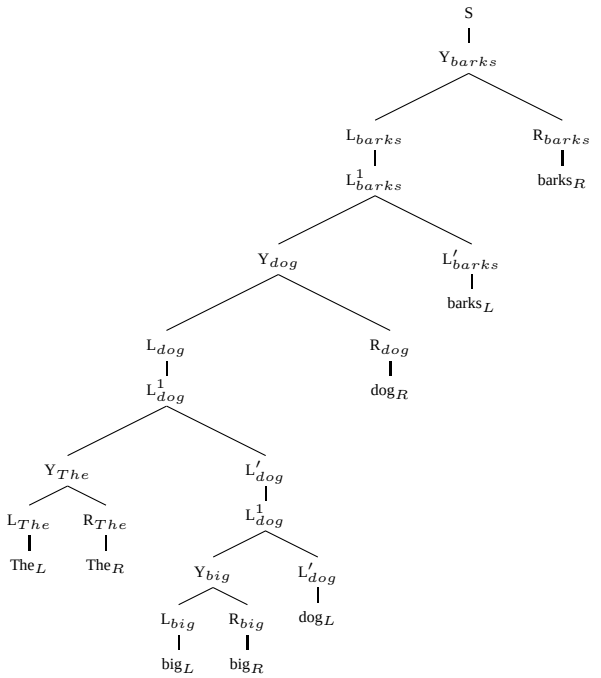

Figure 3: DMV split-head bilexical CFG parse of "The big dog barks."

The grammar schema for this model is shown in Figure 2. The first rule generates the root of the sentence. Note that these rules are for $\forall H, A \in V_\tau$ so there is an instance of the first schema rule for each part-of-speech. Y_H splits words into their left and right components. L_H encodes the stopping decision given that we have not generated any arguments so far. L'_H encodes the same decision after generating one or more arguments. L^1_H represents the distribution over left attachments. To extract dependency relations from these parse trees, we scan for attachment rules (e.g., $L^1_H \rightarrow Y_A\ L'_H$) and record that A depends on H. The schema omits the rules for right arguments since they are symmetric. We show a parse of "The big dog barks" in Figure 3.[2]

Much of the extensions to this work have focused on estimation procedures. Klein and Manning (2004) use Expectation Maximization to estimate the model parameters. Smith and Eisner (2005) and Smith (2006) investigate using Contrastive Estimation to estimate DMV. Contrastive Estimation maximizes the conditional probability of the observed sentences given a neighborhood of similar unseen sequences. The results of this approach vary widely based on regularization and neighborhood, but often outperforms EM.

Smith (2006) also investigates two techniques for maximizing likelihood while incorporating the locality bias encoded in the harmonic initializer for DMV. One technique, skewed deterministic annealing, ameliorates the local maximum problem by flattening the likelihood and adding a bias towards the Klein and Manning initializer, which is decreased during learning. The second technique is structural annealing (Smith and Eisner, 2006; Smith, 2006) which penalizes long dependencies initially, gradually weakening the penalty during estimation. If hand-annotated dependencies on a held-out set are available for parameter selection, this performs far better than EM; however, performing parameter selection on a held-out set without the use of gold dependencies does not perform as well.

Cohen et al. (2008) investigate using Bayesian Priors with DMV. The two priors they use are the Dirichlet (which we use here) and the Logistic Normal prior, which allows the model to capture correlations between different distributions. They initialize using the harmonic initializer of Klein and Manning (2004). They find that the Logistic Normal distribution performs much better than the Dirichlet with this initialization scheme.

Cohen and Smith (2009), investigate (concur-

[2]Note that our examples use words as leaf nodes but in our unlexicalized models, the leaf nodes are in fact parts-of-speech.

Rule	Description
$S \rightarrow Y_H$	Select H as root
$Y_H \rightarrow L_H\ R_H$	Move to split-head representation
$L_H \rightarrow H_L$	STOP | $dir = L, head = H, val = 0$
$L_H \rightarrow L'_H$	CONT | $dir = L, head = H, val = 0$
$L'_H \rightarrow L^1_H$	STOP | $dir = L, head = H, val = 1$
$L'_H \rightarrow L^2_H$	CONT | $dir = L, head = H, val = 1$
$L^2_H \rightarrow Y_A\ L'_H$	Arg A | $dir = L, head = H, val = 1$
$L^1_H \rightarrow Y_A\ H_L$	Arg A | $dir = L, head = H, val = 0$

Figure 4: Extended Valence Grammar schema. As before, we omit rules involving the right parts of words. In this case, $val \in \{0, 1\}$ indicates whether we are generating the nearest argument (0) or not (1).

rently with our work) an extension of this, the Shared Logistic Normal prior, which allows different PCFG rule distributions to share components. They use this machinery to investigate smoothing the attachment distributions for (nouns/verbs), and for learning using multiple languages.

3 Enriched Contexts

DMV models the distribution over arguments identically without regard to their order. Instead, we propose to distinguish the distribution over the argument nearest the head from the distribution of subsequent arguments. [3]

Consider the following changes to the DMV grammar (results shown in Figure 4). First, we will introduce the rule $L^2_H \rightarrow Y_A\ L'_H$ to denote the decision of what argument to generate for positions not nearest to the head. Next, instead of having L'_H expand to H_L or L^1_H, we will expand it to L^1_H (attach to nearest argument and stop) or L^2_H (attach to non-nearest argument and continue). We call this the *Extended Valence Grammar* (EVG).

As a concrete example, consider the phrase "the big hungry dog" (Figure 5). We would expect that distribution over the nearest left argument for "dog" to be different than farther left arguments. The figure shows that EVG allows these two distributions to be different (nonterminals L^2_{dog} and L^1_{dog}) whereas DMV forces them to be equivalent (both use L^1_{dog} as the nonterminal).

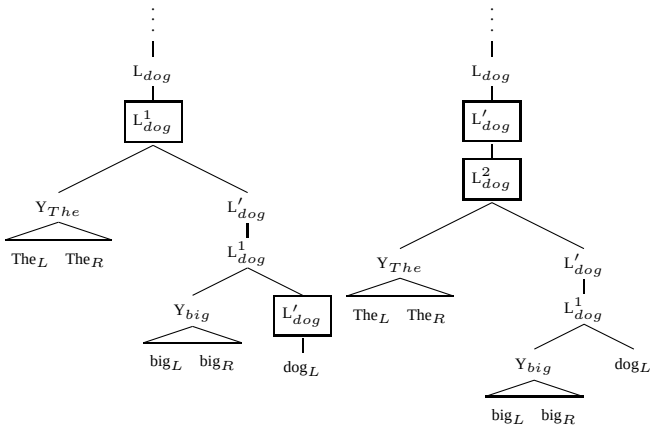

Figure 5: An example of moving from DMV to EVG for a fragment of "The big dog." Boxed nodes indicate changes. The key difference is that EVG distinguishes between the distributions over the argument nearest the head (*big*) from arguments farther away (*The*).

3.1 Lexicalization

All of the probabilistic models discussed thus far have incorporated only part-of-speech information (see Footnote 2). In supervised parsing of both dependencies and constituency, lexical information is critical (Collins, 1999). We incorporate lexical information into EVG (henceforth L-EVG) by extending the distributions over argument parts-of-speech A to condition on the head word h in addition to the head part-of-speech H, direction d and argument position v. The argument word a distribution is merely conditioned on part-of-speech A; we leave refining this model to future work.

In order to incorporate lexicalization, we extend the EVG CFG to allow the nonterminals to be annotated with both the word and part-of-speech of the head. We first remove the old rules $Y_H \rightarrow L_H\ R_H$ for each $H \in V_\tau$. Then we mark each nonterminal which is annotated with a part-of-speech as also annotated with its head, with a single exception: Y_H. We add a new nonterminal $Y_{H,h}$ for each $H \in V_\tau, h \in V_w$, and the rules $Y_H \rightarrow Y_{H,h}$ and $Y_{H,h} \rightarrow L_{H,h}\ R_{H,h}$. The rule $Y_H \rightarrow Y_{H,h}$ corresponds to selecting the word, given its part-of-speech.

[3] McClosky (2008) explores this idea further in an unsmoothed grammar.

4 Smoothing

In supervised estimation one common smoothing technique is *linear interpolation*, (Jelinek, 1997). This section explains how linear interpolation can be represented using a PCFG with tied rule probabilities, and how one might estimate smoothing parameters in an unsupervised framework.

In many probabilistic models it is common to estimate the distribution of some event x conditioned on some set of context information $P(x|N_{(1)} \dots N_{(k)})$ by smoothing it with less complicated conditional distributions. Using linear interpolation we model $P(x|N_{(1)} \dots N_{(k)})$ as a weighted average of two distributions $\lambda_1 P_1(x|N_{(1)}, \dots, N_{(k)}) + \lambda_2 P_2(x|N_{(1)}, \dots, N_{(k-1)})$, where the distribution P_2 makes an independence assumption by dropping the conditioning event $N_{(k)}$.

In a PCFG a nonterminal N can encode a collection of conditioning events $N_{(1)} \dots N_{(k)}$, and θ_N determines a distribution conditioned on $N_{(1)} \dots N_{(k)}$ over events represented by the rules $r \in \mathcal{R}_N$. For example, in EVG the nonterminal L^1_{NN} encodes three separate pieces of conditioning information: the direction $d = \mathit{left}$, the head part-of-speech $H = NN$, and the argument position $v = 0$; $\theta_{L^1_{NN} \to Y_{JJ}\ NN_L}$ represents the probability of generating JJ as the first left argument of NN. Suppose in EVG we are interested in smoothing $P(A \mid d, H, v)$ with a component that excludes the head conditioning event. Using linear interpolation, this would be:

$$P(A \mid d, H, v) = \lambda_1 P_1(A \mid d, H, v) + \lambda_2 P_2(A \mid d, v)$$

We will estimate PCFG rules with linearly interpolated probabilities by creating a tied PCFG which is extended by adding rules that select between the main distribution P_1 and the backoff distribution P_2, and also rules that correspond to draws from those distributions. We will make use of tied rule probabilities to make the independence assumption in the backoff distribution.

We still use the original grammar to parse the sentence. However, we estimate the parameters in the extended grammar and then translate them back into the original grammar for parsing.

More formally, suppose $\mathcal{B} \subseteq \mathcal{N}$ is a set of nonterminals (called the backoff set) with conditioning events $N_{(1)} \dots N_{(k-1)}$ in common (differing in a conditioning event $N_{(k)}$), and with rule sets of the same cardinality. If G is our model's PCFG, we can define a new tied PCFG $G' = (\mathcal{N}', \mathcal{T}, S, \mathcal{R}', \phi)$, where $\mathcal{N}' = \mathcal{N} \cup \left\{ N^{b_\ell} \mid N \in \mathcal{B},\ \ell \in \{1, 2\} \right\}$, meaning for each nonterminal N in the backoff set we add two nonterminals N^{b_1}, N^{b_2} representing each distribution P_1 and P_2. The new rule set $\mathcal{R}' = (\cup_{N \in \mathcal{N}'} \mathcal{R}'_N)$ where for all $N \in \mathcal{B}$ rule set $\mathcal{R}'_N = \left\{ N \to N^{b_\ell} \mid \ell \in \{1, 2\} \right\}$, meaning at N in G' we decide which distribution P_1, P_2 to use; and for $N \in \mathcal{B}$ and $\ell \in \{1, 2\}$, $\mathcal{R}'_{N^{b_\ell}} = \left\{ N^{b_\ell} \to \beta \mid N \to \beta \in \mathcal{R}_N \right\}$ indicating a draw from distribution P_ℓ. For nonterminals $N \notin \mathcal{B}$, $\mathcal{R}'_N = \mathcal{R}_N$. Finally, for each $N, M \in \mathcal{B}$ we specify a tying relation between the rules in $\mathcal{R}'_{N^{b_2}}$ and $\mathcal{R}'_{M^{b_2}}$, grouping together analogous rules. This has the effect of making an independence assumption about P_2, namely that it ignores the conditioning event $N_{(k)}$, drawing from a common distribution each time a nonterminal N^{b_2} is rewritten.

For example, in EVG to smooth $P(A = DT \mid d = \mathit{left}, H = NN, v = 0)$ with $P_2(A = DT \mid d = \mathit{left}, v = 0)$ we define the backoff set to be $\left\{ L^1_H \mid H \in V_\tau \right\}$. In the extended grammar we define the tying relation to form rule equivalence classes by the argument they generate, i.e. for each argument $A \in V_\tau$, we have a rule equivalence class $\left\{ L^{1b_2}_H \to Y_A\ H_L \mid H \in V_\tau \right\}$.

We can see that in grammar G' each $N \in \mathcal{B}$ eventually ends up rewriting to one of N's expansions β in G. There are two indirect paths, one through N^{b_1} and one through N^{b_2}. Thus this defines the probability of $N \to \beta$ in G, $\theta_{N \to \beta}$, as the probability of rewriting N as β in G' via N^{b_1} and N^{b_2}. That is:

$$\theta_{N \to \beta} = \phi_{N \to N^{b_1}} \phi_{N^{b_1} \to \beta} + \phi_{N \to N^{b_2}} \phi_{N^{b_2} \to \beta}$$

The example in Figure 6 shows the probability that L^1_{dog} rewrites to $Y_{big}\ dog_L$ in grammar G.

Typically when smoothing we need to incorporate the prior knowledge that conditioning events that have been seen fewer times should be more strongly smoothed. We accomplish this by setting the Dirichlet hyperparameters for each $N \to N^{b_1}, N \to N^{b_2}$ decision to $(K, 2K)$, where $K = |\mathcal{R}_{N^{b_1}}|$ is the number of rewrite rules for A. This ensures that the model will only start to ignore the backoff distribu-

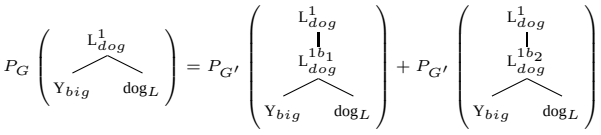

Figure 6: Using linear interpolation to smooth $L^1_{dog} \rightarrow Y_{big}\ dog_L$: The first component represents the distribution fully conditioned on head dog, while the second component represents the distribution ignoring the head conditioning event. This later is accomplished by tying the rule $L^{1b_2}_{dog} \rightarrow Y_{big}\ dog_L$ to, for instance, $L^{1b_2}_{cat} \rightarrow Y_{big}\ cat_L$, $L^{1b_2}_{fish} \rightarrow Y_{big}\ fish_L$ etc.

tion after having seen a sufficiently large number of training examples. [4]

4.1 Smoothed Dependency Models

Our first experiments examine smoothing the distributions over an argument in the DMV and EVG models. In DMV we smooth the probability of argument A given head part-of-speech H and direction d with a distribution that ignores H. In EVG, which conditions on H, d and argument position v we back off two ways. The first is to ignore v and use backoff conditioning event H, d. This yields a backoff distribution with the same conditioning information as the argument distribution from DMV. We call this EVG smoothed-skip-val.

The second possibility is to have the backoff distribution ignore the head part-of-speech H and use backoff conditioning event v, d. This assumes that arguments share a common distribution across heads. We call this EVG smoothed-skip-head. As we see below, backing off by ignoring the part-of-speech of the head H worked better than ignoring the argument position v.

For L-EVG we smooth the argument part-of-speech distribution (conditioned on the head word) with the unlexicalized EVG smoothed-skip-head model.

5 Initialization and Search issues

Klein and Manning (2004) strongly emphasize the importance of smart initialization in getting good performance from DMV. The likelihood function is full of local maxima and different initial parameter values yield vastly different quality solutions. They offer what they call a "harmonic initializer" which initializes the attachment probabilities to favor arguments that appear more closely in the data. This starts EM in a state preferring shorter attachments.

Since our goal is to expand the model to incorporate lexical information, we want an initialization scheme which does not depend on the details of DMV. The method we use is to create M sets of B random initial settings and to run VB some small number of iterations (40 in all our experiments) for each initial setting. For each of the M sets, the model with the best free energy of the B runs is then run out until convergence (as measured by likelihood of a held-out data set); the other models are pruned away. In this paper we use $B = 20$ and $M = 50$.

For the bth setting, we draw a random sample from the prior $\bar{\theta}^{(b)}$. We set the initial $Q(\mathbf{t}) = P(\mathbf{t}|\mathbf{s}, \bar{\theta}^{(b)})$ which can be calculated using the Expectation-Maximization E-Step. $Q(\bar{\theta})$ is then initialized using the standard VB M-step.

For the Lexicalized-EVG, we modify this procedure slightly, by first running MB smoothed EVG models for 40 iterations each and selecting the best model in each cohort as before; each L-EVG distribution is initialized from its corresponding EVG distribution. The new $P(A|h, H, d, v)$ distributions are set initially to their corresponding $P(A|H, d, v)$ values.

6 Results

We trained on the standard Penn Treebank WSJ corpus (Marcus et al., 1993). Following Klein and Manning (2002), sentences longer than 10 words after removing punctuation are ignored. We refer to this variant as WSJ10. Following Cohen et al. (2008), we train on sections 2-21, used 22 as a held-out development corpus, and present results evaluated on section 23. The models were all trained using Variational Bayes, and initialized as described in Section 5. To evaluate, we follow Cohen et al. (2008) in using the mean of the variational posterior Dirichlets as a point estimate $\bar{\theta}'$. For the unsmoothed models we decode by selecting the Viterbi parse given $\bar{\theta}'$, or $argmax_t P(t|s, \bar{\theta}')$.

For the smoothed models we find the Viterbi parse of the unsmoothed CFG, but use the smoothed probabilities. We evaluate against the gold standard

[4] We set the other Dirichlet hyperparameters to 1.

Model	Variant	Dir. Acc.
DMV	harmonic init	46.9*
DMV	random init	55.7 (8.0)
DMV	log normal-families	59.4*
DMV	shared log normal-families	62.4†
DMV	smoothed	61.2 (1.2)
EVG	random init	53.3 (7.1)
EVG	smoothed-skip-val	62.1 (1.9)
EVG	smoothed-skip-head	65.0 (5.7)
L-EVG	smoothed	**68.8** (4.5)

Table 1: Directed accuracy (DA) for WSJ10, section 23. *,† indicate results reported by Cohen et al. (2008), Cohen and Smith (2009) respectively. Standard deviations over 10 runs are given in parentheses

dependencies for section 23, which were extracted from the phrase structure trees using the standard rules by Yamada and Matsumoto (2003). We measure the percent accuracy of the directed dependency edges. For the lexicalized model, we replaced all words that were seen fewer than 100 times with "UNK." We ran each of our systems 10 times, and report the average directed accuracy achieved. The results are shown in Table 1. We compare to work by Cohen et al. (2008) and Cohen and Smith (2009).

Looking at Table 1, we can first of all see the benefit of randomized initialization over the harmonic initializer for DMV. We can also see a large gain by adding smoothing to DMV, topping even the logistic normal prior. The unsmoothed EVG actually performs worse than unsmoothed DMV, but both smoothed versions improve even on smoothed DMV. Adding lexical information (L-EVG) yields a moderate further improvement.

As the greatest improvement comes from moving to model EVG smoothed-skip-head, we show in Table 2 the most probable arguments for each val, dir, using the mean of the appropriate variational Dirichlet. For $d = right, v = 1$, $P(A|v, d)$ largely seems to acts as a way of grouping together various verb types, while for $d = left, v = 0$ the model finds that nouns tend to act as the closest left argument.

Dir,Val	Arg	Prob	Dir,Val	Arg	Prob
left, 0	NN	0.65	right, 0	NN	0.26
	NNP	0.18		RB	0.23
	DT	0.12		NNS	0.12
				IN	0.11
left, 1	CC	0.35	right, 1	IN	0.78
	RB	0.27			
	IN	0.18			

Table 2: Most likely arguments given valence and direction, according to smoothing distribution $P(arg|dir, val)$ in EVG smoothed-skip-head model with lowest free energy.

7 Conclusion

We present a smoothing technique for unsupervised PCFG estimation which allows us to explore more sophisticated dependency grammars. Our method combines linear interpolation with a Bayesian prior that ensures the backoff distribution receives probability mass. Estimating the smoothed model requires running the standard Variational Bayes on an extended PCFG. We used this technique to estimate a series of dependency grammars which extend DMV with additional valence and lexical information. We found that both were helpful in learning English dependency grammars. Our L-EVG model gives the best reported accuracy to date on the WSJ10 corpus.

Future work includes using lexical information more deeply in the model by conditioning argument words and valence on the lexical head. We suspect that successfully doing so will require using much larger datasets. We would also like to explore using our smoothing technique in other models such as HMMs. For instance, we could do unsupervised HMM part-of-speech induction by smooth a tritag model with a bitag model. Finally, we would like to learn the parts-of-speech in our dependency model from text and not rely on the gold-standard tags.

Acknowledgements

This research is based upon work supported by National Science Foundation grants 0544127 and 0631667 and DARPA GALE contract HR0011-06-2-0001. We thank members of BLLIP for their feedback.

References

Shay B. Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of NAACL-HLT 2009*.

Shay B. Cohen, Kevin Gimpel, and Noah A. Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *Advances in Neural Information Processing Systems 21*.

Michael Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, The University of Pennsylvania.

Jason Eisner and John Blatz. 2007. Program transformations for optimization of parsing algorithms and other weighted logic programs. In *Proceedings of the 11th Conference on Formal Grammar*.

Jason Eisner and Giorgio Satta. 1999. Efficient parsing for bilexical context-free grammars and head-automaton grammars. In *Proceedings of ACL 1999*.

Frederick Jelinek. 1997. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of NAACL 2007*.

Mark Johnson. 2007. Transforming projective bilexical dependency grammars into efficiently-parsable CFGs with unfold-fold. In *Proceedings of ACL 2007*.

Dan Klein and Christopher Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of ACL 2002*.

Dan Klein and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL 2004*, July.

Kenichi Kurihara and Taisuke Sato. 2004. An application of the variational bayesian approach to probabilistics context-free grammars. In *IJCNLP 2004 Workshop Beyond Shallow Analyses*.

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. 2008. Modeling valence effects in unsupervised grammar induction. Technical Report CS-09-01, Brown University, Providence, RI, USA.

Noah A. Smith and Jason Eisner. 2005. Guiding unsupervised grammar induction using contrastive estimation. In *International Joint Conference on Artificial Intelligence Workshop on Grammatical Inference Applications*.

Noah A. Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of COLING-ACL 2006*.

Noah A. Smith. 2006. *Novel Estimation Methods for Unsupervised Discovery of Latent Structure in Natural Language Text*. Ph.D. thesis, Department of Computer Science, Johns Hopkins University.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *In Proceedings of the International Workshop on Parsing Technologies*.