Unambiguity Regularization for Unsupervised Learning of Probabilistic Grammars

Kewei Tu*
Departments of Statistics and Computer Science
University of California, Los Angeles
Los Angeles, CA 90095, USA
tukw@ucla.edu

Vasant Honavar
Department of Computer Science
Iowa State University
Ames, IA 50011, USA
honavar@cs.iastate.edu

Abstract

We introduce a novel approach named *unambiguity regularization* for unsupervised learning of probabilistic natural language grammars. The approach is based on the observation that natural language is remarkably unambiguous in the sense that only a tiny portion of the large number of possible parses of a natural language sentence are syntactically valid. We incorporate an inductive bias into grammar learning in favor of grammars that lead to unambiguous parses on natural language sentences. The resulting family of algorithms includes the expectation-maximization algorithm (EM) and its variant, Viterbi EM, as well as a so-called softmax-EM algorithm. The softmax-EM algorithm can be implemented with a simple and computationally efficient extension to standard EM. In our experiments of unsupervised dependency grammar learning, we show that unambiguity regularization is beneficial to learning, and in combination with annealing (of the regularization strength) and sparsity priors it leads to improvement over the current state of the art.

1 Introduction

Machine learning offers a potentially powerful approach to learning probabilistic grammars from data. Because of the high cost of manual sentence annotation, there is substantial interest in unsupervised grammar learning, i.e., the induction of a grammar from a corpus of unannotated sentences. The simplest such approaches attempt to maximize the likelihood of the grammar given the training data, typically using expectation-maximization (EM) (Baker, 1979; Lari and Young, 1990; Klein and Manning, 2004). More recent approaches incorporate additional prior information of the target grammar into learning. For example, Kurihara and Sato (2004) used Dirichlet priors over rule probabilities to obtain smoothed estimates of the probabilities. Johnson et al. (2007) used Dirichlet priors with hyperparameters set to values less than 1 to encourage sparsity of grammar rules. Finkel et al. (2007) and Liang et al. (2007) proposed to use the hierarchical Dirichlet process prior to bias learning towards concise grammars without the need to pre-specify the number of nonterminals. Cohen et al. (2008) and Cohen and Smith (2009) employed the logistic normal prior to model the correlations between grammar symbols. Gillenwater et al. (2010) incorporated a sparsity bias on grammar rules into learning by means of posterior regularization.

More recently, Spitkovsky et al. (2010) and Poon and Domingos (2011) observed that the use of Viterbi EM (also called hard EM) in place of standard EM can lead to significantly improved results in unsupervised learning of probabilistic grammars from natural language and image data respectively, even if no prior information is used. This finding is surprising because Viterbi EM is a degenerate case of standard EM and is therefore generally considered to be less effective in locating the optimum of the objective function. Spitkovsky et al. (2010) speculated that the observed advantage of Viterbi EM over standard EM is due to standard EM reserving too much probability mass to spurious parses in

*Part of the work was done while at Iowa State University.

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1324–1334, Jeju Island, Korea, 12–14 July 2012.

the E-step. However, it is still unclear as to why Viterbi EM can avoid this problem.

Against this background, we propose the use of a novel type of prior information for unsupervised learning of probabilistic natural language grammars, namely the syntactic unambiguity of natural language. Although it is often possible to correctly parse a natural language sentence in more than one way, natural language is remarkably unambiguous in the sense that the number of plausible parses of a natural language sentence is rather small in comparison with the total number of possible parses. Thus, we incorporate into learning an inductive bias in favor of grammars that lead to unambiguous parses on natural language sentences, by using the posterior regularization framework (Ganchev et al., 2010). We name this approach *unambiguity regularization*. The resulting family of algorithms includes standard EM and Viterbi EM, as well as an algorithm that falls between standard EM and Viterbi EM which we call softmax-EM. The softmax-EM algorithm can be implemented with a simple and computationally efficient extension to standard EM. The fact that Viterbi EM is a special case of our approach also gives an explanation of the advantage of Viterbi EM observed in previous work: it is because Viterbi EM implicitly utilizes unambiguity regularization. In our experiments of unsupervised dependency grammar learning, we show that unambiguity regularization is beneficial to learning, and in combination with annealing (of the regularization strength) and sparsity priors it leads to improvement over the current state of the art.

It should be noted that our approach is closely related to the deterministic annealing (DA) technique studied in the optimization literature (Rose, 1998). However, DA has a very different motivation than ours and differs from our approach in a few important algorithmic details, as will be discussed in section 5. When applied to unsupervised grammar learning, DA has been shown to lead to worse parsing accuracy than standard EM (Smith and Eisner, 2004); in contrast, we show that our approach leads to significantly higher parsing accuracy than standard EM in unsupervised dependency grammar learning.

The rest of the paper is organized as follows. Section 2 analyzes the degree of unambiguity of natural language grammars. Section 3 introduces the unambiguity regularization approach and shows that standard EM, Viterbi EM and softmax-EM are its special cases. We show the experimental results in section 4, discuss related work in section 5 and conclude the paper in section 6.

2 The (Un)ambiguity of Natural Language Grammars

A grammar is said to be ambiguous on a sentence if the sentence can be parsed in more than one way by the grammar. It is widely acknowledged that natural language grammars are ambiguous on a significant proportion of natural language sentences. For example, Manning and Schütze (1999) show that a sentence randomly chosen from the Wall Street Journal — "The post office will hold out discounts and service concessions as incentives" — has at least five plausible syntactic parses. When we parse this sentence using the Berkeley parser (Petrov et al., 2006), one of the state-of-the-art English language parsers, we find many alternative parses in addition to the parses shown in (Manning and Schütze, 1999). Indeed, with a probabilistic context-free grammar of only 26 nonterminals (as used in the Berkeley parser), the estimated total number of possible parses[1] of the example sentence is 2×10^{37}. However, upon closer examination, we find that among this very large number of possible parses, only a few have significant probabilities. Figure 1 shows the probabilities of the 100 best parses of the example sentence. We can see that most of the parses have probabilities that are negligible compared with the probability of the best parse (i.e., the parse with the largest probability). Quantitatively, we find that the probabilities of the parses decrease roughly exponentially as we go from the best parse to the less likely parses. We confirmed this observation by examining the parses of many other natural language sentences obtained using the Berkeley parser. This observation suggests that natural language grammars are indeed remarkably unambiguous on natural language sentences, in the sense that for a typical

[1] Given a sentence of length m and a complete Chomsky normal form grammar with n nonterminals, the number of all possible parses is $C_{m-1} \times n^{2m-1}$, where C_{m-1} is the $(m-1)$-th Catalan number. This number is further increased if there are unary rules between nonterminals in the grammar.

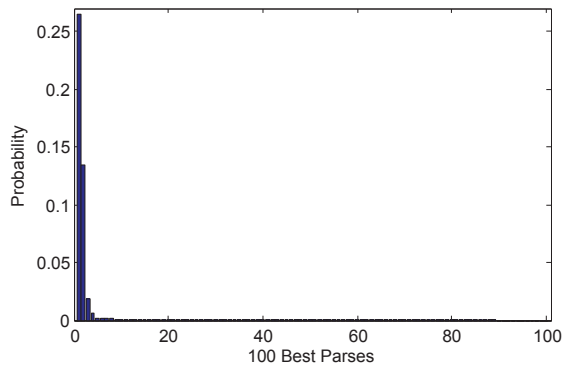


Figure 1: The probabilities of the 100 best parses of the example sentence.

natural language sentence, the probability mass of the parses is concentrated to a tiny portion of all possible parses. This is not surprising in light of the fact that the main purpose of natural language is communication and in the course of language evolution the selection pressure for more efficient communication would favor unambiguous languages.

To highlight the unambiguity of natural language grammars, here we compare the parse probabilities shown in Figure 1 with the parse probabilities produced by two other probabilistic context-free grammars. In figure 2(a) we show the probabilities of the 100 best parses of the example sentence produced by a random grammar. The random grammar has a similar number of nonterminals as in the Berkeley parser, and its grammar rule probabilities are sampled from a uniform distribution and then normalized. It can be seen that unlike the natural language grammar, the random grammar produces a very uniform probability distribution over parses. Figure 2(b) shows the probabilities of the 100 best parses of the example sentence produced by a maximum-likelihood grammar learned from the unannotated Wall Street Journal corpus of the Penn Treebank using the EM algorithm. An exponential decrease can be observed in the probabilities, but the probability mass is still much less concentrated than in the case of the natural language grammar. Again, we confirmed this observation by repeating the experiments on many other natural language sentences. This suggests that both the random grammar and the maximum-likelihood grammar are far more ambiguous on natural language sentences than true natural language grammars.

3 Learning with Unambiguity Regularization

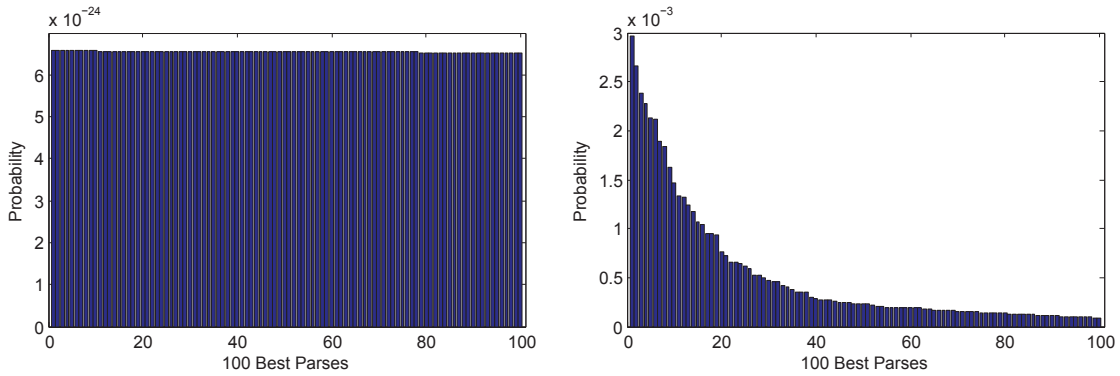
Motivated by the preceding observation, we want to incorporate into learning an inductive bias in favor of grammars that are unambiguous on natural language sentences. First of all, we need a precise definition of the ambiguity of a grammar on a sentence. Assume a grammar with a fixed set of grammar rules and let θ be the rule probabilities. Let x represent a sentence and let z represent the parse of x. One natural measurement of the ambiguity is the information entropy of z conditioned on x and θ:

$$H(z|x,\theta) = -\sum_{z} p_\theta(z|x) \log p_\theta(z|x)$$

The lower the entropy is, the less ambiguous the grammar is on sentence x. When the entropy reaches 0, the grammar is strictly unambiguous on sentence x, i.e., sentence x has a unique parse according to the grammar.

Now we need to modify the objective function of grammar learning to favor low ambiguity of the learned grammar in parsing natural langauge sentences. One approach is to use a prior distribution that favors grammars with low ambiguity on the sentences that they generate. Since the likelihood term in the objective function would ensure that the learned grammar will have high probability of generating natural language sentences, combining the likelihood and the prior would lead to low ambiguity of the learned grammar on natural language sentences. Unfortunately, adding this prior to the objective function makes learning intractable. Hence, here we adopt an alternative approach using the posterior regularization framework (Ganchev et al., 2010). Posterior regularization biases learning in favor of solutions with desired behavior by constraining the model posteriors on the unlabeled data. In our case, we use the constraint that the probability distributions on the parses of the training sentences given the learned grammar must have low entropy, which is equivalent to requiring the learned grammar to have low ambiguity on the training sentences.

Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ denote the set of training sentences, $\mathbf{Z} = \{z_1, z_2, \ldots, z_n\}$ denote the set



(a) (b)

Figure 2: The probabilities of the 100 best parses of the example sentence produced by (a) a random grammar and (b) a maximum-likelihood grammar learned by the EM algorithm.

of parses of the training sentences, and θ denote the rule probabilities of the grammar. We use the slack-penalized version of the posterior regularization objective function:

$$J(\theta) = \log p(\theta|\mathbf{X}) - \min_{q,\xi} \left(\mathbf{KL}(q(\mathbf{Z})||p_\theta(\mathbf{Z}|\mathbf{X})) + \sigma \sum_i \xi_i \right)$$
$$\text{s.t. } \forall i, H(z_i) = -\sum_{z_i} q(z_i) \log q(z_i) \leq \xi_i$$

where σ is a nonnegative constant that controls the strength of the regularization term; q is an auxiliary distribution such that $q(\mathbf{Z}) = \prod_i q(z_i)$. The first term in the objective function is the log posterior probability of the grammar parameters given the training corpus, and the second term minimizes the KL-divergence between the auxiliary distribution q and the posterior distribution on $\mathbf{Z}$ while constrains q to have low entropy. We can incorporate the constraint into the objective function, so we get

$$J(\theta) = \log p(\theta|\mathbf{X}) - \min_{q} \left(\mathbf{KL}(q(\mathbf{Z})||p_\theta(\mathbf{Z}|\mathbf{X})) + \sigma \sum_i H(z_i) \right)$$

To optimize this objective function, we can perform coordinate ascent on a two-variable function:

$$F(\theta, q) = \log p(\theta|\mathbf{X}) - \left(\mathbf{KL}(q(\mathbf{Z})||p_\theta(\mathbf{Z}|\mathbf{X})) + \sigma \sum_i H(z_i) \right)$$

There are two steps in each coordinate ascent iteration. In the first step, we fix q and optimize θ. It can be shown that

$$\begin{aligned} \theta^* &= \arg\max_\theta F(\theta, q) \\ &= \arg\max_\theta \mathbf{E}_q[\log(p_\theta(\mathbf{X}, \mathbf{Z})p(\theta))] \end{aligned}$$

This is equivalent to the M-step in the EM algorithm. The second step fixes θ and optimizes q.

$$q^* = \arg\max_q F(\theta, q) = \arg\min_q \left(\mathbf{KL}(q(\mathbf{Z})||p_\theta(\mathbf{Z}|\mathbf{X})) + \sigma \sum_i H(z_i) \right)$$

It is different from the E-step of the EM algorithm in that it contains an additional regularization term $\sigma \sum_i H(z_i)$. Ganchev et al. (2010) propose to use the projected subgradient method to solve this optimization problem in the general case of posterior regularization. In our case, however, it is possible to obtain an analytical solution as shown below.

First, note that the optimization objective of this step can be rewritten as the sum over functions of individual training sentences.

$$\mathbf{KL}(q(\mathbf{Z})||p_\theta(\mathbf{Z}|\mathbf{X})) + \sigma \sum_i H(z_i) = \sum_i f_i(q)$$

where

$$\begin{aligned} f_i(q) &= \mathbf{KL}(q(z_i)||p_\theta(z_i|x_i)) + \sigma H(z_i) \\ &= \sum_{z_i} \left(q(z_i) \log \frac{q(z_i)^{1-\sigma}}{p_\theta(z_i|x_i)} \right) \end{aligned}$$

So we can optimize $f_i(q)$ for each training sentence x_i. The optimum of $f_i(q)$ depends on the value of the constant σ.

Case 1: $\sigma = 0$**.**

$f_i(q)$ contains only the KL-divergence term, so the second step in the coordinate ascent iteration becomes the standard E-step of the EM algorithm.

$$q^*(z_i) = p_\theta(z_i|x_i)$$

Case 2: $0 < \sigma < 1$**.**

The space of valid assignments of the distribution $q(z_i)$ is a unit $(m-1)$-simplex, where m is the number of valid parses of sentence x_i. Denote this space by Δ.

Theorem 1. *$f_i(q)$ is strictly convex on the unit simplex Δ when $0 < \sigma < 1$.*

Proof Sketch. Define $g(x) = x \log x$, where $g(0)$ is defined to be 0. For any $t \in (0,1)$, for any two points q_1 and q_2 in the unit simplex Δ, we can show that

$$\begin{aligned} & tf_i(q_1) + (1-t)f_i(q_2) - f_i(tq_1 + (1-t)q_2) \\ = \quad & (1-\sigma) \sum_{z_i} \left[\begin{array}{c} tg(q_1(z_i)) + (1-t)g(q_2(z_i)) \\ -\, g(tq_1(z_i) + (1-t)q_2(z_i)) \end{array} \right] \end{aligned}$$

It is easy to prove that $g(x)$ is strictly convex on the interval $[0,1]$. Because $\forall z_i, 0 \le q_1(z_i), q_2(z_i) \le 1$, we have

$$\begin{aligned} & tg(q_1(z_i)) + (1-t)g(q_2(z_i)) \\ > \quad & g(tq_1(z_i) + (1-t)q_2(z_i)) \end{aligned}$$

Because $1 - \sigma > 0$, we have

$$tf_i(q_1) + (1-t)f_i(q_2) - f_i(tq_1 + (1-t)q_2) > 0$$

□

By applying the Lagrange multiplier, we get the stationary point of $f_i(q)$ on the unit simplex Δ:

$$q^*(z_i) = \alpha_i p_\theta(z_i|x_i)^{\frac{1}{1-\sigma}} \tag{1}$$

where α_i is the normalization factor

$$\alpha_i = \frac{1}{\sum_{z_i} p_\theta(z_i|x_i)^{\frac{1}{1-\sigma}}}$$

Because $f_i(q)$ is strictly convex on the unit simplex Δ, this stationary point is the global minimum. Note that because $\frac{1}{1-\sigma} > 1$, $q^*(z_i)$ can be seen as the result of applying a variant of the softmax function to $p_\theta(z_i|x_i)$. To compute q^*, note that $p_\theta(z_i|x_i)$ is the product of a set of grammar rule probabilities, so we can raise all the rule probabilities of the grammar to the power of $\frac{1}{1-\sigma}$ and then run the normal E-step of the EM algorithm. The normalization of q^* is included in the normal E-step.

With q^*, the objective function becomes

$$\begin{aligned} F(\theta, q^*) \quad = \quad & (1-\sigma) \sum_i \log \sum_{z_i} p(z_i, x_i|\theta)^{\frac{1}{1-\sigma}} \\ & + \log p(\theta) - \log p(\mathbf{X}) \end{aligned}$$

The first term is proportional to the log "likelihood" of the corpus computed with the exponentiated rule probabilities. So we can use the parsing algorithm to efficiently compute the value of the objective function (on the training corpus or on a separate development set) to determine when the coordinate ascent iteration shall be terminated.

Case 3: $\sigma = 1$

We need to minimize

$$f_i(q) = -\sum_{z_i} (q(z_i) \log p_\theta(z_i|x_i))$$

Because $\log p_\theta(z_i|x_i) \le 0$ for any z_i, the minimum of $f_i(q)$ is reached at

$$q^*(z_i) = \left\{ \begin{array}{ll} 1 & \text{if } z_i = \arg\max_{z_i} p_\theta(z_i|x_i) \\ 0 & \text{otherwise} \end{array} \right.$$

Case 4: $\sigma > 1$

Theorem 2. *$f_i(q)$ is strictly concave on the unit simplex Δ when $\sigma > 1$.*

The proof is the same as that of theorem 1, except that $1 - \sigma$ is now negative which reverses the direction of the last inequality in the proof.

Theorem 3. *The minimum of $f_i(q)$ is attained at a vertex of the unit simplex Δ.*

Proof. Assume the minimum of $f_i(q)$ is attained at q^* that is not a vertex of the unit simplex Δ, so there are at least two assignments of z_i, say z^1 and z^2, such that $q^*(z^1)$ and $q^*(z^2)$ are nonzero.

Let q' be the same distribution as q^* except that $q'(z^1) = 0$ and $q'(z^2) = q^*(z^1) + q^*(z^2)$. Let q'' be the same distribution as q^* except that $q''(z^1) = q^*(z^1) + q^*(z^2)$ and $q''(z^2) = 0$. Obviously, both q' and q'' are in the unit simplex Δ and $q' \neq q''$. Let $t = \frac{q^*(z^2)}{q^*(z^1)+q^*(z^2)}$, and obviously we have $0 < t < 1$. So we get $q^* = tq' + (1-t)q''$. According to Theorem 2, $f_i(q)$ is strictly concave on the unit simplex Δ, so we have $f_i(q^*) > tf_i(q') + (1-t)f_i(q'')$. Without loss of generality, suppose $f_i(q') \geq f_i(q'')$. So we have $tf_i(q') + (1-t)f_i(q'') \geq f_i(q'')$ and therefore $f_i(q^*) > f_i(q'')$, which means $f_i(q)$ does not attain the minimum at q^*. This contradicts the assumption. □

Now we need to find out at which of the vertices of the unit simplex Δ is the minimum of $f_i(q)$ attained. At the vertex where the probability mass is concentrated at the assignment z, the value of $f_i(q)$ is $-\log p_\theta(z|x_i)$. So the minimum is attained at

$$q^*(z_i) = \begin{cases} 1 & \text{if } z_i = \arg\max_{z_i} p_\theta(z_i|x_i) \\ 0 & \text{otherwise} \end{cases}$$

It can be seen that the minimum in the case of $\sigma > 1$ is attained at the same point as in the case of $\sigma = 1$, at which all the probability mass is assigned to the best parse of the sentence. So q^* can be computed using the E-step of the Viterbi EM algorithm. Denote the best parse by z_i^*. With q^*, the objective function becomes

$$\begin{aligned} F(\theta, q^*) &= \sum_i \log p(z_i^*, x_i|\theta) \\ &\quad + \log p(\theta) - \log p(\mathbf{X}) \end{aligned}$$

The first term is the sum of the log probabilities of the best parses of the corpus. So again we can use the parsing algorithm to efficiently compute it to decide when to terminate the iterative algorithm.

Summary

Our unambiguity regularization approach is an extension of the EM algorithm. The behavior of our approach is controlled by the value of the nonnegative parameter σ. A larger value of σ corresponds to a stronger bias in favor of an unambiguous grammar. When $\sigma = 0$, our approach reduces to the standard EM algorithm. When $\sigma \geq 1$, our approach reduces to the Viterbi EM algorithm, which considers only the best parses of the training sentences in the E-step. When $0 < \sigma < 1$, our approach falls between standard EM and Viterbi EM: it applies a softmax function (Eq.1) to the distributions of parses of the training sentences in the E-step. The softmax function can be computed by simply exponentiating the grammar rule probabilities before the standard E-step, which does not increase the time complexity of the E-step. We refer to the algorithm in the case of $0 < \sigma < 1$ as the *softmax-EM* algorithm.

3.1 Annealing the Strength of Regularization

In unsupervised learning of probabilistic grammars, the initial grammar is typically very ambiguous (e.g., a random grammar). So we need to set σ to a value that is large enough to induce unambiguity. On the other hand, natural language grammars do contain some degree of ambiguity, so if the value of σ is too large, then the learned grammar might be excessively unambiguous and thus not a good model of natural languages. Hence, it is unclear how to choose an optimal value of σ.

One way to avoid choosing a fixed value of σ is to anneal its value. We start learning with a large value of σ (e.g., $\sigma = 1$) to strongly push the learner away from the highly ambiguous initial grammar; then we gradually reduce the value of σ, possibly ending with $\sigma = 0$, to avoid inducing excessive unambiguity in the learned grammar. Note that if the value of σ is annealed to 0, then our approach can be seen as providing an unambiguous initialization for standard EM.

3.2 Unambiguity Regularization with Mean-field Variational Inference

Variational inference approximates the posterior of the model given the data. It typically leads to more accurate predictions than the maximum a posteriori (MAP) estimation. In addition, for certain types of prior distributions (e.g., a Dirichlet prior with hyperparameters set to values less than 1), variational inference is able to find a solution when MAP estimation fails. Here we incorporate unambiguity regularization into mean-field variational inference.

The objective function with unambiguity regular-

ization for mean-field variational inference is:

$$F(q(\theta), q(\mathbf{Z})) = \log p(\mathbf{X}) - \left(\mathbf{KL}(q(\theta)q(\mathbf{Z})||p(\theta, \mathbf{Z}|\mathbf{X})) + \sigma \sum_i H(z_i) \right)$$
$$\text{where } \forall i, H(z_i) = -\sum_{z_i} q(z_i) \log q(z_i)$$

We can perform coordinate ascent that alternately optimizes $q(\theta)$ and $q(\mathbf{Z})$. Since the regularization term does not contain $q(\theta)$, the optimization of $q(\theta)$ is exactly the same as in the standard mean-field variational inference. To optimize $q(\mathbf{Z})$, we have

$$q^*(\mathbf{Z}) = \arg\min_{q(\mathbf{Z})} \left(\mathbf{KL}(q(\mathbf{Z})||\widetilde{p}(\mathbf{X}, \mathbf{Z})) + \sigma \sum_i H(z_i) \right)$$

where $\widetilde{p}(\mathbf{X}, \mathbf{Z})$ is defined as

$$\log \widetilde{p}(\mathbf{X}, \mathbf{Z}) = E_{q(\theta)}[\log p(\theta, \mathbf{Z}, \mathbf{X})] + \text{const}$$

Now we can follow a derivation similar to that in the setting of MAP estimation with unambiguity regularization, and we can obtain a similar result but with $p_\theta(z_i|x_i)$ replaced with $\widetilde{p}(x_i, z_i)$ in each of the four cases.

Note that if Dirichlet priors are used over grammar rule probabilities θ, then $\widetilde{p}(x_i, z_i)$ can be represented as the product of a set of weights in mean-field variational inference (Kurihara and Sato, 2004). Therefore in order to compute $q^*(z_i)$, when $0 < \sigma < 1$, we simply need to raise all the weights to the power of $\frac{1}{1-\sigma}$ before running the normal step of computing $q^*(z_i)$ in standard mean-field variational inference; and when $\sigma \geq 1$, we can simply use the weights to find the best parse of the training sentence and assign probability 1 to it.

4 Experiments

We tested the effectiveness of unambiguity regularization in unsupervised learning of a type of dependency grammar called the dependency model with valence (DMV) (Klein and Manning, 2004). We report the results on the Wall Street Journal corpus (with section 2-21 for training and section 23 for testing) in section 4.1–4.3, and the results on the corpora of eight additional languages in section 4.4. On each corpus, we trained the learner on the gold-standard part-of-speech tags of the sentences of length ≤ 10 with punctuation stripped off. We started our algorithm with the informed initialization proposed in (Klein and Manning, 2004), and terminated the algorithm when the increase in the value of the objective function fell below a threshold of 0.001%. To evaluate a learned grammar, we used the grammar to parse the testing corpus and computed the dependency accuracy which is the percentage of the dependencies that are correctly matched between the parses generated by the grammar and the gold standard parses. We report the dependency accuracy on subsets of the testing corpus corresponding to sentences of length ≤ 10, length ≤ 20, and the entire testing corpus.

Value of σ	Testing Accuracy		
	≤ 10	≤ 20	All
0 (standard EM)	46.2	39.7	34.9
0.25	53.7	44.7	**40.3**
0.5	51.9	42.9	38.8
0.75	51.6	43.1	38.8
1 (Viterbi EM)	**58.3**	**45.2**	39.4

Table 1: The dependency accuracies of grammars learned by our approach with different values of σ.

4.1 Results with Different Values of σ

We compared the performance of our approach with five different values of the parameter σ: 0 (i.e., standard EM), 0.25, 0.5, 0.75, 1 (i.e., Viterbi EM). Table 1 shows the experimental results. It can be seen that learning with unambiguity regularization (i.e., with $\sigma > 0$) consistently outperforms learning without unambiguity regularization (i.e., $\sigma = 0$). The grammar learned by Viterbi EM has significantly higher dependency accuracy in parsing short sentences. We speculate that this is because short sentences are less ambiguous and therefore a strong unambiguity regularization is especially helpful in learning the grammatical structures of short sentences. On the testing sentences of all lengths, $\sigma = 0.25$ achieves the best dependency accuracy, which suggests that controlling the strength of unambiguity regularization can contribute to improved performance.

	Testing Accuracy		
	≤ 10	≤ 20	All
DMV Model			
UR-Annealing	63.6	53.1	47.9
UR-Annealing&Prior	**66.6**	**57.7**	**52.3**
PR-S (Gillenwater et al., 2010)	62.1	53.8	49.1
SLN TieV&N (Cohen and Smith, 2009)	61.3	47.4	41.4
LN Families (Cohen et al., 2008)	59.3	45.1	39.0
Extended Models			
UR-Annealing on E-DMV(2,2)	**71.4**	**62.4**	**57.0**
UR-Annealing on E-DMV(3,3)	71.2	61.5	56.0
L-EVG (Headden et al., 2009)	68.8	-	-
LexTSG-DMV (Blunsom and Cohn, 2010)	67.7	-	55.7

Table 2: The dependency accuracies of grammars learned by our approach (denoted by "UR") with annealing and prior, compared with previous published results.

4.2 Results with Annealing and Prior

We annealed the value of σ from 1 to 0 when running our approach. We reduced the value of σ at a constant speed such that it reaches 0 at iteration 100. The results of this experiment (shown as "UR-Annealing" in Table 2) suggest that annealing the value of σ not only helps circumvent the problem of choosing an optimal value of σ, but may also lead to substantial improvements over the results of learning using any fixed value of σ.

Dirichlet priors with the hyperparameter α set to a value less than 1 are often used to induce parameter sparsity. We added Dirichlet priors over grammar rule probabilities and ran the variational inference version of our approach. The value of α was set to 0.25 as suggested by previous work (Cohen et al., 2008; Gillenwater et al., 2010). When tested with different values of σ, adding Dirichlet priors with $\alpha = 0.25$ consistently boosted the dependency accuracy of the learned grammar by 1–2%. When the value of σ was annealed during variational inference with Dirichlet priors, the dependency accuracy was further improved (shown as "UR-Annealing&Prior" in Table 2).

The first part of Table 2 also compares our results with the best results that have been published in the literature for unsupervised learning of the DMV model (with different priors or regularizations than ours). It can be seen that our best result (unambiguity regularization with annealing and prior) clearly outperforms previous results. Furthermore, we expect our approach to be more computationally efficient than the other approaches, because our approach only inserts an additional parameter exponentiation step into each iteration of standard EM or variational inference, in contrast to the other three approaches all of which involve additional gradient descent optimization steps in each iteration.

4.3 Results on Extended Models

It has been pointed out that the DMV model is very simplistic and cannot capture many linguistic phenomena; therefore a few extensions of DMV have been proposed, which achieve significant improvement over DMV in unsupervised grammar learning (Headden et al., 2009; Blunsom and Cohn, 2010). We examined the effect of unambiguity regularization on E-DMV, an extension of DMV (with two different settings: (2,2) and (3,3)) (Headden et al., 2009; Gillenwater et al., 2010). As shown in the second part of Table 2, unambiguity regularization with annealing on E-DMV achieves better dependency accuracies than the state-of-the-art approaches to unsupervised parsing with extended dependency models. Addition of Dirichlet priors, however, did not further improve the accuracies in this setting. Note that E-DMV is an unlexicalized extension of DMV that is relatively simple. We speculate that the performance of unambiguity regularization can be further improved if applied to more advanced models like LexTSG-DMV (Blunsom and Cohn, 2010).

4.4 Results on More Languages

We examined the effect of unambiguity regularization with the DMV model on the corpora of eight additional languages[2]. The experimental results of all the nine languages are summarized in Table 3. It can be seen that learning with unambiguity regularization (i.e., with $\sigma > 0$) outperforms learning without unambiguity regularization (i.e., $\sigma = 0$) on eight out of the nine languages, but the optimal value of σ is very different across languages. Annealing the value of σ from 1 to 0 does not always lead to further improvement over using the optimal value of σ

[2]The corpora are from the PASCAL Challenge on Grammar Induction (`http://wiki.cs.ox.ac.uk/InducingLinguisticStructure/SharedTask`).

for each language, but on average it has better performance than using any fixed value of σ and hence is useful when the optimal value of σ is hard to identify.

5 Related Work

Deterministic annealing (DA) (Rose, 1998; Smith and Eisner, 2004) also extends the standard EM algorithm by exponentiating the posterior probabilities of the hidden variables in the E-step. However, the goal of DA is to improve the optimization of a non-concave objective function, which is achieved by setting the exponent in the E-step to a value *close to 0*, so that the distribution of the hidden variables becomes *nearly uniform* and the objective function becomes almost concave and therefore easy to optimize; this exponent is then gradually *increased to 1* to optimize the original objective function. In contrast, the goal of unambiguity regularization is to bias learning in favor of unambiguous grammars, which is achieved by setting the exponent in the E-step (i.e., $\frac{1}{1-\sigma}$ in Eq.1) to a value *larger than 1*, so that the distribution of the hidden variables becomes *less uniform* (i.e., parses become less ambiguous); in our annealing approach, the exponent is *initialized to a very large value* (positive infinity in our experiment) to push the learner away from the ambiguous initial grammar, and then gradually *decreased to 1* to avoid inducing excessive unambiguity in the learned grammar. The empirical results of Smith and Eisner (2004) show that DA resulted in lower parsing accuracy compared with standard EM in unsupervised constituent parsing; and a "skew" posterior term had to be inserted into the E-step formulation of DA to boost its accuracy over that of standard EM. In contrast, the results of our experiments show that unambiguity regularization leads to significantly higher parsing accuracy than standard EM.

Unambiguity regularization is also related to the minimum entropy regularization framework for semi-supervised learning (Grandvalet and Bengio, 2005; Smith and Eisner, 2007), which tries to minimize the entropy of the class label or hidden variables on unlabeled data in addition to maximizing the likelihood of labeled data. However, entropy regularization is either motivated by the theoretical result that unlabeled data samples are informative when classes are well separated (Grandvalet and Bengio, 2005), or derived from the expected conditional log-likelihood (Smith and Eisner, 2007). In contrast, our approach is motivated by the observed unambiguity of natural language grammars. One implication of this difference is that if our approach is applied to semi-supervised learning, the regularization term would be applied to labeled sentences as well (by ignoring the labels) because the target grammar shall be unambiguous on all the training sentences.

The sparsity bias, which favors a grammar with fewer grammar rules, has been widely used in unsupervised grammar learning (Chen, 1995; Johnson et al., 2007; Gillenwater et al., 2010). Although a more sparse grammar is often less ambiguous, in general that is not always the case. We have shown that unambiguity regularization could lead to better performance than approaches utilizing the sparsity bias, and that the two types of biases can be applied together for further improvement in the learning performance.

6 Conclusion

We have introduced unambiguity regularization, a novel approach to unsupervised learning of probabilistic natural language grammars. It is based on the observation that natural language grammars are remarkably unambiguous in the sense that in parsing natural language sentences they tend to concentrate the probability mass to a tiny portion of all possible parses. By using posterior regularization, we incorporate an inductive bias into learning in favor of grammars that are unambiguous on natural language sentences. The resulting family of algorithms includes standard EM and Viterbi EM, as well as the softmax-EM algorithm which falls between standard EM and Viterbi EM. The softmax-EM algorithm can be implemented by adding a simple parameter exponentiation step into standard EM. In our experiments of unsupervised dependency grammar learning, we show that unambiguity regularization is beneficial to learning, and by incorporating regularization strength annealing and sparsity priors our approach outperforms the current state-of-the-art grammar learning algorithms. For future work, we plan to combine unambiguity regularization with

	Arabic	Basque	Czech	Danish	Dutch	English	Portuguese	Slovene	Swedish
$\sigma = 0$ (standard EM)	27.4	32.1	27.8	35.6	29.4	34.9	23.7	**30.6**	31.9
$\sigma = 0.25$	30.6	39.3	27.2	35.2	30.9	**40.3**	**27.7**	23.8	**42.0**
$\sigma = 0.5$	**32.6**	40.6	**33.0**	**37.4**	32.7	38.8	27.5	15.3	29.3
$\sigma = 0.75$	31.6	**41.8**	16.1	36.0	**35.1**	38.8	26.2	15.1	32.7
$\sigma = 1$ (Viterbi EM)	29.6	39.8	28.6	33.6	28.0	39.4	27.3	14.6	37.2
UR-Annealing	26.7	41.6	**39.3**	34.1	**43.1**	**47.8**	26.4	16.4	**46.0**

Table 3: The dependency accuracies (on sentences of all lengths in the testing corpus) of grammars learned by our approach from the corpora of the following languages: Arabic (Hajič et al., 2004), Basque (Aduriz et al., 2003), Czech (Hajič et al., 2000), Danish (Buch-Kromann et al., 2007), Dutch (Beek et al., 2002), English, Portuguese (Afonso et al., 2002), Slovene (Erjavec et al., 2010), Swedish (Nivre et al., 2006).

other types of priors and regularizations for unsupervised grammar learning, to apply it to more advanced grammar models, and to explore alternative formulations of unambiguity regularization.

Acknowledgement

The work of Kewei Tu was supported at Iowa State University in part by a research assistantship from the Iowa State University Center for Computational Intelligence, Learning, and Discovery, and at University of California, Los Angeles by the DARPA grant FA 8650-11-1-7149. The work of Vasant Honavar was supported by the National Science Foundation, while working at the Foundation. Any opinion, finding, and conclusions contained in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Diaz de Ilarraza, A. Garmendia, , and M. Oronoz. 2003. Construction of a basque dependency treebank. In *Proc. of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*.

Susana Afonso, Eckhard Bick, Renato Haber, and Diana Santos. 2002. "floresta sintá(c)tica": a treebank for Portuguese. In *Proceedings of the 3rd Intern. Conf. on Language Resources and Evaluation (LREC)*, pages 1968–1703.

J. K. Baker. 1979. Trainable grammars for speech recognition. In *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*.

Van Der Beek, G. Bouma, R. Malouf, G. Van Noord, and Rijksuniversiteit Groningen. 2002. The alpino dependency treebank. In *In Computational Linguistics in the Netherlands (CLIN*, pages 1686–1691.

Phil Blunsom and Trevor Cohn. 2010. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1204–1213, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthias Buch-Kromann, Jürgen Wedekind, , and Jakob Elming. 2007. The copenhagen danish-english dependency treebank v. 2.0. http://www.buch-kromann.dk/matthias/cdt2.0/.

Stanley F. Chen. 1995. Bayesian grammar induction for language modeling. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*.

Shay B. Cohen and Noah A. Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *HLT-NAACL*, pages 74–82.

Shay B. Cohen, Kevin Gimpel, and Noah A. Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *NIPS*, pages 321–328.

Tomaz Erjavec, Darja Fiser, Simon Krek, and Nina Ledinek. 2010. The jos linguistically tagged corpus of slovene. In *LREC*.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2007. The infinite tree. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 272–279. Association for Computational Linguistics, June.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049.

Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *ACL '10: Proceedings of the ACL 2010 Conference Short Papers*, pages 194–199, Morristown, NJ, USA. Association for Computational Linguistics.

Yves Grandvalet and Yoshua Bengio. 2005. Semi-supervised learning by entropy minimization. In

Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 529–536. MIT Press, Cambridge, MA.

Jan Hajič, Alena Böhmová, Eva Hajičová, and Barbora Vidová-Hladká. 2000. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer.

Jan Hajič, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. 2004. Prague arabic dependency treebank: Development in data and tools. In *In Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*, pages 110–117.

William P. Headden, III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *HLT-NAACL*, pages 101–109.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for pcfgs via markov chain monte carlo. In *HLT-NAACL*, pages 139–146.

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

Kenichi Kurihara and Taisuke Sato. 2004. An application of the variational Bayesian approach to probabilistic contextfree grammars. In *IJCNLP-04 Workshop beyond shallow analyses*.

K. Lari and S. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–36.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. 2007. The infinite pcfg using hierarchical Dirichlet processes. In *Proceedings of EMNLP-CoNLL*, pages 688–697.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006), May 24-26, 2006, Genoa, Italy*, pages 1392–1395. European Language Resource Association, Paris.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440, Morristown, NJ, USA. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2011. Sum-product networks : A new deep architecture. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI)*.

Kenneth Rose. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of the IEEE*, pages 2210–2239.

Noah A. Smith and Jason Eisner. 2004. Annealing techniques for unsupervised statistical language learning. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

David A. Smith and Jason Eisner. 2007. Bootstrapping feature-rich dependency parsers with entropic priors. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 667–677, Prague, June.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010. Viterbi training improves unsupervised dependency parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 9–17, Stroudsburg, PA, USA. Association for Computational Linguistics.