Annealing Structural Bias in Multilingual Weighted Grammar Induction*

Noah A. Smith and **Jason Eisner**
Department of Computer Science / Center for Language and Speech Processing
Johns Hopkins University, Baltimore, MD 21218 USA
{nasmith,jason}@cs.jhu.edu

Abstract

We first show how a structural **locality bias** can improve the accuracy of state-of-the-art dependency grammar induction models trained by EM from unannotated examples (Klein and Manning, 2004). Next, by annealing the free parameter that controls this bias, we achieve further improvements. We then describe an alternative kind of structural bias, toward "broken" hypotheses consisting of partial structures over segmented sentences, and show a similar pattern of improvement. We relate this approach to contrastive estimation (Smith and Eisner, 2005a), apply the latter to grammar induction in six languages, and show that our new approach improves accuracy by 1–17% (absolute) over CE (and 8–30% over EM), achieving to our knowledge the best results on this task to date. Our method, **structural annealing**, is a general technique with broad applicability to hidden-structure discovery problems.

1 Introduction

Inducing a weighted context-free grammar from flat text is a hard problem. A common starting point for weighted grammar induction is the Expectation-Maximization (EM) algorithm (Dempster et al., 1977; Baker, 1979). EM's mediocre performance (Table 1) reflects two problems. First, it seeks to maximize likelihood, but a grammar that makes the training data likely does not necessarily assign a linguistically defensible syntactic structure. Second, the likelihood surface is not globally concave, and learners such as the EM algorithm can get trapped on local maxima (Charniak, 1993).

We seek here to capitalize on the intuition that, at least early in learning, the learner should search primarily for *string-local* structure, because most structure is local.[1] By penalizing dependencies between two words that are farther apart in the string, we obtain consistent improvements in accuracy of the learned model (§3).

We then explore how gradually *changing* δ over time affects learning (§4): we start out with a strong preference for short dependencies, then relax the preference. The new approach, **structural annealing**, often gives superior performance.

	model selection among values of λ and $\Theta^{(0)}$			
	worst	unsup.	sup.	oracle
German	19.8	19.8	54.4	54.4
English	21.8	41.6	41.6	42.0
Bulgarian	24.7	44.6	45.6	45.6
Mandarin	31.8	37.2	50.0	50.0
Turkish	32.1	41.2	48.0	51.4
Portuguese	35.4	37.4	42.3	43.0

Table 1: Baseline performance of EM-trained dependency parsing models: F_1 on non-$ attachments in test data, with various model selection conditions (3 initializers $\times$ 6 smoothing values). The languages are listed in decreasing order by the training set size. Experimental details can be found in the appendix.

An alternative structural bias is explored in §5. This approach views a sentence as a sequence of one or more yields of separate, independent trees. The points of segmentation are a hidden variable, and during learning all possible segmentations are entertained probabilistically. This allows the learner to accept hypotheses that explain the sentences as independent pieces.

In §6 we briefly review **contrastive estimation** (Smith and Eisner, 2005a), relating it to the new method, and show its performance alone and when augmented with structural bias.

2 Task and Model

In this paper we use a simple unlexicalized dependency model due to Klein and Manning (2004). The model is a probabilistic head automaton grammar (Alshawi, 1996) with a "split" form that renders it parseable in cubic time (Eisner, 1997).

Let $\mathbf{x} = \langle x_1, x_2, ..., x_n \rangle$ be the sentence. x_0 is a special "wall" symbol, $, on the left of every sentence. A tree $\mathbf{y}$ is defined by a pair of functions $\mathbf{y}_{left}$ and $\mathbf{y}_{right}$ (both $\{0, 1, 2, ..., n\} \rightarrow 2^{\{1,2,...,n\}}$) that map each word to its sets of left and right dependents, respectively. The graph is constrained to be a *projective* tree rooted at $: each word except $ has a single parent, and there are no cycles

*This work was supported by a Fannie and John Hertz Foundation fellowship to the first author and NSF ITR grant IIS-0313193 to the second author. The views expressed are not necessarily endorsed by the sponsors. We thank three anonymous COLING-ACL reviewers for comments.

[1] To be concrete, in the corpora tested here, 95% of dependency links cover $\leq$ 4 words (English, Bulgarian, Portuguese), $\leq$ 5 words (German, Turkish), $\leq$ 6 words (Mandarin).

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 569–576, Sydney, July 2006.

or crossing dependencies.[2] $\mathbf{y}_{left}(0)$ is taken to be empty, and $\mathbf{y}_{right}(0)$ contains the sentence's single head. Let y^i denote the subtree rooted at position i. The probability $P(y^i \mid x_i)$ of generating this subtree, given its head word x_i, is defined recursively:

$$\prod_{D \in \{left, right\}} p_{\text{stop}}(\text{stop} \mid x_i, D, [\mathbf{y}_D(i) = \emptyset]) \quad (1)$$
$$\times \prod_{j \in \mathbf{y}_D(i)} p_{\text{stop}}(\neg\text{stop} \mid x_i, D, \text{first}_{\mathbf{y}}(j))$$
$$\times p_{\text{child}}(x_j \mid x_i, D) \times P(y^j \mid x_j)$$

where $\text{first}_{\mathbf{y}}(j)$ is a predicate defined to be true iff x_j is the closest child (on either side) to its parent x_i. The probability of the entire tree is given by $p_\Theta(\mathbf{x}, \mathbf{y}) = P(y^0 \mid \$)$. The parameters Θ are the conditional distributions p_{stop} and p_{child}.

Experimental baseline: EM. Following common practice, we always replace words by part-of-speech (POS) tags before training or testing. We used the EM algorithm to train this model on POS sequences in six languages. Complete experimental details are given in the appendix. Performance with unsupervised and supervised model selection across different λ values in add-λ smoothing and three initializers $\Theta^{(0)}$ is reported in Table 1. The supervised-selected model is in the 40–55% F_1-accuracy range on directed dependency attachments. (Here $F_1 \approx$ precision $\approx$ recall; see appendix.) Supervised model selection, which uses a small annotated development set, performs almost as well as the oracle, but unsupervised model selection, which selects the model that maximizes likelihood on an *unannotated* development set, is often much worse.

3 Locality Bias among Trees

Hidden-variable estimation algorithms—including EM—typically work by iteratively manipulating the model parameters Θ to improve an objective function $F(\Theta)$. EM explicitly alternates between the computation of a *posterior* distribution over hypotheses, $p_\Theta(\mathbf{y} \mid \mathbf{x})$ (where $\mathbf{y}$ is any tree with yield $\mathbf{x}$), and computing a new parameter estimate Θ.[3]

[2] A projective parser could achieve perfect accuracy on our English and Mandarin datasets, $>$ 99% on Bulgarian, Turkish, and Portuguese, and $>$ 98% on German.

[3] For weighted grammar-based models, the posterior does not need to be explicitly represented; instead expectations under p_Θ are used to compute updates to Θ.

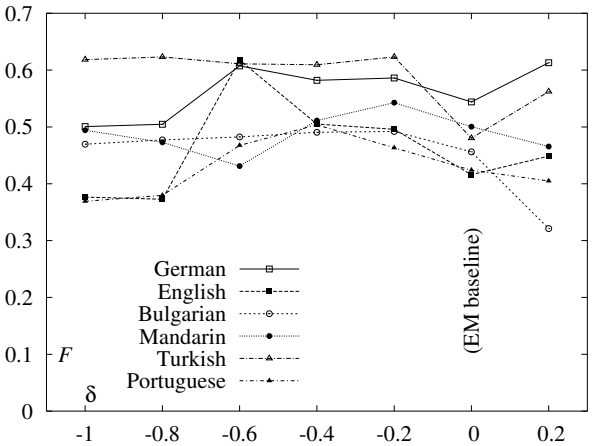

Figure 1: Test-set F_1 performance of models trained by EM with a **locality bias** at varying δ. Each curve corresponds to a different language and shows performance of supervised model selection *within* a given δ, across λ and $\Theta^{(0)}$ values. (See Table 3 for performance of models selected *across* δs.) We decode with $\delta = 0$, though we found that keeping the training-time value of δ would have had almost no effect. The EM baseline corresponds to $\delta = 0$.

One way to bias a learner toward local explanations is to penalize longer attachments. This was done for supervised parsing in different ways by Collins (1997), Klein and Manning (2003), and McDonald et al. (2005), all of whom considered intervening material or coarse distance classes when predicting children in a tree. Eisner and Smith (2005) achieved speed and accuracy improvements by modeling distance directly in a ML-estimated (deficient) generative model.

Here we use *string distance* to measure the length of a dependency link and consider the inclusion of a sum-of-lengths feature in the probabilistic model, for learning only. Keeping our original model, we will simply multiply into the probability of each tree another factor that penalizes long dependencies, giving:

$$p'_\Theta(\mathbf{x}, \mathbf{y}) \propto p_\Theta(\mathbf{x}, \mathbf{y}) \cdot e^{\left(\delta \sum_{i=1}^{n} \sum_{j \in \mathbf{y}(i)} |i - j|\right)} \quad (2)$$

where $\mathbf{y}(i) = \mathbf{y}_{left}(i) \cup \mathbf{y}_{right}(i)$. Note that if $\delta = 0$, we have the original model. As $\delta \to -\infty$, the new model p'_Θ will favor parses with shorter dependencies. The dynamic programming algorithms remain the same as before, with the appropriate $e^{\delta|i-j|}$ factor multiplied in at each attachment between x_i and x_j. Note that when $\delta = 0$, $p'_\Theta \equiv p_\Theta$.

Experiment. We applied a locality bias to the same dependency model by setting δ to different

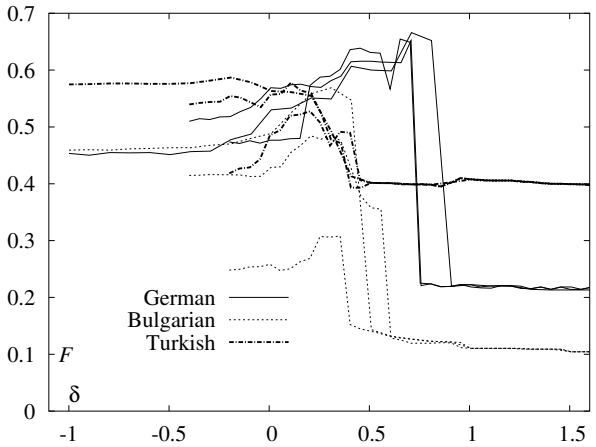

Figure 2: Test-set F_1 performance of models trained by EM with **structural annealing** on the distance weight δ. Here we show performance with add-10 smoothing, the all-zero initializer, for three languages with three different initial values δ_0. Time progresses from left to right. Note that it is generally best to start at $\delta_0 \ll 0$; note also the importance of picking the right point on the curve to stop. See Table 3 for performance of models selected across smoothing, initialization, starting, and stopping choices, in all six languages.

values in $[-1, 0.2]$ (see Eq. 2). The same initializers $\Theta^{(0)}$ and smoothing conditions were tested. Performance of supervised model selection among models trained at different δ values is plotted in Fig. 1. When a model is selected across *all* conditions (3 initializers $\times$ 6 smoothing values $\times$ 7 δs) using annotated development data, performance is notably better than the EM baseline using the same selection procedure (see Table 3, second column).

4 Structural Annealing

The central idea of this paper is to gradually *change* (anneal) the bias δ. Early in learning, local dependencies are emphasized by setting $\delta \ll 0$. Then δ is iteratively increased and training repeated, using the last learned model to initialize.

This idea bears a strong similarity to **deterministic annealing** (DA), a technique used in clustering and classification to smooth out objective functions that are piecewise constant (hence discontinuous) or bumpy (non-concave) (Rose, 1998; Ueda and Nakano, 1998). In unsupervised learning, DA iteratively re-estimates parameters like EM, but begins by requiring that the entropy of the posterior $p_\Theta(\mathbf{y} \mid \mathbf{x})$ be maximal, then gradually relaxes this entropy constraint. Since entropy is concave in Θ, the initial task is easy (maximize a concave, continuous function). At each step the optimization task becomes more difficult, but the initializer is given by the previous step and, in practice, tends to be close to a good local maximum of the more difficult objective. By the last iteration the objective is the same as in EM, but the annealed search process has acted like a good initializer. This method was applied with some success to grammar induction models by Smith and Eisner (2004).

In this work, instead of imposing constraints on the entropy of the model, we manipulate bias toward local hypotheses. As δ increases, we penalize long dependencies less. We call this **structural annealing**, since we are varying the strength of a soft constraint (bias) on structural hypotheses. In structural annealing, the final objective would be the same as EM if our final δ, $\delta_f = 0$, but we found that annealing farther ($\delta_f > 0$) works much better.[4]

Experiment: Annealing δ. We experimented with annealing schedules for δ. We initialized at $\delta_0 \in \{-1, -0.4, -0.2\}$, and increased δ by 0.1 (in the first case) or 0.05 (in the others) up to $\delta_f = 3$. Models were trained to convergence at each δ-epoch. Model selection was applied over the same initialization and regularization conditions as before, δ_0, and also over the choice of δ_f, with stopping allowed at any stage along the δ trajectory.

Trajectories for three languages with three different δ_0 values are plotted in Fig. 2. Generally speaking, $\delta_0 \ll 0$ performs better. There is consistently an early increase in performance as δ increases, but the stopping δ_f matters tremendously. Selected annealed-δ models surpass EM in all six languages; see the third column of Table 3. Note that structural annealing does not always outperform fixed-δ training (English and Portuguese). This is because we only tested a few values of δ_0, since annealing requires longer runtime.

5 Structural Bias via Segmentation

A related way to focus on local structure early in learning is to *broaden* the set of hypotheses to include *partial* parse structures. If $\mathbf{x} = \langle x_1, x_2, ..., x_n \rangle$, the standard approach assumes that $\mathbf{x}$ corresponds to the vertices of a single dependency tree. Instead, we entertain every hypothesis in which $\mathbf{x}$ is a *sequence* of yields from *separate*, independently-generated trees. For example, $\langle x_1, x_2, x_3 \rangle$ is the yield of one tree, $\langle x_4, x_5 \rangle$ is the

[4] The reader may note that $\delta_f > 0$ actually corresponds to a bias toward *longer* attachments. A more apt description in the context of annealing is to say that during early stages the learner starts liking local attachments too much, and we need to exaggerate δ to "coax" it to new hypotheses. See Fig. 2.

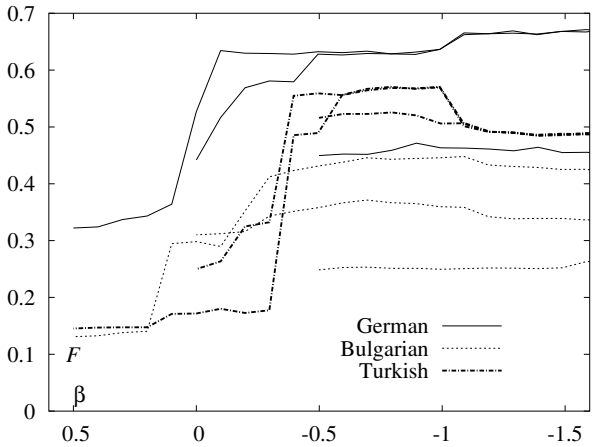

Figure 3: Test-set F_1 performance of models trained by EM with **structural annealing** on the breakage weight β. Here we show performance with add-10 smoothing, the all-zero initializer, for three languages with three different initial values β_0. Time progresses from left (large β) to right. See Table 3 for performance of models selected across smoothing, initialization, and stopping choices, in all six languages.

yield of a second, and $\langle x_6, ..., x_n \rangle$ is the yield of a third. One extreme hypothesis is that $\mathbf{x}$ is n single-node trees. At the other end of the spectrum is the original set of hypotheses—full trees on $\mathbf{x}$. Each has a nonzero probability.

Segmented analyses are intermediate representations that may be helpful for a learner to use to formulate notions of probable local structure, without committing to full trees.[5] We only *allow* unobserved breaks, never positing a hard segmentation of the training sentences. Over time, we increase the bias against broken structures, forcing the learner to commit most of its probability mass to full trees.

5.1 Vine Parsing

At first glance broadening the hypothesis space to entertain all 2^{n-1} possible segmentations may seem expensive. In fact the dynamic programming computation is almost the same as summing or maximizing over connected dependency trees. For the latter, we use an inside-outside algorithm that computes a score for every parse tree by computing the scores of *items*, or partial structures, through a bottom-up process. Smaller items are built first, then assembled using a set of rules defining how larger items can be built.[6]

Now note that any *sequence* of partial trees over $\mathbf{x}$ can be constructed by combining the same items into trees. The only difference is that we are willing to consider unassembled sequences of these partial trees as hypotheses, in addition to the fully connected trees. One way to accomplish this in terms of $\mathbf{y}_{right}(0)$ is to say that the root, \$, is allowed to have multiple children, instead of just one. Here, these children are independent of each other (e.g., generated by a unigram Markov model). In supervised dependency parsing, Eisner and Smith (2005) showed that imposing a hard constraint on the whole structure—specifically that each non-\$ dependency arc cross fewer than k words—can give guaranteed $O(nk^2)$ runtime with little to no loss in accuracy (for simple models). This constraint could lead to highly contrived parse trees, or none at all, for some sentences—both are avoided by the allowance of segmentation into a sequence of trees (each attached to \$). The construction of the "vine" (sequence of \$'s children) takes only $O(n)$ time once the chart has been assembled.

Our broadened hypothesis model is a probabilistic vine grammar with a unigram model over \$'s children. We allow (but do not require) segmentation of sentences, where each independent child of \$ is the root of one of the segments. We do not impose any constraints on dependency length.

5.2 Modeling Segmentation

Now the total probability of an n-length sentence $\mathbf{x}$, marginalizing over its hidden structures, sums up not only over trees, but over segmentations of $\mathbf{x}$. For completeness, we must include a probability model over the number of trees generated, which could be anywhere from 1 to n. The model over the number T of trees given a sentence of length n will take the following log-linear form:

$$P(T = t \mid n) = e^{t\beta} \Big/ \sum_{i=1}^{n} e^{i\beta}$$

where $\beta \in \mathbb{R}$ is the sole parameter. When $\beta = 0$, every value of T is equally likely. For $\beta \ll 0$, the model prefers larger structures with few breaks. At the limit ($\beta \to -\infty$), we achieve the standard learning setting, where the model must explain $\mathbf{x}$ using a single tree. We start however at $\beta \gg 0$, where the model prefers smaller trees with more breaks, in the limit preferring each word in $\mathbf{x}$ to be its own tree. We could describe "brokenness" as a feature in the model whose weight, β, is chosen extrinsically (and time-dependently), rather than empirically—just as was done with δ.

[5] See also work on partial parsing as a task in its own right: Hindle (1990) *inter alia*.

[6] See Eisner and Satta (1999) for the relevant algorithm used in the experiments.

		model selection among values of σ^2 and $\Theta^{(0)}$			
		worst	unsup.	sup.	oracle
Ger.	DORT1	32.5	**59.3**	**63.4**	63.4
	LENGTH	30.5	**56.4**	**57.3**	57.8
Eng.	DORT1	20.9	**56.6**	**57.4**	57.4
	LENGTH	29.1	37.2	**46.2**	46.2
Bul.	DORT1	19.4	26.0	40.5	43.1
	LENGTH	25.1	35.3	38.3	38.3
Man.	DORT1	9.4	24.2	41.1	41.1
	LENGTH	13.7	17.9	26.2	26.2
Tur.	DORT1	7.3	38.6	**58.2**	58.2
	LENGTH	21.5	34.1	**55.5**	55.5
Por.	DORT1	35.0	**59.8**	**71.8**	71.8
	LENGTH	30.8	33.6	33.6	33.6

Table 2: Performance of CE on test data, for different neighborhoods and with different levels of regularization. Boldface marks scores better than EM-trained models selected the same way (Table 1). The score is the F_1 measure on non-\$ attachments.

Annealing β resembles the popular **bootstrapping** technique (Yarowsky, 1995), which starts out aiming for high precision, and gradually improves coverage over time. With strong bias ($\beta \gg 0$), we seek a model that maintains high dependency precision on (non-\$) attachments by attaching most tags to \$. Over time, as this is iteratively weakened ($\beta \to -\infty$), we hope to improve coverage (dependency recall). Bootstrapping was applied to syntax learning by Steedman et al. (2003). Our approach differs in being able to remain partly agnostic about each tag's true parent (e.g., by giving 50% probability to attaching to \$), whereas Steedman et al. make a hard decision to retrain on a whole *sentence* fully or leave it out fully. In earlier work, Brill and Marcus (1992) adopted a "local first" iterative merge strategy for discovering phrase structure.

Experiment: Annealing β. We experimented with different annealing schedules for β. The initial value of β, β_0, was one of $\{-\frac{1}{2}, 0, \frac{1}{2}\}$. After EM training, β was diminished by $\frac{1}{10}$; this was repeated down to a value of $\beta_f = -3$. Performance after training at each β value is shown in Fig. 3.[7] We see that, typically, there is a sharp increase in performance somewhere during training, which typically lessens as $\beta \to -\infty$. Starting β too high can also damage performance. This method, then, is not robust to the choice of λ, β_0, or β_f, nor does it always do as well as annealing δ, although considerable gains are possible; see the fifth column of Table 3.

By testing models trained with a *fixed* value of β (for values in $[-1, 1]$), we ascertained that the performance improvement is due largely to annealing, not just the injection of segmentation bias (fourth vs. fifth column of Table 3).[8]

6 Comparison and Combination with Contrastive Estimation

Contrastive estimation (CE) was recently introduced (Smith and Eisner, 2005a) as a class of alternatives to the likelihood objective function locally maximized by EM. CE was found to outperform EM on the task of focus in this paper, when applied to English data (Smith and Eisner, 2005b). Here we review the method briefly, show how it performs across languages, and demonstrate that it can be combined effectively with structural bias.

Contrastive training defines for each example $\mathbf{x}_i$ a class of presumably poor, but similar, instances called the "neighborhood," $\mathcal{N}(\mathbf{x}_i)$, and seeks to maximize

$$\begin{aligned} C_{\mathcal{N}}(\Theta) &= \sum_i \log p_\Theta(\mathbf{x}_i \mid \mathcal{N}(\mathbf{x}_i)) \\ &= \sum_i \log \frac{\sum_{\mathbf{y}} p_\Theta(\mathbf{x}_i, \mathbf{y})}{\sum_{\mathbf{x}' \in \mathcal{N}(\mathbf{x}_i)} \sum_{\mathbf{y}} p_\Theta(\mathbf{x}', \mathbf{y})} \end{aligned}$$

At this point we switch to a log-linear (rather than stochastic) parameterization of the same weighted grammar, for ease of numerical optimization. All this means is that Θ (specifically, p_{stop} and p_{child} in Eq. 1) is now a set of nonnegative weights rather than probabilities.

Neighborhoods that can be expressed as finite-state lattices built from $\mathbf{x}_i$ were shown to give significant improvements in dependency parser quality over EM. Performance of CE using two of those neighborhoods on the current model and datasets is shown in Table 2.[9] 0-mean diagonal Gaussian smoothing was applied, with different variances, and model selection was applied over smoothing conditions and the same initializers as

[7] Performance measures are given using a *full* parser that finds the single best parse of the sentence with the learned parsing parameters. Had we decoded with a *vine* parser, we would see a precision$\searrow$, recall$\nearrow$ curve as β decreased.

[8] In principle, segmentation can be combined with the locality bias in §3 (δ). In practice, we found that this usually under-performed the EM baseline.

[9] We experimented with DELETE1, TRANSPOSE1, DELETEORTRANSPOSE1, and LENGTH. To conserve space we show only the latter two, which tend to perform best.

	EM	fixed δ	annealed δ	fixed β	annealed β	CE	fixed δ + CE
		δ	$\delta_0 \rightarrow \delta_f$	β	$\beta_0 \rightarrow \beta_f$	$\mathcal{N}$	$\mathcal{N}, \delta$
German	54.4	61.3 0.2	**70.0** -0.4 → 0.4	66.2 0.4	**68.9** 0.5 → -2.4	63.4 DORT1	63.8 DORT1, -0.2
English	41.6	**61.8** -0.6	53.8 -0.4 → 0.3	55.6 0.2	58.4 0.5 → 0.0	57.4 DORT1	**63.5** DORT1, -0.4
Bulgarian	45.6	49.2 -0.2	**58.3** -0.4 → 0.2	47.3 -0.2	56.5 0 → -1.7	40.5 DORT1	–
Mandarin	50.0	51.1 -0.4	**58.0** -1.0 → 0.2	38.0 0.2	**57.2** 0.5 → -1.4	43.4 DEL1	–
Turkish	48.0	**62.3** -0.2	**62.4** -0.2 → -0.15	53.6 -0.2	59.4 0.5 → -0.7	58.2 DORT1	61.8 DORT1, -0.6
Portuguese	42.3	50.4 -0.4	50.2 -0.4 → -0.1	51.5 0.2	62.7 0.5 → -0.5	**71.8** DORT1	**72.6** DORT1, -0.2

Table 3: Summary comparing models trained in a variety of ways with some relevant hyperparameters. Supervised model selection was applied in all cases, including EM (see the appendix). Boldface marks the best performance overall and trials that this performance did not significantly surpass under a sign test (i.e., $p \not< 0.05$). The score is the F_1 measure on non-\$ attachments. The fixed δ + CE condition was tested only for languages where CE improved over EM.

before. Four of the languages have at least one effective CE condition, supporting our previous English results (Smith and Eisner, 2005b), but CE was harmful for Bulgarian and Mandarin. Perhaps better neighborhoods exist for these languages, or there is some ideal neighborhood that would perform well for all languages.

Our approach of allowing broken trees (§5) is a natural extension of the CE framework. Contrastive estimation views learning as a process of moving posterior probability mass *from* (implicit) negative examples *to* (explicit) positive examples. The positive evidence, as in MLE, is taken to be the observed data. As originally proposed, CE allowed a redefinition of the implicit negative evidence from "all other sentences" (as in MLE) to "sentences like $\mathbf{x}_i$, but perturbed." Allowing segmentation of the training sentences redefines the positive *and* negative evidence. Rather than moving probability mass only to full analyses of the training example $\mathbf{x}_i$, we also allow probability mass to go to partial analyses of $\mathbf{x}_i$.

By injecting a bias ($\delta \neq 0$ or $\beta > -\infty$) among tree hypotheses, however, we have gone beyond the CE framework. We have added features to the tree model (dependency length-sum, number of breaks), whose weights we extrinsically manipulate over time to impose locality bias $C_{\mathcal{N}}$ and improve search on $C_{\mathcal{N}}$. Another idea, not explored here, is to change the contents of the neighborhood $\mathcal{N}$ over time.

Experiment: Locality Bias within CE. We combined CE with a fixed-δ locality bias for neighborhoods that were successful in the earlier CE experiment, namely DELETEORTRANSPOSE1 for German, English, Turkish, and Portuguese. Our results, shown in the seventh column of Table 3, show that, in all cases except Turkish, the combination improves over either technique on its own. We leave exploration of structural annealing with CE to future work.

Experiment: Segmentation Bias within CE. For (language, $\mathcal{N}$) pairs where CE was effective, we trained models using CE with a fixed-β segmentation model. Across conditions ($\beta \in [-1, 1]$), these models performed very badly, hypothesizing extremely local parse trees: typically over 90% of dependencies were length 1 and pointed in the same direction, compared with the 60–70% length-1 rate seen in gold standards. To understand why, consider that the CE goal is to maximize the score of a sentence *and* all its segmentations while minimizing the scores of neighborhood sentences and their segmentations. An n-gram model can accomplish this, since the same n-grams are present in all segmentations of $\mathbf{x}$, and (some) different n-grams appear in $\mathcal{N}(\mathbf{x})$ (for LENGTH and DELETEORTRANSPOSE1). A bigram-like model that favors monotone branching, then, is not a bad choice for a CE learner that must account for segmentations of $\mathbf{x}$ and $\mathcal{N}(\mathbf{x})$.

Why doesn't CE *without* segmentation resort to n-gram-like models? Inspection of models trained using the standard CE method (no segmentation) with transposition-based neighborhoods TRANSPOSE1 and DELETEORTRANSPOSE1 *did* have high rates of length-1 dependencies, while the poorly-performing DELETE1 models found *low* length-1 rates. This suggests that a bias toward locality ("n-gram-ness") is built into the former neighborhoods, and may partly explain why CE works when it does. We achieved a similar locality bias in the likelihood framework when we broadened the hypothesis space, but doing so under CE *over*-focuses the model on local structures.

7 Error Analysis

We compared errors made by the selected EM condition with the best overall condition, for each language. We found that the number of corrected attachments always outnumbered the number of new errors by a factor of two or more.

Further, the new models are not getting better by merely reversing the *direction* of links made by EM; undirected accuracy also improved significantly under a sign test ($p < 10^{-6}$), across all six languages. While the most common corrections were to nouns, these account for only 25–41% of corrections, indicating that corrections are not "all of the same kind."

Finally, since more than half of corrections in every language involved reattachment to a noun or a verb (content word), we believe the improved models to be getting closer than EM to the deeper semantic relations between words that, ideally, syntactic models should uncover.

8 Future Work

One weakness of all recent weighted grammar induction work—including Klein and Manning (2004), Smith and Eisner (2005b), and the present paper—is a sensitivity to hyperparameters, including smoothing values, choice of $\mathcal{N}$ (for CE), and annealing schedules—not to mention initialization. This is quite observable in the results we have presented. An obstacle for unsupervised learning in general is the need for automatic, efficient methods for model selection. For annealing, inspiration may be drawn from continuation methods; see, e.g., Elidan and Friedman (2005). Ideally one would like to select values simultaneously for many hyperparameters, perhaps using a small annotated corpus (as done here), extrinsic figures of merit on successful learning trajectories, or plausibility criteria (Eisner and Karakos, 2005).

Grammar induction serves as a tidy example for structural annealing. In future work, we envision that other kinds of structural bias and annealing will be useful in other difficult learning problems where hidden structure is required, including machine translation, where the structure can consist of word correspondences or phrasal or recursive syntax with correspondences. The technique bears some similarity to the estimation methods described by Brown et al. (1993), which started by estimating simple models, using each model to seed the next.

9 Conclusion

We have presented a new unsupervised parameter estimation method, structural annealing, for learning hidden structure that biases toward simplicity and gradually weakens (anneals) the bias over time. We applied the technique to weighted dependency grammar induction and achieved a significant gain in accuracy over EM and CE, raising the state-of-the-art across six languages from 42–54% to 58–73% accuracy.

References

S. Afonso, E. Bick, R. Haber, and D. Santos. 2002. Floresta sintá(c)tica: a treebank for Portuguese. In *Proc. of LREC*.

H. Alshawi. 1996. Head automata and bilingual tiling: Translation with minimal representations. In *Proc. of ACL*.

N. B. Atalay, K. Oflazer, and B. Say. 2003. The annotation process in the Turkish treebank. In *Proc. of LINC*.

J. K. Baker. 1979. Trainable grammars for speech recognition. In *Proc. of the Acoustical Society of America*.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER Treebank. In *Proc. of Workshop on Treebanks and Linguistic Theories*.

E. Brill and M. Marcus. 1992. Automatically acquiring phrase structure using distributional analysis. In *Proc. of DARPA Workshop on Speech and Natural Language*.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proc. of CoNLL*.

E. Charniak. 1993. *Statistical Language Learning*. MIT Press.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of ACL*.

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood estimation from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38.

J. Eisner and D. Karakos. 2005. Bootstrapping without the boot. In *Proc. of HLT-EMNLP*.

J. Eisner and G. Satta. 1999. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proc. of ACL*.

J. Eisner and N. A. Smith. 2005. Parsing with soft and hard constraints on dependency length. In *Proc. of IWPT*.

J. Eisner. 1997. Bilexical grammars and a cubic-time probabilistic parser. In *Proc. of IWPT*.

G. Elidan and N. Friedman. 2005. Learning hidden variable networks: the information bottleneck approach. *Journal of Machine Learning Research*, 6:81–127.

D. Hindle. 1990. Noun classification from predicate-argument structure. In *Proc. of ACL*.

D. Klein and C. D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proc. of ACL*.

D. Klein and C. D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *NIPS 15*.

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

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.
R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proc. of ACL*.
K. Oflazer, B. Say, D. Z. Hakkani-Tür, and G. Tür. 2003. Building a Turkish treebank. In A. Abeille, editor, *Building and Exploiting Syntactically-Annotated Corpora*. Kluwer.
K. Rose. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc. of the IEEE*, 86(11):2210–2239.
K. Simov and P. Osenova. 2003. Practical annotation scheme for an HPSG treebank of Bulgarian. In *Proc. of LINC*.
K. Simov, G. Popova, and P. Osenova. 2002. HPSG-based syntactic treebank of Bulgarian (BulTreeBank). In A. Wilson, P. Rayson, and T. McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*, pages 135–42. Lincom-Europa.
K. Simov, P. Osenova, A. Simov, and M. Kouylekov. 2004. Design and implementation of the Bulgarian HPSG-based Treebank. *Journal of Research on Language and Computation*, 2(4):495–522.
N. A. Smith and J. Eisner. 2004. Annealing techniques for unsupervised statistical language learning. In *Proc. of ACL*.
N. A. Smith and J. Eisner. 2005a. Contrastive estimation: Training log-linear models on unlabeled data. In *Proc. of ACL*.
N. A. Smith and J. Eisner. 2005b. Guiding unsupervised grammar induction using contrastive estimation. In *Proc. of IJCAI Workshop on Grammatical Inference Applications*.
M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proc. of EACL*.
N. Ueda and R. Nakano. 1998. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282.
N. Xue, F. Xia, F.-D. Chiou, and M. Palmer. 2004. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 10(4):1–30.
D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of ACL*.

A Experimental Setup

Following the usual conventions (Klein and Manning, 2002), our experiments use treebank POS sequences of length ≤ 10, stripped of words and punctuation. For smoothing, we apply add-λ, with six values of λ (in CE trials, we use a 0-mean diagonal Gaussian prior with five different values of σ^2). Our training datasets are:

- 8,227 **German** sentences from the TIGER Treebank (Brants et al., 2002),
- 5,301 **English** sentences from the WSJ Penn Treebank (Marcus et al., 1993),
- 4,929 **Bulgarian** sentences from the BulTreeBank (Simov et al., 2002; Simov and Osenova, 2003; Simov et al., 2004),
- 2,775 **Mandarin** sentences from the Penn Chinese Treebank (Xue et al., 2004),
- 2,576 **Turkish** sentences from the METU-Sabanci Treebank (Atalay et al., 2003; Oflazer et al., 2003), and
- 1,676 **Portuguese** sentences from the Bosque portion of the Floresta Sintá(c)tica Treebank (Afonso et al., 2002).

The Bulgarian, Turkish, and Portuguese datasets come from the CoNLL-X shared task (Buchholz and Marsi, 2006); we thank the organizers.

When comparing a hypothesized tree $\mathbf{y}$ to a gold standard $\mathbf{y}^*$, precision and recall measures are available. If every tree in the gold standard and every hypothesis tree is such that $|\mathbf{y}_{right}(0)| = 1$, then precision = recall = F_1, since $|\mathbf{y}| = |\mathbf{y}^*|$. $|\mathbf{y}_{right}(0)| = 1$ for all hypothesized trees in this paper, but not all treebank trees; hence we report the F_1 measure. The test set consists of around 500 sentences (in each language).

Iterative training proceeds until either 100 iterations have passed, or the objective converges within a relative tolerance of $\epsilon = 10^{-5}$, whichever occurs first.

Models trained at different hyperparameter settings and with different initializers are selected using a 500-sentence development set. *Unsupervised* model selection means the model with the highest training objective value on the development set was chosen. *Supervised* model selection chooses the model that performs best on the annotated development set. (*Oracle* and *worst* model selection are chosen based on performance on the test data.)

We use three initialization methods. We run a single special E step (to get expected counts of model events) then a single M step that renormalizes to get a probabilistic model $\Theta^{(0)}$. In initializer 1, the E step scores each tree as follows (only connected trees are scored):

$$u(\mathbf{x}, \mathbf{y}_{left}, \mathbf{y}_{right}) = \prod_{i=1}^{n} \prod_{j \in \mathbf{y}(i)} \left(1 + \frac{1}{|i-j|}\right)$$

(Proper) expectations under these scores are computed using an inside-outside algorithm. Initializer 2 computes expected counts directly, without dynamic programming. For an n-length sentence, $p(\mathbf{y}_{right}(0) = \{i\}) = \frac{1}{n}$ and $p(j \in \mathbf{y}(i)) \propto \frac{1}{|i-j|}$. These are scaled by an appropriate constant for each sentence, then summed across sentences to compute expected event counts. Initializer 3 assumes a uniform distribution over hidden structures in the special E step by setting all log probabilities to zero.