Semi-supervised Learning of Dependency Parsers using Generalized Expectation Criteria

Gregory Druck
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
gdruck@cs.umass.edu

Gideon Mann
Google, Inc.
76 9th Ave.
New York, NY 10011
gideon.mann@gmail.com

Andrew McCallum
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
mccallum@cs.umass.edu

Abstract

In this paper, we propose a novel method for semi-supervised learning of non-projective log-linear dependency parsers using directly expressed linguistic prior knowledge (e.g. a *noun*'s parent is often a *verb*). Model parameters are estimated using a *generalized expectation* (GE) objective function that penalizes the mismatch between model predictions and linguistic expectation constraints. In a comparison with two prominent "unsupervised" learning methods that require indirect biasing toward the correct syntactic structure, we show that GE can attain better accuracy with as few as 20 intuitive constraints. We also present positive experimental results on longer sentences in multiple languages.

1 Introduction

Early approaches to parsing assumed a grammar provided by human experts (Quirk et al., 1985). Later approaches avoided grammar writing by learning the grammar from sentences explicitly annotated with their syntactic structure (Black et al., 1992). While such supervised approaches have yielded accurate parsers (Charniak, 2001), the syntactic annotation of corpora such as the Penn Treebank is extremely costly, and consequently there are few treebanks of comparable size.

As a result, there has been recent interest in unsupervised parsing. However, in order to attain reasonable accuracy, these methods have to be carefully biased towards the desired syntactic structure. This weak supervision has been encoded using priors and initializations (Klein and Manning, 2004; Smith, 2006), specialized models (Klein and Manning, 2004; Seginer, 2007; Bod, 2006), and implicit negative evidence (Smith, 2006). These indirect methods for leveraging prior knowledge can be cumbersome and unintuitive for a non-machine-learning expert.

This paper proposes a method for directly guiding the learning of dependency parsers with naturally encoded linguistic insights. *Generalized expectation* (GE) (Mann and McCallum, 2008; Druck et al., 2008) is a recently proposed framework for incorporating prior knowledge into the learning of conditional random fields (CRFs) (Lafferty et al., 2001). GE criteria express a preference on the value of a model expectation. For example, we know that "in English, when a *determiner* is directly to the left of a *noun*, the *noun* is usually the parent of the *determiner*". With GE we may add a term to the objective function that encourages a feature-rich CRF to match this expectation on unlabeled data, and in the process learn about related features. In this paper we use a non-projective dependency tree CRF (Smith and Smith, 2007).

While a complete exploration of linguistic prior knowledge for dependency parsing is beyond the scope of this paper, we provide several promising demonstrations of the proposed method. On the English WSJ10 data set, GE training outperforms two prominent unsupervised methods using only 20 constraints either elicited from a human or provided by an "oracle" simulating a human. We also present experiments on longer sentences in Dutch, Spanish, and Turkish in which we obtain accuracy comparable to supervised learning with tens to hundreds of complete parsed sentences.

2 Related Work

This work is closely related to the *prototype-driven* grammar induction method of Haghighi and Klein (2006), which uses prototype phrases to guide the EM algorithm in learning a PCFG. Direct comparison with this method is not possible because we are interested in dependency syntax rather than phrase structure syntax. However, the approach we advocate has several significant advantages. GE is more general than prototype-driven learning because GE constraints can be uncertain. Additionally prototype-driven grammar induction needs to be used in conjunction with other unsupervised methods (distributional similarity and CCM (Klein and Manning, 2004)) to attain reasonable accuracy, and is only evaluated on length 10 or less sentences with no lexical information. In contrast, GE uses only the provided constraints and unparsed sentences, and is used to train a feature-rich discriminative model.

Conventional semi-supervised learning requires parsed sentences. Kate and Mooney (2007) and McClosky et al. (2006) both use modified forms of self-training to bootstrap parsers from limited labeled data. Wang et al. (2008) combine a structured loss on parsed sentences with a least squares loss on unlabeled sentences. Koo et al. (2008) use a large unlabeled corpus to estimate cluster features which help the parser generalize with fewer examples. Smith and Eisner (2007) apply entropy regularization to dependency parsing. The above methods can be applied to small seed corpora, but McDonald[1] has criticized such methods as working from an unrealistic premise, as a significant amount of the effort required to build a treebank comes in the first 100 sentences (both because of the time it takes to create an appropriate rubric and to train annotators).

There are also a number of methods for unsupervised learning of dependency parsers. Klein and Manning (2004) use a carefully initialized and structured generative model (DMV) in conjunction with the EM algorithm to get the first positive results on unsupervised dependency parsing. As empirical evidence of the sensitivity of DMV to initialization, Smith (2006) (pg. 37) uses three different initializations, and only one, the method of Klein and Manning (2004), gives accuracy higher than 31% on the WSJ10 corpus (see Section 5). This initialization encodes the prior knowledge that long distance attachments are unlikely.

Smith and Eisner (2005) develop *contrastive estimation* (CE), in which the model is encouraged to move probability mass away from implicit negative examples defined using a carefully chosen neighborhood function. For instance, Smith (2006) (pg. 82) uses eight different neighborhood functions to estimate parameters for the DMV model. The best performing neighborhood function DEL1ORTRANS1 provides accuracy of 57.6% on WSJ10 (see Section 5). Another neighborhood, DEL1ORTRANS2, provides accuracy of 51.2%. The remaining six neighborhood functions provide accuracy below 50%. This demonstrates that constructing an appropriate neighborhood function can be delicate and challenging.

Smith and Eisner (2006) propose *structural annealing* (SA), in which a strong bias for local dependency attachments is enforced early in learning, and then gradually relaxed. This method is sensitive to the annealing schedule. Smith (2006) (pg. 136) use 10 annealing schedules in conjunction with three initializers. The best performing combination attains accuracy of 66.7% on WSJ10, but the worst attains accuracy of 32.5%.

Finally, Seginer (2007) and Bod (2006) approach unsupervised parsing by constructing novel syntactic models. The development and tuning of the above methods constitute the encoding of prior domain knowledge about the desired syntactic structure. In contrast, our framework provides a straightforward and explicit method for incorporating prior knowledge.

Ganchev et al. (2009) propose a related method that uses posterior constrained EM to learn a projective target language parser using only a source language parser and word alignments.

3 Generalized Expectation Criteria

Generalized expectation criteria (Mann and McCallum, 2008; Druck et al., 2008) are terms in a parameter estimation objective function that express a preference on the value of a model expectation. Let $\mathbf{x}$ represent input variables (i.e. a sentence) and $\mathbf{y}$ represent output variables (i.e. a parse tree). A generalized expectation term $\mathcal{G}(\lambda)$ is defined by a constraint function $G(\mathbf{y}, \mathbf{x})$ that returns a non-negative real value given input and output variables, an empirical distribution $\tilde{p}(\mathbf{x})$ over input variables (i.e. unlabeled data), a model distribution $p_\lambda(\mathbf{y}|\mathbf{x})$, and a score function S:

$$\mathcal{G}(\lambda) = S(E_{\tilde{p}(\mathbf{x})}[E_{p_\lambda(\mathbf{y}|\mathbf{x})}[G(\mathbf{y}, \mathbf{x})]]).$$

In this paper, we use a score function that is the squared difference of the model expectation of G and some target expectation $\tilde{G}$:

$$S_{sq} = -(\tilde{G} - E_{\tilde{p}(\mathbf{x})}[E_{p_\lambda(\mathbf{y}|\mathbf{x})}[G(\mathbf{y}, \mathbf{x})]])^2 \quad (1)$$

We can incorporate prior knowledge into the training of $p_\lambda(\mathbf{y}|\mathbf{x})$ by specifying the from of the constraint function G and the target expectation $\tilde{G}$.

[1] R. McDonald, personal communication, 2007

Importantly, G does not need to match a particular feature in the underlying model.

The complete objective function[2] includes multiple GE terms and a prior on parameters[3], $p(\lambda)$

$$\mathcal{O}(\lambda;\mathcal{D}) = p(\lambda) + \sum_{\mathcal{G}} \mathcal{G}(\lambda)$$

GE has been applied to logistic regression models (Mann and McCallum, 2007; Druck et al., 2008) and linear chain CRFs (Mann and McCallum, 2008). In the following sections we apply GE to non-projective CRF dependency parsing.

3.1 GE in General CRFs

We first consider an arbitrarily structured conditional random field (Lafferty et al., 2001) $p_\lambda(\mathbf{y}|\mathbf{x})$. We describe the CRF for non-projective dependency parsing in Section 3.2. The probability of an output $\mathbf{y}$ conditioned on an input $\mathbf{x}$ is

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\Big(\sum_j \lambda_j F_j(\mathbf{y},\mathbf{x})\Big),$$

where F_j are feature functions over the cliques of the graphical model and $Z(\mathbf{x})$ is a normalizing constant that ensures $p_\lambda(\mathbf{y}|\mathbf{x})$ sums to 1. We are interested in the expectation of constraint function $G(\mathbf{x},\mathbf{y})$ under this model. We abbreviate this *model expectation* as:

$$G_\lambda = E_{\tilde{p}(\mathbf{x})}[E_{p_\lambda(\mathbf{y}|\mathbf{x})}[G(\mathbf{y},\mathbf{x})]]$$

It can be shown that partial derivative of $\mathcal{G}(\lambda)$ using ${S_{sq}}$[4] with respect to model parameter λ_j is

$$\frac{\partial}{\partial \lambda_j}\mathcal{G}(\lambda) = 2(\tilde{G} - G_\lambda) \qquad (2)$$
$$\Big(E_{\tilde{p}(\mathbf{x})}\Big[E_{p_\lambda(\mathbf{y}|\mathbf{x})}\left[G(\mathbf{y},\mathbf{x})F_j(\mathbf{y},\mathbf{x})\right] - E_{p_\lambda(\mathbf{y}|\mathbf{x})}\left[G(\mathbf{y},\mathbf{x})\right]E_{p_\lambda(\mathbf{y}|\mathbf{x})}\left[F_j(\mathbf{y},\mathbf{x})\right]\Big]\Big).$$

Equation 2 has an intuitive interpretation. The first term (on the first line) is the difference between the model and target expectations. The second term (the rest of the equation) is the predicted covariance between the constraint function G and the model feature function F_j. Therefore, if the constraint is not satisfied, GE updates parameters for features that the model predicts are related to the constraint function.

If there are constraint functions G for all model feature functions F_j, and the target expectations $\tilde{G}$ are estimated from labeled data, then the globally optimal parameter setting under the GE objective function is equivalent to the maximum likelihood solution. However, GE does not require such a one-to-one correspondence between constraint functions and model feature functions. This allows bootstrapping of feature-rich models with a small number of prior expectation constraints.

3.2 Non-Projective Dependency Tree CRFs

We now define a CRF $p_\lambda(\mathbf{y}|\mathbf{x})$ for unlabeled, non-projective[5] dependency parsing. The tree $\mathbf{y}$ is represented as a vector of the same length as the sentence, where y_i is the index of the parent of word i. The probability of a tree $\mathbf{y}$ given sentence $\mathbf{x}$ is

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\Big(\sum_{i=1}^{n}\sum_j \lambda_j f_j(x_i, x_{y_i}, \mathbf{x})\Big),$$

where f_j are *edge-factored* feature functions that consider the child input (word, tag, or other feature), the parent input, and the rest of the sentence. This factorization implies that dependency decisions are independent conditioned on the input sentence $\mathbf{x}$ if $\mathbf{y}$ is a tree. Computing $Z_\mathbf{x}$ and the edge expectations needed for partial derivatives requires summing over all possible trees for $\mathbf{x}$.

By relating the sum of the scores of all possible trees to counting the number of spanning trees in a graph, it can be shown that $Z_\mathbf{x}$ is the determinant of the *Kirchoff matrix* K, which is constructed using the scores of possible edges. (McDonald and Satta, 2007; Smith and Smith, 2007). Computing the determinant takes $O(n^3)$ time, where n is the length of the sentence. To compute the marginal probability of a particular edge $k \rightarrow i$ (i.e. $y_i = k$), the score of any edge $k' \rightarrow i$ such that $k' \neq k$ is set to 0. The determinant of the resulting modified Kirchoff matrix $K_{k\rightarrow i}$ is then the sum of the scores of all trees that include the edge $k \rightarrow i$. The

[2] In general, the objective function could also include the likelihood of available labeled data, but throughout this paper we assume we have no parsed sentences.

[3] Throughout this paper we use a Gaussian prior on parameters with $\sigma^2 = 10$.

[4] In previous work, S was the KL-divergence from the target expectation. The partial derivative of the KL divergence score function includes the same covariance term as above but substitutes a different multiplicative term: $\tilde{G}/G_\lambda$.

[5] Note that we could instead define a CRF for projective dependency parse trees and use a variant of the inside outside algorithm for inference. We choose non-projective because it is the more general case.

marginal $p(y_i = k|\mathbf{x}; \theta)$ can be computed by dividing this score by $Z_{\mathbf{x}}$ (McDonald and Satta, 2007). Computing all edge expectations with this algorithm takes $O(n^5)$ time. Smith and Smith (2007) describe a more efficient algorithm that can compute all edge expectations in $O(n^3)$ time using the inverse of the Kirchoff matrix K^{-1}.

3.3 GE for Non-Projective Dependency Tree CRFs

While in general constraint functions G may consider multiple edges, in this paper we use edge-factored constraint functions. In this case $E_{p_\lambda(\mathbf{y}|\mathbf{x})}[G(\mathbf{y}, \mathbf{x})]E_{p_\lambda(\mathbf{y}|\mathbf{x})}[F_j(\mathbf{y}, \mathbf{x})]$, the second term of the covariance in Equation 2, can be computed using the edge marginal distributions $p_\lambda(y_i|\mathbf{x})$. The first term of the covariance $E_{p_\lambda(\mathbf{y}|\mathbf{x})}[G(\mathbf{y}, \mathbf{x})F_j(\mathbf{y}, \mathbf{x})]$ is more difficult to compute because it requires the marginal probability of two edges $p_\lambda(y_i, y_{i'}|\mathbf{x})$. It is important to note that the model p_λ is still edge-factored.

The sum of the scores of all trees that contain edges $k \rightarrow i$ and $k' \rightarrow i'$ can be computed by setting the scores of edges $j \rightarrow i$ such that $j \neq k$ and $j' \rightarrow i'$ such that $j' \neq k'$ to 0, and computing the determinant of the resulting modified Kirchoff matrix $K_{k \rightarrow i, k' \rightarrow i'}$. There are $O(n^4)$ pairs of possible edges, and the determinant computation takes time $O(n^3)$, so this naive algorithm takes $O(n^7)$ time.

An improved algorithm computes, for each possible edge $k \rightarrow i$, a modified Kirchoff matrix $K_{k \rightarrow i}$ that requires the presence of that edge. Then, the method of Smith and Smith (2007) can be used to compute the probability of every possible edge conditioned on the presence of $k \rightarrow i$, $p_\lambda(y_{i'} = k'|y_i = k, \mathbf{x})$, using $K^{-1}_{k \rightarrow i}$. Multiplying this probability by $p_\lambda(y_i = k|\mathbf{x})$ yields the desired two edge marginal. Because this algorithm pulls the $O(n^3)$ matrix operation out of the inner loop over edges, the run time is reduced to $O(n^5)$.

If it were possible to perform only one $O(n^3)$ matrix operation per sentence, then the gradient computation would take only $O(n^4)$ time, the time required to consider all pairs of edges. Unfortunately, there is no straightforward generalization of the method of Smith and Smith (2007) to the two edge marginal problem. Specifically, *Laplace expansion* generalizes to second-order matrix minors, but it is not clear how to compute second-order cofactors from the inverse Kirchoff matrix alone (c.f. (Smith and Smith, 2007)).

Consequently, we also propose an approximation that can be used to speed up GE training at the expense of a less accurate covariance computation. We consider different cases of the edges $k \rightarrow i$, and $k' \rightarrow i'$.

- $p_\lambda(y_i = k, y_{i'} = k'|\mathbf{x}) = 0$ when $i = i'$ and $k \neq k'$ (different parent for the same word), or when $i = k'$ and $k = i'$ (cycle), because these pairs of edges break the tree constraint.
- $p_\lambda(y_i = k, y_{i'} = k'|\mathbf{x}) = p_\lambda(y_i = k|\mathbf{x})$ when $i = i', k = k'$.
- $p_\lambda(y_i = k, y_{i'} = k'|\mathbf{x}) \approx p_\lambda(y_i = k|\mathbf{x}) p_\lambda(y_{i'} = k'|\mathbf{x})$ when $i \neq i'$ and $i \neq k'$ or $i' \neq k$ (different words, do not create a cycle). This approximation assumes that pairs of edges that do not fall into one of the above cases are conditionally independent given $\mathbf{x}$. This is not true because there are partial trees in which $k \rightarrow i$ and $k' \rightarrow i'$ can appear separately, but not together (for example if $i = k'$ and the partial tree contains $i' \rightarrow k$).

Using this approximation, the covariance for one sentence is approximately equal to

$$\sum_i^n E_{p_\lambda(y_i|\mathbf{x})}[f_j(x_i, x_{y_i}, \mathbf{x}) g(x_i, x_{y_i}, \mathbf{x})]$$
$$- \sum_i^n E_{p_\lambda(y_i|\mathbf{x})}[f_j(x_i, x_{y_i}, \mathbf{x})] E_{p_\lambda(y_i|\mathbf{x})}[g(x_i, x_{y_i}, \mathbf{x})]$$
$$- \sum_i^n \sum_k^n p_\lambda(y_i = k|\mathbf{x}) p_\lambda(y_k = i|\mathbf{x}) f_j(x_i, x_k, \mathbf{x}) g(x_k, x_i, \mathbf{x}).$$

Intuitively, the first and second terms compute a covariance over possible parents for a single word, and the third term accounts for cycles. Computing the above takes $O(n^3)$ time, the time required to compute single edge marginals. In this paper, we use the $O(n^5)$ exact method, though we find that the accuracy attained by approximate training is usually within 5% of the exact method.

If G is not edge-factored, then we need to compute a marginal over three or more edges, making exact training intractable. An appealing alternative to a similar approximation to the above would use loopy belief propagation to efficiently approximate the marginals (Smith and Eisner, 2008).

In this paper g is binary and normalized by its total count in the corpus. The expectation of g is then the probability that it indicates a true edge.

4 Linguistic Prior Knowledge

Training parsers using GE with the aid of linguists is an exciting direction for future work. In this paper, we use constraints derived from several basic types of linguistic knowledge.

One simple form of linguistic knowledge is the set of possible parent tags for a given child tag. This type of constraint was used in the development of a rule-based dependency parser (Debusmann et al., 2004). Additional information can be obtained from small grammar fragments. Haghighi and Klein (2006) provide a list of prototype phrase structure rules that can be augmented with dependencies and used to define constraints involving parent and child tags, surrounding or interposing tags, direction, and distance. Finally there are well known hypotheses about the direction and distance of attachments that can be used to define constraints. Eisner and Smith (2005) use the fact that short attachments are more common to improve unsupervised parsing accuracy.

4.1 "Oracle" constraints

For some experiments that follow we use "oracle" constraints that are estimated from labeled data. This involves choosing feature templates (motivated by the linguistic knowledge described above) and estimating target expectations. Oracle methods used in this paper consider three simple statistics of candidate constraint functions: count $\tilde{c}(g)$, edge count $\tilde{c}_{edge}(g)$, and edge probability $\tilde{p}(edge|g)$. Let $\mathcal{D}$ be the labeled corpus.

$$\tilde{c}(g) = \sum_{\mathbf{x}\in\mathcal{D}} \sum_i \sum_j g(x_i, x_j, \mathbf{x})$$

$$\tilde{c}_{edge}(g) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_i g(x_i, x_{y_i}, \mathbf{x})$$

$$\tilde{p}(edge|g) = \frac{\tilde{c}_{edge}(g)}{\tilde{c}(g)}$$

Constraint functions are selected according to some combination of the above statistics. In some cases we additionally prune the candidate set by considering only certain templates. To compute the target expectation, we simply use $\text{bin}(\tilde{p}(edge|g))$, where bin returns the closest value in the set $\{0, 0.1, 0.25, 0.5, 0.75, 1\}$. This can be viewed as specifying that g is *very indicative of edge*, *somewhat indicative of edge*, etc.

5 Experimental Comparison with Unsupervised Learning

In this section we compare GE training with methods for unsupervised parsing. We use the WSJ10 corpus (as processed by Smith (2006)), which is comprised of English sentences of ten words or fewer (after stripping punctuation) from the WSJ portion of the Penn Treebank. As in previous work sentences contain only part-of-speech tags.

We compare GE and supervised training of an edge-factored CRF with unsupervised learning of a DMV model (Klein and Manning, 2004) using EM and contrastive estimation (CE) (Smith and Eisner, 2005). We also report the accuracy of an attach-right baseline[6]. Finally, we report the accuracy of a constraint baseline that assigns a score to each possible edge that is the sum of the target expectations for all constraints on that edge. Possible edges without constraints receive a score of 0. These scores are used as input to the maximum spanning tree algorithm, which returns the best tree. Note that this is a strong baseline because it can handle uncertain constraints, and the tree constraint imposed by the MST algorithm helps information propagate across edges.

We note that there are considerable differences between the DMV and CRF models. The DMV model is more expressive than the CRF because it can model the arity of a head as well as sibling relationships. Because these features consider multiple edges, including them in the CRF model would make exact inference intractable (McDonald and Satta, 2007). However, the CRF may consider the distance between head and child, whereas DMV does not model distance. The CRF also models non-projective trees, which when evaluating on English is likely a disadvantage.

Consequently, we experiment with two sets of features for the CRF model. The first, *restricted* set includes features that consider the head and child tags of the dependency conjoined with the direction of the attachment, (*parent-POS,child-POS,direction*). With this feature set, the CRF model is less expressive than DMV. The second *full* set includes standard features for edge-factored dependency parsers (McDonald et al., 2005), though still unlexicalized. The CRF cannot consider valency even with the *full* feature set, but this is balanced by the ability to use distance.

[6] The reported accuracies with the DMV model and the attach-right baseline are taken from (Smith, 2006).

feature	ex.	feature	ex.
MD → VB	1.00	NNS ← **VBD**	0.75
POS ← **NN**	0.75	PRP ← **VBD**	0.75
JJ ← **NNS**	0.75	**VBD** → TO	1.00
NNP ← **POS**	0.75	**VBD** → VBN	0.75
ROOT → MD	0.75	NNS ← **VBP**	0.75
ROOT → VBD	1.00	PRP ← **VBP**	0.75
ROOT → VBP	0.75	**VBP** → VBN	0.75
ROOT → VBZ	0.75	PRP ← **VBZ**	0.75
TO → VB	1.00	NN ← **VBZ**	0.75
VBN → IN	0.75	**VBZ** → VBN	0.75

Table 1: 20 constraints that give 61.3% accuracy on WSJ10. Tags are grouped according to heads, and are in the order they appear in the sentence, with the arrow pointing from head to modifier.

We generate constraints in two ways. First, we use oracle constraints of the form (*parent-POS,child-POS,direction*) such that $\tilde{c}(g) \geq 200$. We choose constraints in descending order of $\tilde{p}(edge|g)$. The first 20 constraints selected using this method are displayed in Table 1.

Although the reader can verify that the constraints in Table 1 are reasonable, we additionally experiment with human-provided constraints. We use the prototype phrase-structure constraints provided by Haghighi and Klein (2006), and with the aid of head-finding rules, extract 14 (*parent-pos,child-pos,direction*) constraints.[7] We then estimated target expectations for these constraints using our prior knowledge, without looking at the training data. We also created a second constraint set with an additional six constraints for tag pairs that were previously underrepresented.

5.1 Results

We present results varying the number of constraints in Figures 1 and 2. Figure 1 compares supervised and GE training of the CRF model, as well as the feature constraint baseline. First we note that GE training using the *full* feature set substantially outperforms the *restricted* feature set, despite the fact that the same set of constraints is used for both experiments. This result demonstrates GE's ability to learn about related but non-constrained features. GE training also outperforms the baseline[8].

We compare GE training of the CRF model with unsupervised learning of the DMV model in Figure 2[9]. Despite the fact that the *restricted* CRF is less expressive than DMV, GE training of this model outperforms EM with 30 constraints and CE with 50 constraints. GE training of the *full* CRF outperforms EM with 10 constraints and CE with 20 constraints (those displayed in Table 1). GE training of the *full* CRF with the set of 14 constraints from (Haghighi and Klein, 2006), gives accuracy of 53.8%, which is above the interpolated oracle constraints curve (43.5% accuracy with 10 constraints, 61.3% accuracy with 20 constraints). With the 6 additional constraints, we obtain accuracy of 57.7% and match CE.

Recall that CE, EM, and the DMV model incorporate prior knowledge indirectly, and that the reported results are heavily-tuned ideal cases (see Section 2). In contrast, GE provides a method to directly encode intuitive linguistic insights.

Finally, note that structural annealing (Smith and Eisner, 2006) provides 66.7% accuracy on WSJ10 when choosing the best performing annealing schedule (Smith, 2006). As noted in Section 2 other annealing schedules provide accuracy as low as 32.5%. GE training of the *full* CRF attains accuracy of 67.0% with 30 constraints.

6 Experimental Comparison with Supervised Training on Long Sentences

Unsupervised parsing methods are typically evaluated on short sentences, as in Section 5. In this section we show that GE can be used to train parsers for longer sentences that provide comparable accuracy to supervised training with tens to hundreds of parsed sentences.

We use the standard train/test splits of the Spanish, Dutch, and Turkish data from the 2006 CoNLL Shared Task. We also use standard edge-factored feature templates (McDonald et al., 2005)[10]. We experiment with versions of the dat-

[7] Because the CFG rules in (Haghighi and Klein, 2006) are "flattened" and in some cases do not generate appropriate dependency constraints, we only used a subset.

[8] The baseline eventually matches the accuracy of the restricted CRF but this is understandable because GE's ability to bootstrap is greatly reduced with the restricted feature set.

[9] Klein and Manning (2004) report 43.2% accuracy for DMV with EM on WSJ10. When jointly modeling constituency and dependencies, Klein and Manning (2004) report accuracy of 47.5%. Seginer (2007) and Bod (2006) propose unsupervised phrase structure parsing methods that give better unlabeled F-scores than DMV with EM, but they do not report directed dependency accuracy.

[10] Typical feature processing uses only *supported* features, or those features that occur on at least one true edge in the training data. Because we assume that the data is unlabeled, we instead use features on all possible edges. This generates tens of millions features, so we prune those features that occur fewer than 10 total times, as in (Smith and Eisner, 2007).

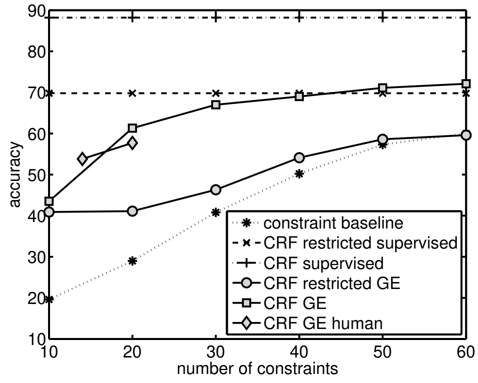

Figure 1: Comparison of the constraint baseline and both GE and supervised training of the *restricted* and *full* CRF. Note that supervised training uses 5,301 parsed sentences. GE with human provided constraints closely matches the oracle results.

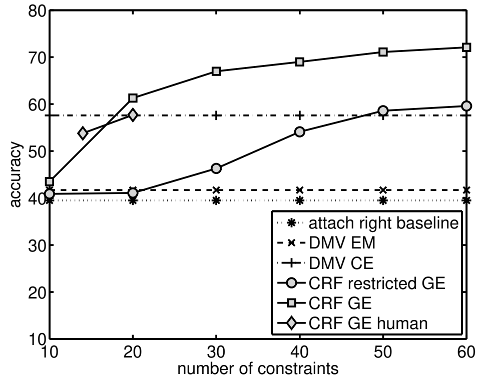

Figure 2: Comparison of GE training of the *restricted* and *full* CRFs with unsupervised learning of DMV. GE training of the *full* CRF outperforms CE with just 20 constraints. GE also matches CE with 20 human provided constraints.

sets in which we remove sentences that are longer than 20 words and 60 words.

For these experiments, we use an oracle constraint selection method motivated by the linguistic prior knowledge described in Section 4. The first set of constraints specify the most frequent head tag, attachment direction, and distance combinations for each child tag. Specifically, we select oracle constraints of the type (*parent-CPOS,child-CPOS,direction,distance*)[11]. We add constraints for every g such that $\tilde{c}_{edge}(g) > 100$ for max length 60 data sets, and $\tilde{c}_{edge}(g) > 10$ times for max length 20 data sets.

In some cases, the *possible parent* constraints described above will not be enough to provide high accuracy, because they do not consider other tags in the sentence (McDonald et al., 2005). Consequently, we experiment with adding an additional 25 *sequence* constraints (for what are often called "between" and "surrounding" features). The oracle feature selection method aims to choose such constraints that help to reduce uncertainty in the *possible parents* constraint set. Consequently, we consider sequence features g_s with $\tilde{p}(edge|g_s = 1) \geq 0.75$, and whose corresponding (*parent-CPOS,child-CPOS,direction,distance*) constraint g, has edge probability $\tilde{p}(edge|g) \leq 0.25$. Among these candidates, we sort by $\tilde{c}(g_s = 1)$, and select the top 25.

We compare with the constraint baseline described in Section 5. Additionally, we report the number of parsed sentences required for supervised CRF training (averaged over 5 random splits) to match the accuracy of GE training using the *possible parents* + *sequence* constraint set.

The results are provided in Table 2. We first observe that GE always beats the baseline, especially on parent decisions for which there are no constraints (not reported in Table 2, but for example 53.8% vs. 20.5% on Turkish 20). Second, we note that accuracy is always improved by adding *sequence* constraints. Importantly, we observe that GE gives comparable performance to supervised training with tens or hundreds of parsed sentences. These parsed sentences provide a tremendous amount of information to the model, as for example in 20 Spanish length $\leq$ 60 sentences, a total of 1,630,466 features are observed, 330,856 of them unique. In contrast, the constraint-based methods are provided at most a few hundred constraints. When comparing the human costs of parsing sentences and specifying constraints, remember that parsing sentences requires the development of detailed annotation guidelines, which can be extremely time-consuming (see also the discussion is Section 2).

Finally, we experiment with iteratively adding constraints. We sort constraints with $\tilde{c}(g) > 50$ by $\tilde{p}(edge|g)$, and ensure that 50% are (*parent-CPOS,child-CPOS,direction,distance*) constraints and 50% are *sequence* constraints. For lack of space, we only show the results for Spanish 60. In Figure 3, we see that GE beats the baseline more soundly than above, and that

[11] For these experiments we use coarse-grained part-of-speech tags in constraints.

	possible parent constraints		+ sequence constraints		complete trees
	baseline	GE	baseline	GE	
dutch 20	69.5	70.7	69.8	**71.8**	80-160
dutch 60	66.5	69.3	66.7	**69.8**	40-80
spanish 20	70.0	73.2	71.2	**75.8**	40-80
spanish 60	62.1	66.2	62.7	**66.9**	20-40
turkish 20	66.3	71.8	67.1	**72.9**	80-160
turkish 60	62.1	65.5	62.3	**66.6**	20-40

Table 2: Experiments on Dutch, Spanish, and Turkish with maximum sentence lengths of 20 and 60. Observe that GE outperforms the baseline, adding *sequence* constraints improves accuracy, and accuracy with GE training is comparable to supervised training with tens to hundreds of parsed sentences.

parent tag	true	predicted
det.	0.005	0.005
adv.	0.018	0.013
conj.	0.012	0.001
pron.	0.011	0.009
verb	0.355	0.405
adj.	0.067	0.075
punc.	0.031	0.013
noun	0.276	0.272
prep.	0.181	0.165

direction	true	predicted
right	0.621	0.598
left	0.339	0.362
distance	true	predicted
1	0.495	0.564
2	0.194	0.206
3	0.066	0.050
4	0.042	0.037
5	0.028	0.031
6-10	0.069	0.033
> 10	0.066	0.039

feature (distance)	false pos. occ.
verb → punc. (>10)	1183
noun → prep. (1)	1139
adj. → prep. (1)	855
verb → verb (6-10)	756
verb → verb (>10)	569
noun ← punc. (1)	512
verb ← punc. (2)	509
prep. ← punc. (1)	476
verb → punc. (4)	427
verb → prep. (1)	422

Table 3: Error analysis for GE training with *possible parent* + *sequence* constraints on Spanish 60 data. On the left, the predicted and true distribution over parent coarse part-of-speech tags. In the middle, the predicted and true distributions over attachment directions and distances. On the right, common features on false positive edges.

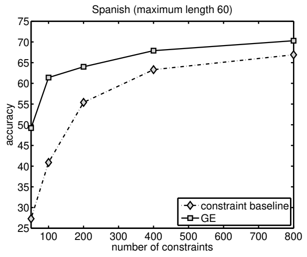

Figure 3: Comparing GE training of a CRF and constraint baseline while increasing the number of oracle constraints.

adding constraints continues to increase accuracy.

7 Error Analysis

In this section, we analyze the errors of the model learned with the *possible parent* + *sequence* constraints on the Spanish 60 data. In Table 3, we present four types of analysis. First, we present the predicted and true distributions over coarse-grained parent part of speech tags. We can see that verb is being predicted as a parent tag more often then it should be, while most other tags are predicted less often than they should be. Next, we show the predicted and true distributions over attachment direction and distance. From this we see that the model is often incorrectly predicting left attachments, and is predicting too many short attachments. Finally, we show the most common parent-child tag with direction and distance features that occur on false positive edges. From this table, we see that many errors concern the attachments of punctuation. The second line indicates a prepositional phrase attachment ambiguity.

This analysis could also be performed by a linguist by looking at predicted trees for selected sentences. Once errors are identified, GE constraints could be added to address these problems.

8 Conclusions

In this paper, we developed a novel method for the semi-supervised learning of a non-projective CRF dependency parser that directly uses linguistic prior knowledge as a training signal. It is our hope that this method will permit more effective leveraging of linguistic insight and resources and enable the construction of parsers in languages and domains where treebanks are not available.

Acknowledgments

We thank Ryan McDonald, Keith Hall, John Hale, Xiaoyun Wu, and David Smith for helpful discussions. This work was completed in part while Gregory Druck was an intern at Google. This work was supported in part by the Center for Intelligent Information Retrieval, The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

References

E. Black, J. Lafferty, and S. Roukos. 1992. Development and evaluation of a broad-coverage probabilistic grammar of english language computer manuals. In *ACL*, pages 185–192.

Rens Bod. 2006. An all-subtrees approach to unsupervised parsing. In *ACL*, pages 865–872.

E. Charniak. 2001. Immediate-head parsing for language models. In *ACL*.

R. Debusmann, D. Duchier, A. Koller, M. Kuhlmann, G. Smolka, and S. Thater. 2004. A relational syntax-semantics interface based on dependency grammar. In *COLING*.

G. Druck, G. S. Mann, and A. McCallum. 2008. Learning from labeled features using generalized expectation criteria. In *SIGIR*.

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

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency grammar induction via bitext projection constraints. In *ACL*.

A. Haghighi and D. Klein. 2006. Prototype-driven grammar induction. In *COLING*.

R. J. Kate and R. J. Mooney. 2007. Semi-supervised learning for semantic parsing using support vector machines. In *HLT-NAACL (Short Papers)*.

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

T. Koo, X. Carreras, and M. Collins. 2008. Simple semi-supervised dependency parsing. In *ACL*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

G. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*.

G. Mann and A. McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *HLT-NAACL*.

Ryan McDonald and Giorgio Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proc. of IWPT*, pages 121–132.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL*, pages 91–98.

R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman.

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *ACL*, pages 384–391, Prague, Czech Republic.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: training log-linear models on unlabeled data. In *ACL*, pages 354–362.

Noah A. Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *COLING-ACL*, pages 569–576.

David A. Smith and Jason Eisner. 2007. Bootstrapping feature-rich dependency parsers with entropic priors. In *EMNLP-CoNLL*, pages 667–677.

David A. Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *EMNLP*.

David A. Smith and Noah A. Smith. 2007. Probabilistic models of nonprojective dependency trees. In *EMNLP-CoNLL*, pages 132–140.

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

Qin Iris Wang, Dale Schuurmans, and Dekang Lin. 2008. Semi-supervised convex training for dependency parsing. In *ACL*, pages 532–540.