7-2010

Sparsity in Dependency Grammar Induction

Jennifer Gillenwater
University of Pennsylvania

Kuzman Ganchev
University of Pennsylvania

Joao V. Graca
L2F INESC-ID

Fernando Pereira
Google Inc.

Ben Taskar
University of Pennsylvania, taskar@cis.upenn.edu

Sparsity in Dependency Grammar Induction

Abstract
A strong inductive bias is essential in unsupervised grammar induction. We explore a particular sparsity bias in dependency grammars that encourages a small number of unique dependency types. Specifically, we investigate sparsity-inducing penalties on the posterior distributions of parent-child POS tag pairs in the posterior regularization (PR) framework of Graça et al. (2007). In experiments with 12 languages, we achieve substantial gains over the standard expectation maximization (EM) baseline, with average improvement in attachment accuracy of 6.3%. Further, our method outperforms models based on a standard Bayesian sparsity-inducing prior by an average of 4.9%. On English in particular, we show that our approach improves on several other state-of-the-art techniques.

Disciplines
Computer Sciences

Comments
Sparsity in Dependency Grammar Induction, J. Gillenwater, K. Ganchev, J. Graca, F. Pereira, and B. Taskar. Association for Computational Linguistics (ACL), Uppsala, Sweden, July 2010.

Sparsity in Dependency Grammar Induction

Jennifer Gillenwater and **Kuzman Ganchev**
University of Pennsylvania
Philadelphia, PA, USA
{jengi,kuzman}@cis.upenn.edu

João Graça
L^{2}F INESC-ID
Lisboa, Portugal
joao.graca@l2f.inesc-id.pt

Fernando Pereira
Google Inc.
Mountain View, CA, USA
pereira@google.com

Ben Taskar
University of Pennsylvania
Philadelphia, PA, USA
taskar@cis.upenn.edu

Abstract

A strong inductive bias is essential in unsupervised grammar induction. We explore a particular sparsity bias in dependency grammars that encourages a small number of unique dependency types. Specifically, we investigate sparsity-inducing penalties on the posterior distributions of parent-child POS tag pairs in the posterior regularization (PR) framework of Graça et al. (2007). In experiments with 12 languages, we achieve substantial gains over the standard expectation maximization (EM) baseline, with average improvement in attachment accuracy of 6.3%. Further, our method outperforms models based on a standard Bayesian sparsity-inducing prior by an average of 4.9%. On English in particular, we show that our approach improves on several other state-of-the-art techniques.

1 Introduction

We investigate an unsupervised learning method for dependency parsing models that imposes sparsity biases on the dependency types. We assume a corpus annotated with POS tags, where the task is to induce a dependency model from the tags for corpus sentences. In this setting, the *type* of a dependency is defined as a pair: tag of the dependent (also known as the child), and tag of the head (also known as the parent). Given that POS tags are designed to convey information about grammatical relations, it is reasonable to assume that only some of the possible dependency types will be realized for a given language. For instance, in English it is ungrammatical for nouns to dominate verbs, adjectives to dominate adverbs, and determiners to dominate almost any part of speech. Thus, the realized dependency types should be a sparse subset of all possible types.

Previous work in unsupervised grammar induction has tried to achieve sparsity through priors. Liang et al. (2007), Finkel et al. (2007) and Johnson et al. (2007) proposed hierarchical Dirichlet process priors. Cohen et al. (2008) experimented with a discounting Dirichlet prior, which encourages a standard dependency parsing model (see Section 2) to limit the number of dependent types for each head type.

Our experiments show a more effective sparsity pattern is one that limits the total number of unique head-dependent tag pairs. This kind of sparsity bias avoids inducing competition between dependent types for each head type. We can achieve the desired bias with a constraint on model posteriors during learning, using the posterior regularization (PR) framework (Graça et al., 2007). Specifically, to implement PR we augment the maximum marginal likelihood objective of the dependency model with a term that penalizes head-dependent tag distributions that are too permissive.

Although not focused on sparsity, several other studies use soft parameter sharing to couple different types of dependencies. To this end, Cohen et al. (2008) and Cohen and Smith (2009) investigated logistic normal priors, and Headden III et al. (2009) used a backoff scheme. We compare to their results in Section 5.

The remainder of this paper is organized as fol-

lows. Section 2 and 3 review the models and several previous approaches for learning them. Section 4 describes learning with PR. Section 5 describes experiments across 12 languages and Section 6 analyzes the results. For additional details on this work see Gillenwater et al. (2010).

2 Parsing Model

The models we use are based on the generative dependency model with valence (DMV) (Klein and Manning, 2004). For a sentence with tags $\mathbf{x}$, the root POS $r(\mathbf{x})$ is generated first. Then the model decides whether to generate a right dependent conditioned on the POS of the root and whether other right dependents have already been generated for this head. Upon deciding to generate a right dependent, the POS of the dependent is selected by conditioning on the head POS and the directionality. After stopping on the right, the root generates left dependents using the mirror reversal of this process. Once the root has generated all its dependents, the dependents generate their own dependents in the same manner.

2.1 Model Extensions

For better comparison with previous work we implemented three model extensions, borrowed from Headden III et al. (2009). The first extension alters the stopping probability by conditioning it not only on whether there are *any* dependents in a particular direction already, but also on *how many* such dependents there are. When we talk about models with maximum stop valency $V_s = S$, this means it distinguishes S different cases: $0, 1, \ldots, S-2$, and $\geq S-1$ dependents in a given direction. The basic DMV has $V_s = 2$.

The second model extension we implement is analogous to the first, but applies to dependent tag probabilities instead of stop probabilities. Again, we expand the conditioning such that the model considers how many other dependents were already generated in the same direction. When we talk about a model with maximum child valency $V_c = C$, this means we distinguish C different cases. The basic DMV has $V_c = 1$. Since this extension to the dependent probabilities dramatically increases model complexity, the third model extension we implement is to add a backoff for the dependent probabilities that does not condition on the identity of the parent POS (see Equation 2).

More formally, under the extended DMV the probability of a sentence with POS tags $\mathbf{x}$ and dependency tree $\mathbf{y}$ is given by:

$$p_\theta(\mathbf{x}, \mathbf{y}) = p_{root}(r(\mathbf{x})) \times \prod_{y \in \mathbf{y}} p_{stop}(false \mid y_p, y_d, y_{v_s}) p_{child}(y_c \mid y_p, y_d, y_{v_c}) \times \prod_{x \in \mathbf{x}} p_{stop}(true \mid x, left, x_{v_l})\, p_{stop}(true \mid x, right, x_{v_r}) \quad (1)$$

where y is the dependency of y_c on head y_p in direction y_d, and y_{v_c}, y_{v_s}, x_{v_r}, and x_{v_l} indicate valence. For the third model extension, the backoff to a probability not dependent on parent POS can be formally expressed as:

$$\lambda p_{child}(y_c \mid y_p, y_d, y_{v_c}) + (1-\lambda) p_{child}(y_c \mid y_d, y_{v_c}) \quad (2)$$

for $\lambda \in [0, 1]$. We fix $\lambda = 1/3$, which is a crude approximation to the value learned by Headden III et al. (2009).

3 Previous Learning Approaches

In our experiments, we compare PR learning to standard expectation maximization (EM) and to Bayesian learning with a sparsity-inducing prior. The EM algorithm optimizes marginal likelihood $\mathcal{L}(\theta) = \log \sum_{\mathbf{Y}} p_\theta(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^n\}$ denotes the entire unlabeled corpus and $\mathbf{Y} = \{\mathbf{y}^1, \ldots, \mathbf{y}^n\}$ denotes a set of corresponding parses for each sentence. Neal and Hinton (1998) view EM as block coordinate ascent on a function that lower-bounds $\mathcal{L}(\theta)$. Starting from an initial parameter estimate θ^0, the algorithm iterates two steps:

$$\mathbf{E}: q^{t+1} = \arg\min_q \mathbf{KL}(q(\mathbf{Y}) \parallel p_{\theta^t}(\mathbf{Y} \mid \mathbf{X})) \quad (3)$$

$$\mathbf{M}: \theta^{t+1} = \arg\max_\theta \mathbf{E}_{q^{t+1}} [\log p_\theta(\mathbf{X}, \mathbf{Y})] \quad (4)$$

Note that the E-step just sets $q^{t+1}(\mathbf{Y}) = p_{\theta^t}(\mathbf{Y}|\mathbf{X})$, since it is an unconstrained minimization of a KL-divergence. The PR method we present modifies the E-step by adding constraints.

Besides EM, we also compare to learning with several Bayesian priors that have been applied to the DMV. One such prior is the Dirichlet, whose hyperparameter we will denote by α. For $\alpha < 0.5$, this prior encourages parameter sparsity. Cohen et al. (2008) use this method with $\alpha = 0.25$ for training the DMV and achieve improvements over basic EM. In this paper we will refer to our own implementation of the Dirichlet prior as the "discounting Dirichlet" (DD) method. In addition to

the Dirichlet, other types of priors have been applied, in particular logistic normal priors (LN) and shared logistic normal priors (SLN) (Cohen et al., 2008; Cohen and Smith, 2009). LN and SLN aim to tie parameters together. Essentially, this has a similar goal to sparsity-inducing methods in that it posits a more concise explanation for the grammar of a language. Headden III et al. (2009) also implement a sort of parameter tying for the E-DMV through a learning a backoff distribution on child probabilities. We compare against results from all these methods.

4 Learning with Sparse Posteriors

We would like to penalize models that predict a large number of distinct dependency types. To enforce this penalty, we use the posterior regularization (PR) framework (Graça et al., 2007). PR is closely related to generalized expectation constraints (Mann and McCallum, 2007; Mann and McCallum, 2008; Bellare et al., 2009), and is also indirectly related to a Bayesian view of learning with constraints on posteriors (Liang et al., 2009). The PR framework uses constraints on posterior expectations to guide parameter estimation. Here, PR allows a natural and tractable representation of sparsity constraints based on edge type counts that cannot easily be encoded in model parameters. We use a version of PR where the desired bias is a penalty on the log likelihood (see Ganchev et al. (2010) for more details). For a distribution p_θ, we define a penalty as the (generic) β-norm of expectations of some features ϕ:

$$||\mathbf{E}_{p_\theta}[\phi(\mathbf{X}, \mathbf{Y})]||_\beta \tag{5}$$

For computational tractability, rather than penalizing the model's posteriors directly, we use an auxiliary distribution q, and penalize the marginal log-likelihood of a model by the KL-divergence of p_θ from q, plus the penalty term with respect to q. For a fixed set of model parameters θ the full PR penalty term is:

$$\min_q \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) \;+\; \sigma\, ||\mathbf{E}_q[\phi(\mathbf{X}, \mathbf{Y})]||_\beta \tag{6}$$

where σ is the strength of the regularization. PR seeks to maximize $\mathcal{L}(\theta)$ minus this penalty term. The resulting objective can be optimized by a variant of the EM (Dempster et al., 1977) algorithm used to optimize $\mathcal{L}(\theta)$.

4.1 ℓ_1/ℓ_∞ Regularization

We now define precisely how to count dependency types. For each child tag c, let i range over an enumeration of all occurrences of c in the corpus, and let p be another tag. Let the indicator $\phi_{cpi}(\mathbf{X}, \mathbf{Y})$ have value 1 if p is the parent tag of the ith occurrence of c, and value 0 otherwise. The number of unique dependency types is then:

$$\sum_{cp} \max_i \phi_{cpi}(\mathbf{X}, \mathbf{Y}) \tag{7}$$

Note there is an asymmetry in this count: occurrences of child type c are enumerated with i, but all occurrences of parent type p are or-ed in ϕ_{cpi}. That is, $\phi_{cpi} = 1$ if *any* occurrence of p is the parent of the ith occurrence of c. We will refer to PR training with this constraint as PR-AS. Instead of counting pairs of a child token and a parent type, we can alternatively count pairs of a child token and a parent token by letting p range over all *tokens* rather than *types*. Then each potential dependency corresponds to a different indicator ϕ_{cpij}, and the penalty is symmetric with respect to parents and children. We will refer to PR training with this constraint as PR-S. Both approaches perform very well, so we report results for both.

Equation 7 can be viewed as a mixed-norm penalty on the features ϕ_{cpi} or ϕ_{cpij}: the sum corresponds to an ℓ_1 norm and the max to an ℓ_∞ norm. Thus, the quantity we want to minimize fits precisely into the PR penalty framework. Formally, to optimize the PR objective, we complete the following E-step:

$$\arg\min_q \mathbf{KL}(q(\mathbf{Y})||p_\theta(\mathbf{Y}|\mathbf{X})) + \sigma \sum_{cp} \max_i \mathbf{E}_q[\phi(\mathbf{X}, \mathbf{Y})], \tag{8}$$

which can equivalently be written as:

$$\begin{aligned} \min_{q(\mathbf{Y}), \xi_{cp}} \quad & \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) + \sigma \sum_{cp} \xi_{cp} \\ \text{s.t.} \quad & \xi_{cp} \leq \mathbf{E}_q[\phi(\mathbf{X}, \mathbf{Y})] \end{aligned} \tag{9}$$

where ξ_{cp} corresponds to the maximum expectation of ϕ over all instances of c and p. Note that the projection problem can be solved efficiently in the dual (Ganchev et al., 2010).

5 Experiments

We evaluate on 12 languages. Following the example of Smith and Eisner (2006), we strip punctuation from the sentences and keep only sentences of length ≤ 10. For simplicity, for all models we use the "harmonic" initializer from Klein

Model	EM	PR	Type	σ
DMV	45.8	**62.1**	PR-S	140
2-1	45.1	**62.7**	PR-S	100
2-2	54.4	**62.9**	PR-S	80
3-3	55.3	**64.3**	PR-S	140
4-4	55.1	**64.4**	PR-AS	140

Table 1: Attachment accuracy results. **Column 1**: V_c-V_s used for the E-DMV models. **Column 3**: Best PR result for each model, which is chosen by applying each of the two types of constraints (PR-S and PR-AS) and trying $\sigma \in \{80, 100, 120, 140, 160, 180\}$. **Columns 4 & 5**: Constraint type and σ that produced the values in column 3.

Learning Method	Accuracy		
	≤ 10	≤ 20	all
PR-S ($\sigma = 140$)	**62.1**	**53.8**	**49.1**
LN families	59.3	45.1	39.0
SLN TieV & N	61.3	47.4	41.4
PR-AS ($\sigma = 140$)	64.4	55.2	50.5
DD ($\alpha = 1$, λ learned)	**65.0** ($\pm$5.7)		

Table 2: Comparison with previous published results. Rows 2 and 3 are taken from Cohen et al. (2008) and Cohen and Smith (2009), and row 5 from Headden III et al. (2009).

and Manning (2004), which we refer to as K&M. We always train for 100 iterations and evaluate on the test set using Viterbi parses. Before evaluating, we smooth the resulting models by adding e^{-10} to each learned parameter, merely to remove the chance of zero probabilities for unseen events. (We did not tune this as it should make very little difference for final parses.) We score models by their attachment accuracy — the fraction of words assigned the correct parent.

5.1 Results on English

We start by comparing English performance for EM, PR, and DD. To find α for DD we searched over five values: $\{0.01, 0.1, 0.25, 1\}$. We found 0.25 to be the best setting for the DMV, the same as found by Cohen et al. (2008). DD achieves accuracy 46.4% with this α. For the E-DMV we tested four model complexities with valencies V_c-V_s of 2-1, 2-2, 3-3, and 4-4. DD's best accuracy was 53.6% with the 4-4 model at $\alpha = 0.1$. A comparison between EM and PR is shown in Table 1. PR-S generally performs better than the PR-AS for English. Comparing PR-S to EM, we also found PR-S is always better, independent of the particular σ, with improvements ranging from 2% to 17%. Note that in this work we do not perform the PR projection at test time; we found it detrimental, probably due to a need to set the (corpus-size-dependent) σ differently for the test set. We also note that development likelihood and the best setting for σ are not well-correlated, which unfortunately makes it hard to pick these parameters without some supervision.

5.2 Comparison with Previous Work

In this section we compare to previously published unsupervised dependency parsing results for English. It might be argued that the comparison is unfair since we do supervised selection of model complexity and regularization strength. However, we feel the comparison is not so unfair as we perform only a very limited search of the model-σ space. Specifically, the only values of σ we search over are $\{80, 100, 120, 140, 160, 180\}$.

First, we consider the top three entries in Table 2, which are for the basic DMV. The first entry was generated using our implementation of PR-S. The second two entries are logistic normal and shared logistic normal parameter tying results (Cohen et al., 2008; Cohen and Smith, 2009). The PR-S result is the clear winner, especially as length of test sentences increases. For the bottom two entries in the table, which are for the E-DMV, the last entry is best, corresponding to using a DD prior with $\alpha = 1$ (non-sparsifying), but with a special "random pools" initialization and a learned weight λ for the child backoff probability. The result for PR-AS is well within the variance range of this last entry, and thus we conjecture that combining PR-AS with random pools initialization and learned λ would likely produce the best-performing model of all.

5.3 Results on Other Languages

Here we describe experiments on 11 additional languages. For each we set σ and model complexity (DMV versus one of the four E-DMV experimented with previously) based on the best configuration found for English. This likely will not result in the ideal parameters for all languages, but provides a realistic test setting: a user has available a labeled corpus in one language, and would like to induce grammars for many other languages. Table 3 shows the performance for all models and training procedures. We see that the sparsifying methods tend to improve over EM most of the time. For the basic DMV, average improvements are 1.6% for DD, 6.0% for PR-S, and 7.5% for PR-AS. PR-AS beats PR-S in 8 out of 12 cases,

	Bg	Cz	De	Dk	En	Es	Jp	Nl	Pt	Se	Si	Tr
	DMV Model											
EM	37.8	29.6	35.7	**47.2**	45.8	40.3	52.8	37.1	35.7	39.4	42.3	46.8
DD 0.25	39.3	30.0	38.6	43.1	46.4	47.5	57.8	35.1	38.7	40.2	48.8	43.8
PR-S 140	53.7	31.5	**39.6**	44.0	**62.1**	61.1	58.8	31.0	47.0	**42.2**	39.9	51.4
PR-AS 140	**54.0**	**32.0**	**39.6**	42.4	61.9	**62.4**	**60.2**	**37.9**	**47.8**	38.7	**50.3**	**53.4**
	Extended Model											
EM (3,3)	41.7	48.9	40.1	46.4	55.3	44.3	48.5	**47.5**	35.9	**48.6**	47.5	46.2
DD 0.1 (4,4)	47.6	48.5	42.0	44.4	53.6	48.9	57.6	45.2	48.3	47.6	35.6	48.9
PR-S 140 (3,3)	59.0	**54.7**	**47.4**	45.8	64.3	**57.9**	**60.8**	33.9	**54.3**	45.6	49.1	56.3
PR-AS 140 (4,4)	**59.8**	54.6	45.7	**46.6**	**64.4**	**57.9**	59.4	38.8	49.5	41.4	**51.2**	**56.9**

Table 3: Attachment accuracy results. The parameters used are the best settings found for English. Values for hyperparameters (α or σ) are given after the method name. For the extended model (V_c, V_s) are indicated in parentheses. En is the English Penn Treebank (Marcus et al., 1993) and the other 11 languages are from the CoNLL X shared task: Bulgarian [Bg] (Simov et al., 2002), Czech [Cz] (Bohomovà et al., 2001), German [De] (Brants et al., 2002), Danish [Dk] (Kromann et al., 2003), Spanish [Es] (Civit and Martí, 2004), Japanese [Jp] (Kawata and Bartels, 2000), Dutch [Nl] (Van der Beek et al., 2002), Portuguese [Pt] (Afonso et al., 2002), Swedish [Se] (Nilsson et al., 2005), Slovene [Sl] (Džeroski et al., 2006), and Turkish [Tr] (Oflazer et al., 2003).

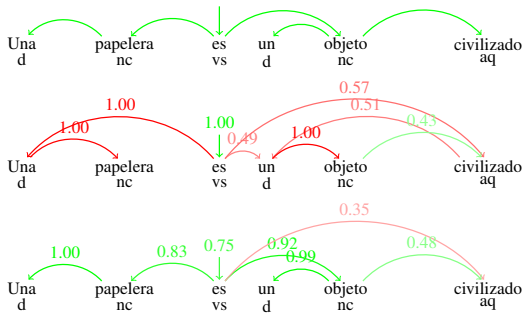

Figure 1: Posterior edge probabilities for an example sentence from the Spanish test corpus. At the top are the gold dependencies, the middle are EM posteriors, and bottom are PR posteriors. Green indicates correct dependencies and red indicates incorrect dependencies. The numbers on the edges are the values of the posterior probabilities.

though the average increase is only 1.5%. PR-S is also better than DD for 10 out of 12 languages. If we instead consider these methods for the E-DMV, DD performs worse, just 1.4% better than the E-DMV EM, while both PR-S and PR-AS continue to show substantial average improvements over EM, 6.5% and 6.3%, respectively.

6 Analysis

One common EM error that PR fixes in many languages is the directionality of the noun-determiner relation. Figure 1 shows an example of a Spanish sentence where PR significantly outperforms EM because of this. Sentences such as "Lleva tiempo entenderlos" which has tags "main-verb common-noun main-verb" (no determiner tag) provide an explanation for PR's improvement—when PR sees that sometimes nouns can appear without determiners but that the opposite situation does not occur, it shifts the model parameters to make nouns the parent of determiners instead of the reverse. Then it does not have to pay the cost of assigning a parent with a new tag to cover each noun that doesn't come with a determiner.

7 Conclusion

In this paper we presented a new method for unsupervised learning of dependency parsers. In contrast to previous approaches that constrain model parameters, we constrain model posteriors. Our approach consistently outperforms the standard EM algorithm and a discounting Dirichlet prior.

We have several ideas for further improving our constraints, such as: taking into account the directionality of the edges, using different regularization strengths for the root probabilities than for the child probabilities, and working directly on word types rather than on POS tags. In the future, we would also like to try applying similar constraints to the more complex task of joint induction of POS tags and dependency parses.

Acknowledgments

J. Gillenwater was supported by NSF-IGERT 0504487. K. Ganchev was supported by ARO MURI SUBTLE W911NF-07-1-0216. J. Graça was supported by FCT fellowship SFRH/BD/27528/2006 and by FCT project CMU-PT/HuMach/0039/2008. B. Taskar was partly supported by DARPA CSSG and ONR Young Investigator Award N000141010746.

References

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

K. Bellare, G. Druck, and A. McCallum. 2009. Alternating projections for learning with expectation constraints. In *Proc. UAI*.

A. Bohomovà, J. Hajic, E. Hajicova, and B. Hladka. 2001. The prague dependency treebank: Three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*.

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

M. Civit and M.A. Martí. 2004. Building cast3lb: A Spanish Treebank. *Research on Language & Computation*.

S.B. Cohen and N.A. Smith. 2009. The shared logistic normal distribution for grammar induction. In *Proc. NAACL*.

S.B. Cohen, K. Gimpel, and N.A. Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *Proc. NIPS*.

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

S. Džeroski, T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky, and A. Žele. 2006. Towards a Slovene dependency treebank. In *Proc. LREC*.

J. Finkel, T. Grenager, and C. Manning. 2007. The infinite tree. In *Proc. ACL*.

K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*.

J. Gillenwater, K. Ganchev, J. Graça, F. Pereira, and B. Taskar. 2010. Posterior sparsity in unsupervised dependency parsing. Technical report, MS-CIS-10-19, University of Pennsylvania.

J. Graça, K. Ganchev, and B. Taskar. 2007. Expectation maximization and posterior constraints. In *Proc. NIPS*.

W.P. Headden III, M. Johnson, and D. McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proc. NAACL*.

M. Johnson, T.L. Griffiths, and S. Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Proc. NIPS*.

Y. Kawata and J. Bartels. 2000. Stylebook for the Japanese Treebank in VERBMOBIL. Technical report, Eberhard-Karls-Universitat Tubingen.

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

M.T. Kromann, L. Mikkelsen, and S.K. Lynge. 2003. Danish Dependency Treebank. In *Proc. TLT*.

P. Liang, S. Petrov, M.I. Jordan, and D. Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proc. EMNLP*.

P. Liang, M.I. Jordan, and D. Klein. 2009. Learning from measurements in exponential families. In *Proc. ICML*.

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

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

M. Marcus, M. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

R. Neal and G. Hinton. 1998. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press.

J. Nilsson, J. Hall, and J. Nivre. 2005. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. *NODALIDA Special Session on Treebanks*.

K. Oflazer, B. Say, D.Z. Hakkani-Tür, and G. Tür. 2003. Building a Turkish treebank. *Treebanks: Building and Using Parsed Corpora*.

K. Simov, P. Osenova, M. Slavcheva, S. Kolkovska, E. Balabanova, D. Doikoff, K. Ivanova, A. Simov, E. Simov, and M. Kouylekov. 2002. Building a linguistically interpreted corpus of bulgarian: the bultreebank. In *Proc. LREC*.

N. Smith and J. Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *Proc. ACL*.

L. Van der Beek, G. Bouma, R. Malouf, and G. Van Noord. 2002. The Alpino dependency treebank. *Language and Computers*.