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

We investigate an unsupervised learning method for dependency parsing 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. The models we use are based on the generative dependency model with valence (DMV) (Klein and Manning, 2004). 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 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 POS. Thus, 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. For example, Cohen et al. (2008) experimented with a Dirichlet prior that encourages a standard dependency parsing model 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 likelihood objective of the dependency model with a term that penalizes head-dependent tag distributions that are too permissive.

To briefly review, the standard optimization technique for the DMV is the expectation maximization (EM) algorithm. EM 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:

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

$$\mathbf{M}: \theta^{t+1} = \arg\max_\theta \mathbf{E}_{q^{t+1}}[\log p_\theta(\mathbf{X}, \mathbf{Y})] \quad (2)$$

Note that the E-step just sets $q^{t+1}(\mathbf{Y}) = p_{\theta^t}(\mathbf{Y} \mid \mathbf{X})$, since it is an unconstrained minimization of a KL-divergence. The PR method we use is almost identical to EM, except that it modifies the E-step by adding a penalty term for grammars that are too permissive.

To be specific, the exact form the penalty takes is basically a count of the number of distinct dependency types the grammar allows. It is easiest to express a count of the number of types by using edge posteriors—the joint probability $p(c, p)$ that a child c has parent p—and this is why the penalty can be most naturally enforced using the PR framework. 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 given a set of gold parse trees is then:

$$\sum_{cp} \max_{i} \phi_{cpi}(\mathbf{X}, \mathbf{Y}) \tag{3}$$

which can also be written using mixed norm notation: $||\phi_{cpi}(\mathbf{X}, \mathbf{Y})||_{\ell_1/\ell_\infty}$. Note there is an asymmetry in this: $\phi_{cpi} = 1$ if *any* occurrence of p is parent of the ith occurrence of c. We call PR training with this constraint 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*. We call PR training with this constraint PR-S.

Since we are exploring unsupervised learning, instead of gold trees with $\phi_{cpi}(\mathbf{X}, \mathbf{Y})$ always 0 or 1, we actually have a distribution over parse trees and expectations of edges $\mathbf{E}[\phi(\mathbf{X}, \mathbf{Y})]$. Equation 3 can thus be re-written:

$$\sum_{cp} \max_{i} \mathbf{E}[\phi(\mathbf{X}, \mathbf{Y})]. \tag{4}$$

For computational tractability, the way PR works is that rather than penalizing the model's posteriors directly, it uses an auxiliary distribution q, and penalizes 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 E-step is:

$$\min_{q} \mathbf{KL}(q(\mathbf{Y}) \parallel p_\theta(\mathbf{Y}|\mathbf{X})) \;+\; \sigma\, ||\mathbf{E}_q[\phi(\mathbf{X}, \mathbf{Y})]||_{\ell_1/\ell_\infty} \tag{5}$$

where σ is the strength of the regularization.

In our experiments, we have compared PR primarily to two other methods: EM and learning with a discounting (sparsifying) Dirichlet prior (DD). For the basic DMV, average improvements over EM across 11 different languages are 1.6% for DD, 6.0% for PR-S, and 7.5% for PR-AS. For better comparison with previous work we also implemented two model extensions, borrowed from Headden III et al. (2009). On the extended DMV, DD performs worse, just 1.4% better than EM, while both PR-S and PR-AS continue to show substantial average improvements over EM, 6.5% and 6.3%, respectively.

To give some intuition as to why PR works, we highlight one common EM error that PR fixes in many languages—the directionality of the noun-determiner relation. Figure 1 shows an example of a Spanish sentence where PR significantly outperforms EM because of this. The reason PR succeeds here is that in the corpora sometimes nouns can appear without determiners but the opposite situation does not occur. Thus, in order to avoid paying the cost of assigning a new parent tag to cover each noun that doesn't have a determiner, PR instead reverses the noun-determiner relation.

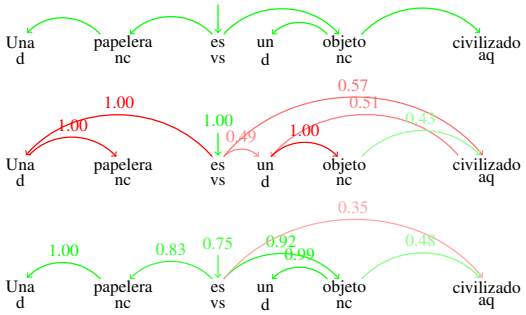

Figure 1: Posterior edge probabilities for an example sentence from the Spanish test corpus. At the top are the gold dependencies, the middle are EM posteriors, and bottom are PR posteriors. Green indicates correct dependencies and red indicates incorrect dependencies. The numbers on the edges are the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

In summary, we present 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 sparsifying Dirichlet prior.

References

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

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*.

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

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.