Combining the Sparsity and Unambiguity Biases for Grammar Induction

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

Abstract

In this paper we describe our participating system for the dependency induction track of the PASCAL Challenge on Grammar Induction. Our system incorporates two types of inductive biases: the sparsity bias and the unambiguity bias. The sparsity bias favors a grammar with fewer grammar rules. The unambiguity bias favors a grammar that leads to unambiguous parses, which is motivated by the observation that natural language is remarkably unambiguous in the sense that the number of plausible parses of a natural language sentence is very small. We introduce our approach to combining these two types of biases and discuss the system implementation. Our experiments show that both types of inductive biases are beneficial to grammar induction.

1 Introduction

Grammar induction refers to the induction of a formal grammar from a corpus of unannotated sentences. There has been significant progress over the past decade in the research of natural language grammar induction. A variety of approaches and techniques have been proposed, most of which are designed to induce probabilistic dependency grammars. The PASCAL Challenge on Grammar Induction aims to provide a thorough evaluation of approaches to natural language grammar induction. The challenge includes three tracks: inducing dependency structures using the gold standard part-of-speech tags, inducing both dependency structures and part-of-speech tags directly from text, and an open-resource track which allows other external resources to be used. Ten corpora of nine different languages are used in the challenge: Arabic (Hajič et al., 2004), Basque (Aduriz et al., 2003), Czech (Hajič et al., 2000), Danish (Buch-Kromann et al., 2007), Dutch (Beek et al., 2002), English WSJ (Marcus et al., 1993), English CHILDES (Sagae et al., 2007), Portuguese (Afonso et al., 2002), Slovene (Erjavec et al., 2010), and Swedish (Nivre et al., 2006). For each corpus, a large set of unannotated sentences are provided as the training data, along with a small set of annotated sentences as the development data; the predictions on the unannotated test data submitted by challenge participants are evaluated against the gold standard annotations.

We participate in the track of inducing dependency structures from gold standard part-of-speech tags. Our system incorporates two types of inductive biases in learning dependency grammars: the sparsity bias and the unambiguity bias. The sparsity bias favors a grammar with fewer grammar rules. We employ two different approaches to inducing sparsity: Dirichlet priors over grammar rule probabilities and an approach based on posterior regularization (Gillenwater et al., 2010). The unambiguity bias favors a grammar that leads to unambiguous parses, which is motivated by the observation that natural language is remarkably unambiguous in the sense that the number of plausible parses of a natural language sentence is very small. To induce unambiguity in the learned grammar we propose an approach named unambiguity regularization based on the posterior regularization framework (Ganchev et al., 2010). To combine Dirichlet priors with unam-

NAACL-HLT Workshop on the Induction of Linguistic Structure, pages 105–110,
Montréal, Canada, June 3-8, 2012.

biguity regularization, we derive a mean-field variational inference algorithm. To combine the sparsity-inducing posterior regularization approach with unambiguity regularization, we employ a simplistic approach that optimizes the two regularization terms separately.

The rest of the paper is organized as follows. Section 2 introduces the two approaches that we employ to induce sparsity. Section 3 introduces the unambiguity bias and the unambiguity regularization approach. Section 4 discusses how we combine the sparsity bias with the unambiguity bias. Section 5 provides details of our implementation and training procedure. Section 6 concludes the paper.

2 Sparsity Bias

A sparsity bias in grammar induction favors a grammar that has fewer grammar rules. We employ two different approaches to inducing sparsity: Dirichlet priors over grammar rule probabilities and an approach based on posterior regularization (Gillenwater et al., 2010).

A probabilistic grammar consists of a set of probabilistic grammar rules. A discrete distribution is defined over each set of grammar rules with the same left-hand side, and a Dirichlet distribution can be used as the prior of the discrete distribution. Denote vector θ of dimension K as the parameter of a discrete distribution. Then a Dirichlet prior over θ is defined as:

$$P(\theta; \alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

where $\alpha = (\alpha_1, \ldots, \alpha_K)$ are the hyperparameters, and $B(\alpha)$ is the normalization constant. Typically, all the hyperparameters are set to the same value. It can be shown that if the hyperparameters are less than 1, then the Dirichlet prior assigns larger probabilities to vectors that have more elements close to zero. Therefore, Dirichlet priors can be used to encourage parameter sparsity. It has been found that when applied to dependency grammar induction, Dirichlet priors with hyperparamters set to values less than 1 can slightly improve the accuracy of the learned grammar over the maximum-likelihood estimation (Cohen et al., 2008; Gillenwater et al., 2010).

Gillenwater et al. (2010) proposed a differnt approach to inducing sparsity in dependency grammar induction based on the posterior regularization framework (Ganchev et al., 2010). They added a regularization term to the posterior of the grammar that penalizes the number of unique dependency types in the parses of the training data. More specifically, their objective function is:

$$J(\theta) = \log p(\theta|\mathbf{X}) - \min_q \Bigg(\mathbf{KL}(q(\mathbf{Z}) || p_\theta(\mathbf{Z}|\mathbf{X})) + \sigma_s \sum_{cp} \max_i \mathbf{E}_q[\phi_{cpi}(\mathbf{X}, \mathbf{Z})] \Bigg)$$

where θ is the parameter of the grammar, $\mathbf{X}$ is the training data, $\mathbf{Z}$ is the dependency parses of the training data $\mathbf{X}$, σ_s is a constant that controls the strength of the regularization term, c and p range over all the tags of the dependency grammar, i ranges over all the occurrences of tag c in the training data $\mathbf{X}$, and $\phi_{cpi}(\mathbf{X}, \mathbf{Z})$ is an indicator function of whether tag p is the dependency head of the i-th occurrence of tag c in the dependency parses $\mathbf{Z}$. This objective function is optimized using a variant of the expectation-maximization algorithm (EM), which contains an E-step that optimizes the auxiliary distribution q using the projected subgradient method. It has been shown that this approach achieves higher degree of sparsity than Dirichlet priors and leads to significant improvement in accuracy of the learned grammars.

3 Unambiguity Bias

The unambiguity bias favors a grammar that leads to unambiguous parses on natural language sentences (Tu and Honavar, 2012). This bias is motivated by the observation that natural language is remarkably unambiguous in the sense that the number of plausible parses of a natural language sentence is very small in comparison with the total number of possible parses. To illustrate this, we randomly sample an English sentence from the Wall Street Journal and parse the sentence using the Berkeley parser (Petrov et al., 2006), one of the state-of-the-art English language parsers. The estimated total number of possible parses of this sentence is 2×10^{20} (by assuming a complete Chomsky normal form grammar with

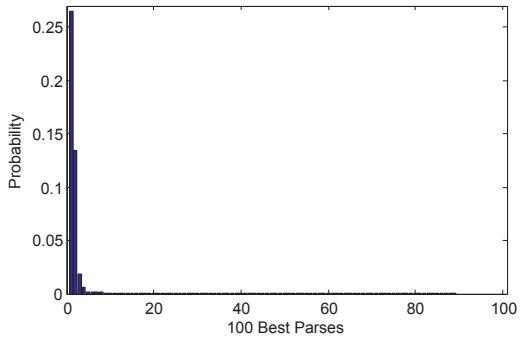

Figure 1: The probabilities of the 100 best parses of the sample sentence.

the same number of nonterminals as in the Berkeley parser). However, as shown in Figure 1, most of the parses have probabilities that are negligible compared with the probability of the best parse.

To induce unambiguity in the learned grammar, we derive an approach named *unambiguity regularization* (Tu and Honavar, 2012) based on the posterior regularization framework (Ganchev et al., 2010). Specifically, we add into the objective function a regularization term that penalizes the entropy of the parses given the training sentences. Let $\mathbf{X}$ denote the set of training sentences, $\mathbf{Z}$ denote the set of parses of the training sentences, and θ denote the rule probabilities of the grammar. Our objective function is

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

where σ_u is a nonnegative constant that controls the strength of the regularization term; q is an auxiliary distribution. The first term in the objective function is the log posterior probability of the grammar parameters given the training corpus, and the second term minimizes the KL-divergence between the auxiliary distribution q and the posterior distribution of $\mathbf{Z}$ while also minimizes the entropy of q. This objective function is optimized using coordinate ascent in our approach. It can be shown that the behavior of our approach is controlled by the value of the parameter σ_u. When $\sigma_u = 0$, our approach reduces to the standard EM algorithm. When $\sigma_u \geq 1$, our approach reduces to the Viterbi EM algorithm, which considers only the best parses of the training sentences in the E-step. When $0 < \sigma_u < 1$, our approach falls between standard EM and Viterbi EM: it applies a softmax function to the distribution of the parse z_i of each training sentence x_i in the E-step:

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

where α_i is the normalization factor. To compute q, note that $p_\theta(z_i|x_i)$ is the product of a set of grammar rule probabilities, so we can raise all the rule probabilities of the grammar to the power of $\frac{1}{1-\sigma_u}$ and then run the normal E-step of the EM algorithm. The normalization of q is included in the normal E-step. We refer to the algorithm in the case of $0 < \sigma_u < 1$ as the *softmax-EM* algorithm.

The choice of the value of σ_u is important in unambiguity regularization. Considering that in grammar induction the initial grammar is typically very ambiguous, the value of σ_u should be set large enough to induce unambiguity. On the other hand, natural language grammars do contain some degree of ambiguity, so the value of σ_u should not be set too large. One way to avoid choosing a fixed value of σ_u is to anneal its value. We start learning with a large value of σ_u (e.g., $\sigma_u = 1$) to strongly push the learner away from the highly ambiguous initial grammar; then we gradually reduce the value of σ_u, possibly ending with $\sigma_u = 0$, to avoid inducing excessive unambiguity in the learned grammar.

4 Combining Sparsity and Unambiguity Biases

To incorporate Dirichlet priors over grammar rule probabilities into our unambiguity regularization approach, we derive a mean-field variational inference algorithm (Tu and Honavar, 2012). The algorithm alternately optimizes $q(\theta)$ and $q(\mathbf{Z})$. The optimization of $q(\theta)$ is exactly the same as in the standard mean-field variational inference with Dirichlet priors, in which we obtain a set of weights that are summarized from $q(\theta)$ (Kurihara and Sato, 2004). The optimization of $q(\mathbf{Z})$ is similar to the E-step of our approach discussed in section 3: when $0 < \sigma_u < 1$, we raise all the weights to the power of $\frac{1}{1-\sigma_u}$ before running the normal step of computing $q(\mathbf{Z})$ in the standard mean-field variational inference; and when $\sigma_u \geq 1$, we use the weights to find the best parse of each training sentence and assign probability 1 to it.

The sparsity-inducing posterior regularization approach and our unambiguity regularization approach are based on the same posterior regularization framework. To combine these two approaches, the standard method is to optimize a linear combination of the sparsity and unambiguity regularization terms in the E-step of the posterior regularization algorithm. Here we employ a simplistic approach instead which optimizes the two regularization terms separately in the E-step. Specifically, we first ignore the sparsity regularization term and optimize $q(\mathbf{Z})$ with respect to the unambiguity regularization term using the approach discussed in section 3. The optimization result is an intermediate distribution $q'(\mathbf{Z})$. Then we ignore the unambiguity regularization term and optimize $q(\mathbf{Z})$ to minimize the sparsity regularization term as well as the KL-divergence between $q(\mathbf{Z})$ and $q'(\mathbf{Z})$.

5 Implementation and Experiments

Our system was built on top of the PR-Dep-Parsing package[1]. We implemented both approaches introduced in section 4, i.e., unambiguity regularization with Dirichlet priors and combined posterior regularization of sparsity and unambiguity. For the latter, we did not implement the $\sigma_u \geq 1$ case and the annealing of σ_u because of time constraint.

We preprocessed the corpora to remove all the punctuations as denoted by the universal POS tags. One exception is that for the English WSJ corpus we did not remove the \$ symbol because we found that removing it significantly decreased the accuracy of the learned grammar. We combined the provided training, development and test set as our training set. We trained our system on the fine POS tags except for the Dutch corpus. In the Dutch corpus, the fine POS tags are the same as the coarse POS tags except that each multi-word unit is annotated with the concatenation of the POS tags of all the component words, making the training data for such tags extremely sparse. So we chose to use the coarse POS tags for the Dutch corpus.

We employed the informed initialization proposed in (Klein and Manning, 2004) and ran our two approaches on the training set. We tuned the parameters by coordinate ascent on the development set. The parameters that we tuned include the maximal length of sentences used in training, the valence and back-off strength of the E-DMV model, the hyperparameter α of Dirichlet priors, the type (PR-S or PR-AS) and strength σ_s of sparsity-inducing posterior regularization, and the strength σ_u of unambiguity regularization. Sparsity-inducing posterior regularization has a high computational cost. Consequently, we were not able to run our second approach on the English CHILDES corpus and the Czech corpus, and performed relatively limited parameter tuning of the second approach on the other eight corpora.

Table 1 shows, for each corpus, the approach and the parameters that we found to perform the best on the development set and were hence used to learn the final grammar that produced the submitted predictions on the test set. Each of our two approaches was found to be the better approach for five of the ten corpora. The sparsity bias was found to be beneficial (i.e., $\alpha < 1$ if Dirichlet priors were used, or $\sigma_s > 0$ if sparsity-inducing posterior regularization was used) for six of the ten corpora. The unambiguity bias was found to be beneficial (i.e., $\sigma_u > 0$) for seven of the ten corpora. This implies the usefulness of both types of inductive biases in grammar induction. For only one corpus, the English CHILDES corpus, neither the sparsity bias nor the unambiguity bias was found to be beneficial, probably because this corpus is a collection of child language and the corresponding grammar might be less sparse and more ambiguous than adult grammars.

6 Conclusion

In this paper we have described our participating system for the dependency induction track of the PASCAL Challenge on Grammar Induction. Our system incorporates two types of inductive biases: the sparsity bias and the unambiguity bias. The sparsity bias favors a grammar with fewer grammar rules. We employ two types of sparsity biases: Dirichlet priors over grammar rule probabilities and the sparsity-inducing posterior regularization. The unambiguity bias favors a grammar that leads to unambiguous parses, which is motivated by the observation that natural language is remarkably unambiguous in the sense that the number of plausible

[1] Available at `http://code.google.com/p/pr-toolkit/`

Corpus	Approach	Parameters
Arabic	Dir+UR	maxlen = 20, valence = 4/4, back-off = 0.1, $\alpha = 10^{-5}$, $\sigma_u = 0.75$
Basque	PR+UR	maxlen = 10, valence = 3/3, back-off = 0.1, PR-AS, $\sigma_s = 100$, $\sigma_u = 0$
Czech	Dir+UR	maxlen = 10, valence = 3/3, back-off = 0.1, $\alpha = 1$, $\sigma_u = 1 - 0.1 \times iter$
Danish	PR+UR	maxlen = 20, valence = 2/1, back-off = 0.33, PR-AS, $\sigma_s = 100$, $\sigma_u = 0.5$
Dutch	PR+UR	maxlen = 10, valence = 3/3, back-off = 0, PR-S, $\sigma_s = 140$, $\sigma_u = 0$
English WSJ	Dir+UR	maxlen = 10, valence = 2/2, back-off = 0.33, $\alpha = 1$, $\sigma_u = 1 - 0.01 \times iter$
English CHILDES	Dir+UR	maxlen = 15, valence = 4/4, back-off = 0.1, $\alpha = 10$, $\sigma_u = 0$
Portuguese	PR+UR	maxlen = 15, valence = 2/1, back-off = 0, PR-AS, $\sigma_s = 140$, $\sigma_u = 0.5$
Slovene	PR+UR	maxlen = 10, valence = 4/4, back-off = 0.1, PR-AS, $\sigma_s = 140$, $\sigma_u = 0$
Swedish	Dir+UR	maxlen = 10, valence = 4/4, back-off = 0.1, $\alpha = 1$, $\sigma_u = 1 - 0.5 \times iter$

Table 1: For each corpus, the approach and the parameters that we found to perform the best on the development set and were hence used to learn the final grammar that produced the submitted predictions on the test set. In the second column, "Dir+UR" denotes our approach of unambiguity regularization with Dirichlet priors, and "PR+UR" denotes our approach of combined posterior regularization of sparsity and unambiguity. The parameters in the third column are explained in the main text.

parses of a natural language sentence is very small. We propose an approach named unambiguity regularization to induce unambiguity based on the posterior regularization framework. To combine Dirichlet priors with unambiguity regularization, we derive a mean-field variational inference algorithm. To combine the sparsity-inducing posterior regularization approach with unambiguity regularization, we employ a simplistic approach that optimizes the two regularization terms separately. We have also introduced our implementation and training procedure for the challenge. Our experimental results show that both types of inductive biases are beneficial to grammar induction.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *COMPUTATIONAL LINGUISTICS*, 19(2):313–330.

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

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

Kenji Sagae, Eric Davis, Alon Lavie, Brian MacWhinney, and Shuly Wintner. 2007. High-accuracy annotation and parsing of childes transcripts. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, CACLA '07, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kewei Tu and Vasant Honavar. 2012. Unambiguity regularization for unsupervised learning of probabilistic grammars. Technical report, Computer Science, Iowa State University.