Non-Local Modeling with a Mixture of PCFGs

Slav Petrov **Leon Barrett** **Dan Klein**
Computer Science Division, EECS Department
University of California at Berkeley
Berkeley, CA 94720
{petrov, lbarrett, klein}@eecs.berkeley.edu

Abstract

While most work on parsing with PCFGs has focused on local correlations between tree configurations, we attempt to model non-local correlations using a finite mixture of PCFGs. A mixture grammar fit with the EM algorithm shows improvement over a single PCFG, both in parsing accuracy and in test data likelihood. We argue that this improvement comes from the learning of specialized grammars that capture non-local correlations.

1 Introduction

The probabilistic context-free grammar (PCFG) formalism is the basis of most modern statistical parsers. The symbols in a PCFG encode context-freedom assumptions about statistical dependencies in the derivations of sentences, and the relative conditional probabilities of the grammar rules induce scores on trees. Compared to a basic treebank grammar (Charniak, 1996), the grammars of high-accuracy parsers weaken independence assumptions by splitting grammar symbols and rules with either lexical (Charniak, 2000; Collins, 1999) or non-lexical (Klein and Manning, 2003; Matsuzaki et al., 2005) conditioning information. While such splitting, or conditioning, can cause problems for statistical estimation, it can dramatically improve the accuracy of a parser.

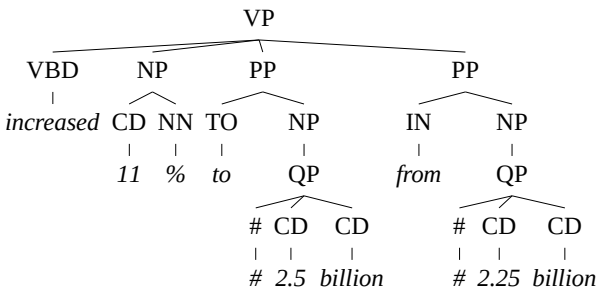
However, the configurations exploited in PCFG parsers are quite local: rules' probabilities may depend on parents or head words, but do not depend on arbitrarily distant tree configurations. For example, it is generally not modeled that if one quantifier phrase (QP in the Penn Treebank) appears in a sentence, the likelihood of finding another QP in that same sentence is greatly increased. This kind of effect is neither surprising nor unknown – for example, Bock and Loebell (1990) show experimentally that human language generation demonstrates priming effects. The mediating variables can not only include priming effects but also genre or stylistic conventions, as well as many other factors which are not adequately modeled by local phrase structure.

A reasonable way to add a latent variable to a generative model is to use a mixture of estimators, in this case a mixture of PCFGs (see Section 3). The general mixture of estimators approach was first suggested in the statistics literature by Titterington et al. (1962) and has since been adopted in machine learning (Ghahramani and Jordan, 1994). In a mixture approach, we have a new global variable on which all PCFG productions for a given sentence can be conditioned. In this paper, we experiment with a finite mixture of PCFGs. This is similar to the latent nonterminals used in Matsuzaki et al. (2005), but because the latent variable we use is global, our approach is more oriented toward learning non-local structure. We demonstrate that a mixture fit with the EM algorithm gives improved parsing accuracy and test data likelihood. We then investigate what is and is not being learned by the latent mixture variable. While mixture components are difficult to interpret, we demonstrate that the patterns learned are better than random splits.

2 Empirical Motivation

It is commonly accepted that the context freedom assumptions underlying the PCFG model are too

Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X),
pages 14–20, New York City, June 2006.



Rule	Score
QP → # CD CD	131.6
PRN → -LRB- ADJP -RRB	77.1
VP → VBD NP , PP PP	33.7
VP → VBD NP NP PP	28.4
PRN → -LRB- NP -RRB-	17.3
ADJP → QP	13.3
PP → IN NP ADVP	12.3
NP → NP PRN	12.3
VP → VBN PP PP PP	11.6
ADVP → NP RBR	10.1

Figure 1: Self-triggering: QP → # CD CD. If one British financial occurs in the sentence, the probability of seeing a second one in the same sentence is highly inreased. There is also a similar, but weaker, correlation for the American financial ($). On the right hand side we show the ten rules whose likelihoods are most increased in a sentence containing this rule.

strong and that weakening them results in better models of language (Johnson, 1998; Gildea, 2001; Klein and Manning, 2003). In particular, certain grammar productions often cooccur with other productions, which may be either near or distant in the parse tree. In general, there exist three types of correlations: (i) local (e.g. parent-child), (ii) non-local, and (iii) self correlations (which may be local or non-local).

In order to quantify the strength of a correlation, we use a likelihood ratio (LR). For two rules $\mathrm{X} \rightarrow \alpha$ and $\mathrm{Y} \rightarrow \beta$, we compute

$$LR(\mathrm{X} \rightarrow \alpha, \mathrm{Y} \rightarrow \beta) = \frac{\mathrm{P}(\alpha, \beta | X, Y)}{\mathrm{P}(\alpha | X, Y)\mathrm{P}(\beta | X, Y)}$$

This measures how much more often the rules occur together than they would in the case of independence. For rules that are correlated, this score will be high ($\gg 1$); if the rules are independent, it will be around 1, and if they are anti-correlated, it will be near 0.

Among the correlations present in the Penn Treebank, the local correlations are the strongest ones; they contribute 65% of the rule pairs with LR scores above 90 and 85% of those with scores over 200. Non-local and self correlations are in general common but weaker, with non-local correlations contributing approximately 85% of all correlations[1]. By adding a latent variable conditioning all productions, we aim to capture some of this interdependence between rules.

Correlations at short distances have been captured effectively in previous work (Johnson, 1998; Klein and Manning, 2003); vertical markovization (annotating nonterminals with their ancestor symbols) does this by simply producing a different distribution for each set of ancestors. This added context leads to substantial improvement in parsing accuracy. With local correlations already well captured, our main motivation for introducing a mixture of grammars is to capture long-range rule cooccurrences, something that to our knowledge has not been done successfully in the past.

As an example, the rule QP → # CD CD, representing a quantity of British currency, cooccurs with itself 132 times as often as if occurrences were independent. These cooccurrences appear in cases such as seen in Figure 1. Similarly, the rules VP → VBD NP PP , S and VP → VBG NP PP PP cooccur in the Penn Treebank 100 times as often as we would expect if they were independent. They appear in sentences of a very particular form, telling of an action and then giving detail about it; an example can be seen in Figure 2.

3 Mixtures of PCFGs

In a probabilistic context-free grammar (PCFG), each rule $\mathrm{X} \rightarrow \alpha$ is associated with a conditional probability $\mathrm{P}(\alpha | X)$ (Manning and Schütze, 1999). Together, these rules induce a distribution over trees $\mathrm{P}(T)$. A mixture of PCFGs enriches the basic model

[1]Quantifying the amount of non-local correlation is problematic; most pairs of cooccuring rules are non-local and will, due to small sample effects, have LR ratios greater than 1 even if they were truly independent in the limit.

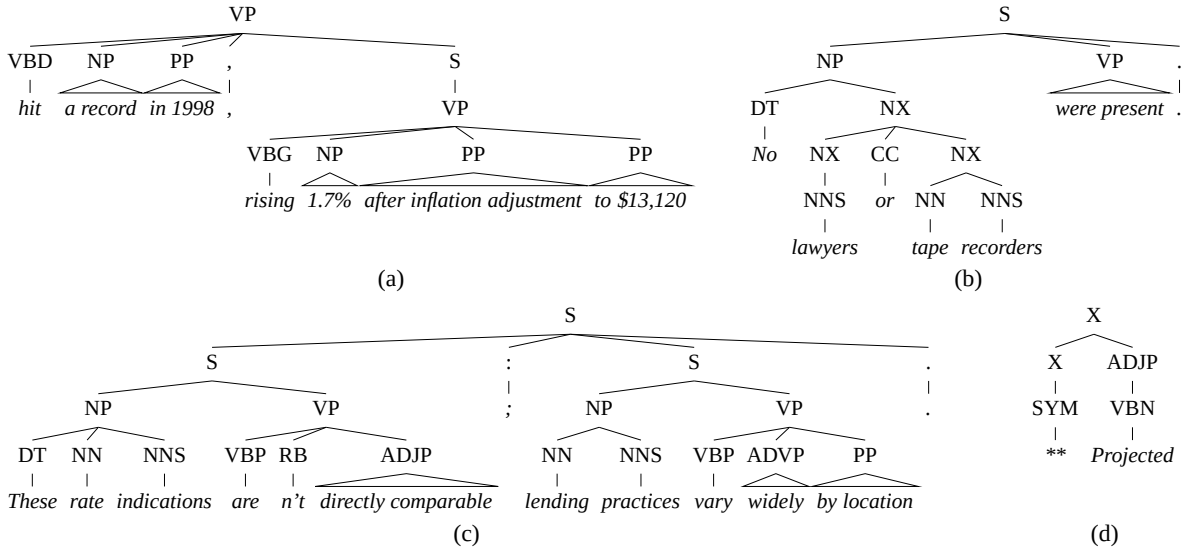


Figure 2: Tree fragments demonstrating coocurrences. (a) and (c) Repeated formulaic structure in one grammar: rules VP → VBD NP PP , S and VP → VBG NP PP PP and rules VP → VBP RB ADJP and VP → VBP ADVP PP. (b) Sibling effects, though not parallel structure, rules: NX → NNS and NX → NN NNS. (d) A special structure for footnotes has rules ROOT → X and X → SYM coocurring with high probability.

by allowing for multiple grammars, G_i, which we call *individual grammars*, as opposed to a single grammar. Without loss of generality, we can assume that the individual grammars share the same set of rules. Therefore, each original rule $\mathrm{X} \rightarrow \alpha$ is now associated with a vector of probabilities, $\mathrm{P}(\alpha|X, i)$. If, in addition, the individual grammars are assigned prior probabilities $\mathrm{P}(i)$, then the entire mixture induces a joint distribution over *derivations* $\mathrm{P}(T, i) = \mathrm{P}(i)\mathrm{P}(T|i)$ from which we recover a distribution over trees by summing over the grammar index i.

As a generative derivation process, we can think of this in two ways. First, we can imagine G to be a latent variable on which all productions are conditioned. This view emphasizes that any otherwise unmodeled variable or variables can be captured by the latent variable G. Second, we can imagine selecting an individual grammar G_i and then generating a sentence using that grammar. This view is associated with the expectation that there are multiple grammars for a language, perhaps representing different genres or styles. Formally, of course, the two views are the same.

3.1 Hierarchical Estimation

So far, there is nothing in the formal mixture model to say that rule probabilities in one component have any relation to those in other components. However, we have a strong intuition that many rules, such as NP → DT NN, will be common in all mixture components. Moreover, we would like to pool our data across components when appropriate to obtain more reliable estimators.

This can be accomplished with a hierarchical estimator for the rule probabilities. We introduce a *shared grammar* G_s. Associated to each rewrite is now a latent variable $L = \{\mathrm{S}, \mathrm{I}\}$ which indicates whether the used rule was derived from the shared grammar G_s or one of the individual grammars G_i:

$$\mathrm{P}(\alpha|X, i) = \lambda \mathrm{P}(\alpha|X, i, \ell{=}\mathrm{I}) + (1 - \lambda)\mathrm{P}(\alpha|X, i, \ell{=}\mathrm{S}),$$

where $\lambda \equiv P(\ell = \mathrm{I})$ is the probability of choosing the individual grammar and can also be viewed as a mixing coefficient. Note that $\mathrm{P}(\alpha|X, i, \ell{=}\mathrm{S}) = \mathrm{P}(\alpha|X, \ell{=}\mathrm{S})$, since the shared grammar is the same for all individual grammars. This kind of hierarchical estimation is analogous to that used in hierarchical mixtures of naive-Bayes for

text categorization (McCallum et al., 1998).

The hierarchical estimator is most easily described as a generative model. First, we choose a individual grammar G_i. Then, for each nonterminal, we select a level from the back-off hierarchy grammar: the individual grammar G_i with probability λ, and the shared grammar G_s with probability $1 - \lambda$. Finally, we select a rewrite from the chosen level. To emphasize: the derivation of a phrase-structure tree in a hierarchically-estimated mixture of PCFGs involves two kinds of hidden variables: the grammar G used for each sentence, and the level L used at each tree node. These hidden variables will impact both learning and inference in this model.

3.2 Inference: Parsing

Parsing involves inference for a given sentence S. One would generally like to calculate the *most probable parse* – that is, the tree T which has the highest probability $\mathrm{P}(T|S) \propto \sum_i \mathrm{P}(i)\mathrm{P}(T|i)$. However, this is difficult for mixture models. For a single grammar we have:

$$\mathrm{P}(T, i) = \mathrm{P}(i) \prod_{X \to \alpha \in T} \mathrm{P}(\alpha|X, i).$$

This score decomposes into a product and it is simple to construct a dynamic programming algorithm to find the optimal T (Baker, 1979). However, for a mixture of grammars we need to sum over the individual grammars:

$$\sum_i \mathrm{P}(T, i) = \sum_i \mathrm{P}(i) \prod_{X \to \alpha \in T} \mathrm{P}(\alpha|X, i).$$

Because of the outer sum, this expression unfortunately does not decompose into a product over scores of subparts. In particular, a tree which maximizes the sum need not be a top tree for any single component.

As is true for many other grammar formalisms in which there is a derivation / parse distinction, an alternative to finding the most probable parse is to find the *most probable derivation* (Vijay-Shankar and Joshi, 1985; Bod, 1992; Steedman, 2000). Instead of finding the tree T which maximizes $\sum_i \mathrm{P}(T, i)$, we find both the tree T and component i which maximize $\mathrm{P}(T, i)$. The most probable derivation can be found by simply doing standard PCFG parsing once for each component, then comparing the resulting trees' likelihoods.

3.3 Learning: Training

Training a mixture of PCFGs from a treebank is an incomplete data problem. We need to decide which individual grammar gave rise to a given observed tree. Moreover, we need to select a generation path (individual grammar or shared grammar) for each rule in the tree. To learn estimate parameters, we can use a standard Expectation-Maximization (EM) approach.

In the E-step, we compute the posterior distributions of the latent variables, which are in this case both the component G of each sentence and the hierarchy level L of each rewrite. Note that, unlike during parsing, there is no uncertainty over the actual rules used, so the E-step does not require summing over possible trees. Specifically, for the variable G we have

$$\mathrm{P}(i|T) = \frac{\mathrm{P}(T, i)}{\sum_j \mathrm{P}(T, j)}.$$

For the hierarchy level L we can write

$$\mathrm{P}(\ell = \textsc{i}|X \to \alpha, i, T) = \frac{\lambda \mathrm{P}(\alpha|X, \ell{=}\textsc{i})}{\lambda \mathrm{P}(\alpha|X, i, \ell{=}\textsc{i}) + (1 - \lambda)\mathrm{P}(\alpha|X, \ell{=}\textsc{s})},$$

where we slightly abuse notation since the rule $X \to \alpha$ can occur multiple times in a tree T.

In the M-step, we find the maximum-likelihood model parameters given these posterior assignments; i.e., we find the best grammars given the way the training data's rules are distributed between individual and shared grammars. This is done exactly as in the standard single-grammar model using relative expected frequencies. The updates are shown in Figure 3.3, where $\mathbf{T} = \{T_1, T_2, \dots\}$ is the training set.

We initialize the algorithm by setting the assignments from sentences to grammars to be uniform between all the individual grammars, with a small random perturbation to break symmetry.

4 Results

We ran our experiments on the Wall Street Journal (WSJ) portion of the Penn Treebank using the standard setup: We trained on sections 2 to 21, and we used section 22 as a validation set for tuning model hyperparameters. Results are reported

$$\mathrm{P}(i) \leftarrow \frac{\sum_{T_k \in \mathbf{T}} \mathrm{P}(i|T_k)}{\sum_i \sum_{T_k \in \mathbf{T}} \mathrm{P}(i|T_k)} = \frac{\sum_{T_k \in \mathbf{T}} \mathrm{P}(i|T_k)}{k}$$

$$\mathrm{P}(l = \text{I}) \leftarrow \frac{\sum_{T_k \in \mathbf{T}} \sum_{X \to \alpha \in T_k} \mathrm{P}(\ell = \text{I}|X \to \alpha)}{\sum_{T_k \in \mathbf{T}} |T_k|}$$

$$\mathrm{P}(\alpha|X, i, \ell = \text{I}) \leftarrow \frac{\sum_{T_k \in \mathbf{T}} \sum_{X \to \alpha \in T_k} \mathrm{P}(i|T_k)\mathrm{P}(\ell = \text{I}|T_k, i, X \to \alpha)}{\sum_{\alpha'} \sum_{T_k \in \mathbf{T}} \sum_{X \to \alpha' \in T_k} \mathrm{P}(i|T_k)\mathrm{P}(\ell = \text{I}|T_k, i, X \to \alpha')}$$

Figure 3: Parameter updates. The shared grammar's parameters are re-estimated in the same manner.

on all sentences of 40 words or less from section 23. We use a markovized grammar which was annotated with parent and sibling information as a baseline (see Section 4.2). Unsmoothed maximum-likelihood estimates were used for rule probabilities as in Charniak (1996). For the tagging probabilities, we used maximum-likelihood estimates for $\mathrm{P}(tag|word)$. Add-one smoothing was applied to unknown and rare (seen ten times or less during training) words before inverting those estimates to give $\mathrm{P}(word|tag)$. Parsing was done with a simple Java implementation of an agenda-based chart parser.

4.1 Parsing Accuracy

The EM algorithm is guaranteed to continuously increase the likelihood on the training set until convergence to a local maximum. However, the likelihood on unseen data will start decreasing after a number of iterations, due to overfitting. This is demonstrated in Figure 4. We use the likelihood on the validation set to stop training before overfitting occurs.

In order to evaluate the performance of our model, we trained mixture grammars with various numbers of components. For each configuration, we used EM to obtain twelve estimates, each time with a different random initialization. We show the F_1-score for the model with highest log-likelihood on the validation set in Figure 4. The results show that a mixture of grammars outperforms a standard, single grammar PCFG parser.[2]

4.2 Capturing Rule Correlations

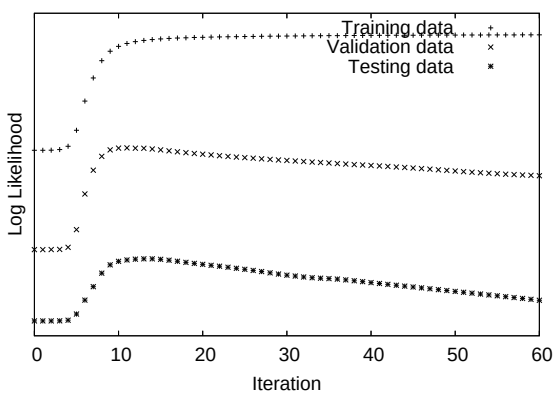
As described in Section 2, we hope that the mixture model will capture long-range correlations in the data. Since local correlations can be captured by adding parent annotation, we combine our mixture model with a grammar in which node probabilities depend on the parent (the last vertical ancestor) and the closest sibling (the last horizontal ancestor). Klein and Manning (2003) refer to this grammar as a markovized grammar of vertical order = 2 and horizontal order = 1. Because many local correlations are captured by the markovized grammar, there is a greater hope that observed improvements stem from non-local correlations.

In fact, we find that the mixture does capture non-local correlations. We measure the degree to which a grammar captures correlations by calculating the total squared error between LR scores of the grammar and corpus, weighted by the probability of seeing nonterminals. This is 39422 for a single PCFG, but drops to 37125 for a mixture with five individual grammars, indicating that the mixture model better captures the correlations present in the corpus. As a concrete example, in the Penn Treebank, we often see the rules FRAG → ADJP and PRN → , SBAR , cooccurring; their LR is 134. When we learn a single markovized PCFG from the treebank, that grammar gives a likelihood ratio of only 61. However, when we train with a hierarchical model composed of a shared grammar and four individual grammars, we find that the grammar likelihood ratio for these rules goes up to 126, which is very similar to that of the empirical ratio.

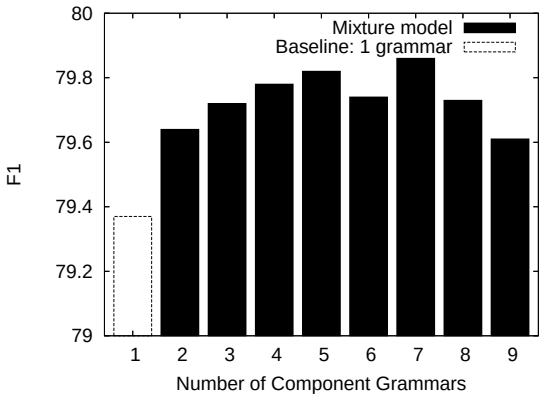
4.3 Genre

The mixture of grammars model can equivalently be viewed as capturing either non-local correlations or variations in grammar. The latter view suggests that the model might benefit when the syntactic structure

[2] This effect is statistically significant.



(a)



(b)

Figure 4: (a) Log likelihood of training, validation, and test data during training (transformed to fit on the same plot). Note that when overfitting occurs the likelihood on the validation and test data starts decreasing (after 13 iterations). (b) The accuracy of the mixture of grammars model with $\lambda = 0.4$ versus the number of grammars. Note the improvement over a 1-grammar PCFG model.

varies significantly, as between different genres. We tested this with the Brown corpus, of which we used 8 different genres (*f*, *g*, *k*, *l*, *m*, *n*, *p*, and *r*). We follow Gildea (2001) in using the ninth and tenth sentences of every block of ten as validation and test data, respectively, because a contiguous test section might not be representative due to the genre variation.

To test the effects of genre variation, we evaluated various training schemes on the Brown corpus. The single grammar baseline for this corpus gives $F_1 = 79.75$, with log likelihood (LL) on the testing data=-242561. The first test, then, was to estimate each individual grammar from only one genre. We did this by assigning sentences to individual grammars by genre, without using any EM training. This increases the data likelihood, though it reduces the F_1 score ($F_1 = 79.48$, LL=-242332). The increase in likelihood indicates that there *are* genre-specific features that our model can represent. (The lack of F_1 improvement may be attributed to the increased difficulty of estimating rule probabilities after dividing the already scant data available in the Brown corpus. This small quantity of data makes overfitting almost certain.)

However, local minima and lack of data cause difficulty in learning genre-specific features. If we start with sentences assigned by genre as before, but then train with EM, both F_1 and test data log likelihood drop ($F_1 = 79.37$, LL=-242100). When we use EM with a random initialization, so that sentences are not assigned directly to grammars, the scores go down even further ($F_1 = 79.16$, LL=-242459). This indicates that the model can capture variation between genres, but that maximum training data likelihood does not necessarily give maximum accuracy. Presumably, with more genre-specific data available, learning would generalize better. So, genre-specific grammar variation is real, but it is difficult to capture via EM.

4.4 Smoothing Effects

While the mixture of grammars captures rule correlations, it may also enhance performance via smoothing effects. Splitting the data randomly could produce a smoothed shared grammar, G_s, that is a kind of held-out estimate which could be superior to the unsmoothed ML estimates for the single-component grammar.

We tested the degree of generalization by evaluating the shared grammar alone and also a mixture of the shared grammar with the known single grammar. Those shared grammars were extracted after training the mixture model with four individual grammars. We found that both the shared grammar alone (F_1=79.13, LL=-333278) and the shared grammar mixed with the single grammar (F_1=79.36, LL=-331546) perform worse than a sin-

gle PCFG (F_1=79.37, LL=-327658). This indicates that smoothing is not the primary learning effect contributing to increased F_1.

5 Conclusions

We examined the sorts of rule correlations that may be found in natural language corpora, discovering non-local correlations not captured by traditional models. We found that using a model capable of representing these non-local features gives improvement in parsing accuracy and data likelihood. This improvement is modest, however, primarily because local correlations are so much stronger than non-local ones.

References

J. Baker. 1979. Trainable grammars for speech recognition. *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550.

K. Bock and H. Loebell. 1990. Framing sentences. *Cognition*, 35:1–39.

R. Bod. 1992. A computational model of language performance: Data oriented parsing. *International Conference on Computational Linguistics (COLING)*.

E. Charniak. 1996. Tree-bank grammars. In *Proc. of the 13th National Conference on Artificial Intelligence (AAAI)*, pages 1031–1036.

E. Charniak. 2000. A maximum–entropy–inspired parser. In *Proc. of the Conference of the North American chapter of the Association for Computational Linguistics (NAACL)*, pages 132–139.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, Univ. of Pennsylvania.

Z. Ghahramani and M. I. Jordan. 1994. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NIPS)*, pages 120–127.

D. Gildea. 2001. Corpus variation and parser performance. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

M. Johnson. 1998. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24:613–632.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. *Proc. of the 41st Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430.

C. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

T. Matsuzaki, Y. Miyao, and J. Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proc. of the 43rd Meeting of the Association for Computational Linguistics (ACL)*, pages 75–82.

A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Int. Conf. on Machine Learning (ICML)*, pages 359–367.

M. Steedman. 2000. *The Syntactic Process.* The MIT Press, Cambridge, Massachusetts.

D. Titterington, A. Smith, and U. Makov. 1962. *Statistical Analysis of Finite Mixture Distributions*. Wiley.

K. Vijay-Shankar and A. Joshi. 1985. Some computational properties of tree adjoining grammars. *Proc. of the 23th Meeting of the Association for Computational Linguistics (ACL)*, pages 82–93.