A Distributional Analysis of a Lexicalized Statistical Parsing Model

Daniel M. Bikel
Department of Computer and Information Science
University of Pennsylvania
3330 Walnut Street
Philadelphia, PA 19104
dbikel@cis.upenn.edu

Abstract

This paper presents some of the first data visualizations and analysis of distributions for a lexicalized statistical parsing model, in order to better understand their nature. In the course of this analysis, we have paid particular attention to parameters that include bilexical dependencies. The prevailing view has been that such statistics are very informative but suffer greatly from sparse data problems. By using a parser to constrain-parse its own output, and by hypothesizing and testing for distributional similarity with back-off distributions, we have evidence that finally explains that (a) bilexical statistics are actually getting used quite often but that (b) the distributions are so similar to those that do not include head words as to be nearly indistinguishable insofar as making parse decisions. Finally, our analysis has provided for the first time an effective way to do parameter selection for a generative lexicalized statistical parsing model.

1 Introduction

Lexicalized statistical parsing models, such as those built by Black et al. (1992a), Magerman (1994), Collins (1999) and Charniak (2000), have been enormously successful, but they also have an enormous complexity. Their success has often been attributed to their sensitivity to individual lexical items, and it is precisely this incorporation of lexical items into features or parameter schemata that gives rise to their complexity. In order to help determine which features are helpful, the somewhat crude-but-effective method has been to compare a model's overall parsing performance with and without a feature. Often, it has seemed that features that are derived from linguistic principles result in higher-performing models (cf. (Collins, 1999)). While this may be true, it is clearly inappropriate to highlight *ex post facto* the linguistically-motivated features and rationalize their inclusion and state how effective they are. A rigorous analysis of features or parameters in relation to the entire model is called for. Accordingly, this work aims to provide a thorough analysis of the nature of the parameters in a Collins-style parsing model, with particular focus on the two parameter classes that generate lexicalized modifying nonterminals, for these are where all a sentence's words are generated except for the head word of the entire sentence; also, these two parameter classes have by far the most parameters and suffer the most from sparse data problems. In spite of using a Collins-style model as the basis for analysis, throughout this paper, we will attempt to present information that is widely applicable because it pertains to properties of the widely-used Treebank (Marcus et al., 1993) and lexicalized parsing models in general.

This work also sheds light on the much-discussed "bilexical dependencies" of statistical parsing models. Beginning with the seminal work at IBM (Black et al., 1991; Black et al., 1992b; Black et al., 1992a), and continuing with such lexicalist approaches as (Eisner, 1996), these features have been lauded for their ability to approximate a word's semantics as a means to override syntactic preferences with semantic ones (Collins, 1999; Eisner, 2000). However, the work of Gildea (2001) showed that, with an approximate reimplementation of Collins' Model 1, removing all parameters that involved dependencies between a modifier word and its head resulted in a surprisingly small decrease in overall parse accuracy. The prevailing assumption was then that such bilexical statistics were *not* useful for making syntactic decisions, although it was not entirely clear why. Subsequently, we replicated Gildea's experiment with a complete emulation of Model 2 and presented additional evidence that bilexical statistics were barely getting used during decoding (Bikel, 2004), appearing to confirm the original result. However, the present work will show that such statistics *do* get frequently used for the highest-probability parses, but that when a Collins-style model generates modifier words, the bilexical parameters are so similar to their back-off distributions as to provide almost no extra predictive infor-

mation.

2 Motivation

A parsing model coupled with a decoder (an algorithm to search the space of possible trees for a given terminal sequence) is largely an engineering effort. In the end, the performance of the parser with respect to its evaluation criteria—typically accuracy, and perhaps also speed—are all that matter. Consequently, the engineer must understand what the model is doing only to the point that it helps make the model perform better. Given the somewhat crude method of determining a feature's benefit by testing a model with and without the feature, a researcher can argue for the efficacy of that feature without truly understanding its effect on the model. For example, while adding a particular feature may improve parse accuracy, the reason may have little to do with the nature of the feature and everything to do with its canceling other features that were theretofore *hurting* performance. In any case, since this is engineering, the rationalization for a feature is far less important than the model's overall performance increase.

On the other hand, science would demand that, at some point, we analyze the multitude of features in a state-of-the-art lexicalized statistical parsing model. Such analysis is warranted for two reasons: replicability and progress. The first is a basic tenet of most sciences: without proper understanding of what has been done, the relevant experiment(s) cannot be replicated and therefore verified. The second has to do with the idea that, when a discipline matures, it can be difficult to determine what new features can provide the most gain (or *any* gain, for that matter). A thorough analysis of the various distributions being estimated in a parsing model allows researchers to discover what is being learned most and least well. Understanding what is learned most well can shed light on the *types* of features or dependencies that are most efficacious, pointing the way to new features of that type. Understanding what is learned least well defines the space in which to look for those new features.

3 Frequencies

3.1 Definitions and notation

In this paper we will refer to any estimated distribution as a *parameter* that has been instantiated from a *parameter class*. For example, in an *n*-gram language model, $p(w_i \mid w_{i-1})$ is a parameter class, whereas the estimated distribution $\hat{p}(\cdot \mid \texttt{the})$ is a particular parameter from this class, consisting of estimates of every word that can follow the word "the".

For this work, we used the model described in (Bikel, 2002; Bikel, 2004). Our emulation of Collins' Model 2 (hereafter referred to simply as "the model") has eleven parameter classes, each of which employs up to three back-off levels, where back-off level 0 is just the "un-backed-off" maximal context history.[1] In other words, a smoothed probability estimate is the interpolation of up to three different unsmoothed estimates. The notation and description for each of these parameter classes is shown in Table 1.

3.2 Basic frequencies

Before looking at the number of parameters in the model, it is important to bear in mind the amount of data on which the model is trained and on which actual parameters will be induced from parameter classes. The standard training set for English consists of Sections 02–21 of the Penn Treebank, which in turn consist of 39,832 sentences with a total of 950,028 word tokens (not including null elements). There are 44,113 unique words (again, not including null elements), 10,437 of which occur 6 times or more.[2] The trees consist of 904,748 brackets with 28 basic nonterminal labels, to which function tags such as `-TMP` and indices are added in the data to form 1184 observed nonterminals, not including preterminals. After tree transformations, the model maps these 1184 nonterminals down to just 43. There are 42 unique part of speech tags that serve as preterminals in the trees; the model prunes away three of these (`"`, `“` and `.`).

Induced from these training data, the model contains 727,930 parameters; thus, there are nearly as many parameters as there are brackets or word tokens. From a history-based grammar perspective, there are 727,930 types of history contexts from which futures are generated. However, 401,447 of these are singletons. The average count for a history context is approximately 35.56, while the average diversity is approximately 1.72. The model contains 1,252,280 unsmoothed maximum-likelihood probability estimates ($727,930 \cdot 1.72 \approx 1,252,280$). Even when a given future was not seen with a particular history, it is possible that one of its associated

[1] Collins' model splits out the P_M and P_{M_w} classes into left- and right-specific versions, and has two additional classes for dealing with coordinating conjunctions and inter-phrasal punctuation. Our emulation of Collins' model incorporates the information of these specialized parameter classes into the existing P_M and P_{M_w} parameters.

[2] We mention this statistic because Collins' thesis experiments were performed with an unknown word threshold of 6.

Notation	Description	No. of back-off levels
P_H	Generates unlexicalized head child given lexicalized parent	3
P_{subcat_L}	Generates subcat bag on left side of head child	3
P_{subcat_R}	Generates subcat bag on right side of head child	3
P_M ($P_{M,\texttt{NPB}}$)	Generates partially-lexicalized modifying nonterminal (with NPB parent)	3
P_{M_w} ($P_{M_w,\texttt{NPB}}$)	Generates head word of modifying nonterminal (with NPB parent)	3
$P_{prior_{NT}}$	Priors for nonterminal conditioning on its head word and part of speech	2
$P_{prior_{lex}}$	Priors for head word/part of speech pairs (unconditional probabilities)	0
$P_{TOP_{NT}}$	Generates partially-lexicalized child of +TOP+†	1
P_{TOP_w}	Generates the head word for children of +TOP+†	2

Table 1: All eleven parameter classes in our emulation of Collins' Model 2. A partially-lexicalized nonterminal is a nonterminal label and its head word's part of speech (such as NP(NN)). †The hidden nonterminal +TOP+ is added during training to be the parent of every observed tree.

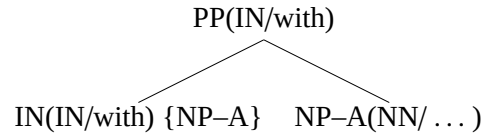


Figure 1: A frequent P_{M_w} history context, illustrated as a tree fragment. The . . . represents the future that is to be generated given this history.

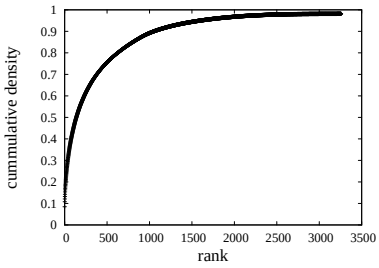


Figure 2: Cumulative density function for the P_{M_w} history context illustrated in Figure 1.

back-off contexts was seen with that future, leading to a non-zero smoothed estimate. The total number of possible non-zero smoothed estimates in the model is 562,596,053. Table 2 contains count and diversity statistics for the two parameter classes on which we will focus much of our attention, P_M and P_{M_w}. Note how the maximal-context back-off levels (level 0) for both parameter classes have relatively little training: on average, raw estimates are obtained with history counts of only 10.3 and 4.4 in the P_M and P_{M_w} classes, respectively. Conversely, observe how drastically the average number of transitions $\overline{n}$ increases as we remove dependence on the head word going from back-off level 0 to 1.

3.3 Exploratory data analysis: a common distribution

To begin to get a handle on these distributions, particularly the relatively poorly-trained and/or high-entropy distributions of the P_{M_w} class, it is useful to perform some exploratory data analysis. Figure 1 illustrates the 25th-most-frequent P_{M_w} history context as a tree fragment. In the top-down model, the following elements have been generated:

- a parent nonterminal PP(IN/with) (a PP headed by the word with with the part-of-speech tag IN)
- the parent's head child IN
- a right subcat bag containing NP-A (a single NP argument must be generated somewhere on the right side of the head child)
- a partially-lexicalized right-modifying nonterminal

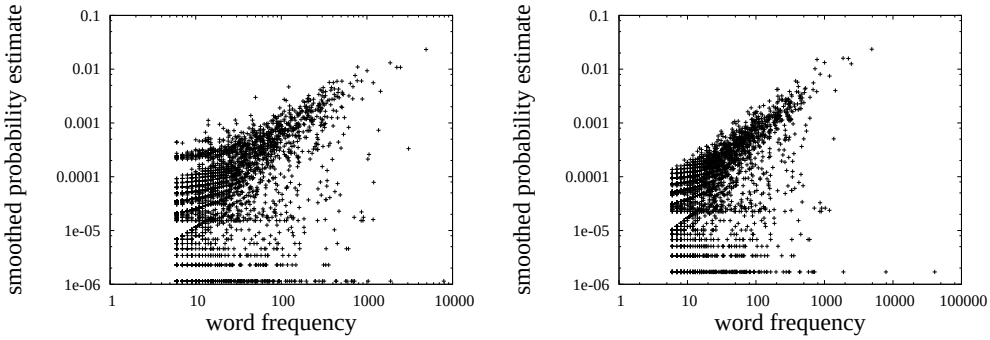
At this point in the process, a P_{M_w} parameter conditioning on all of this context will be used to estimate the probability of the head word of the NP-A(NN), completing the lexicalization of that nonterminal. If a candidate head word was seen in training in this configuration, then it will be generated conditioning on the full context that crucially includes the head word with; otherwise, the model will back off to a history context that does not include the head word.

In Figure 2, we plot the cumulative density function of this history context. We note that of the 3258 words with non-zero probability in this context, 95% of the probability mass is covered by the 1596 most likely words.

In order to get a better visualization of the probability distribution, we plotted smoothed probability estimates versus the training-data frequencies of the words being generated. Figure 3(a) shows smoothed estimates that make use of the full context (*i.e.*, include the head word with) wherever possible, and Figure 3(b) shows smoothed estimates that do not use the head word. Note how the plot in Figure 3(b) appears remarkably similar to the "true" distribu-

Back-off level	P_M			$P_{M_{w\_}}$		
	$\overline{c}$	$\overline{d}$	$\overline{n}$	$\overline{c}$	$\overline{d}$	$\overline{n}$
0	10.268	1.437	7.145	4.413	1.949	2.264
1	558.047	3.643	153.2	60.19	8.454	7.120
2	1169.6	5.067	230.8	21132.1	370.6	57.02

Table 2: Average counts and diversities of histories of the P_M and P_{M_w} parameter classes. $\overline{c}$ and $\overline{d}$ are average history count and diversity, respectively. $\overline{n} = \frac{\overline{c}}{\overline{d}}$ is the average number of transitions from a history context to some future.



(a) prob. vs. word freq., back-off level 1

(b) prob. vs. word freq., back-off level 2

Figure 3: Probability versus word frequency for head words of `NP-A(NN)` in the `PP` construction.

tion of 3(a). 3(b) looks like a slightly "compressed" version of 3(b) (in the vertical dimension), but the shape of the two distributions appears to be roughly the same. This observation will be confirmed and quantified by the experiments of §5.[3]

4 Entropies

A good measure of the discriminative efficacy of a parameter is its entropy. Table 3 shows the average entropy of all distributions for each parameter class.[4] By far the highest average entropy is for the P_{M_w} parameter class.

Having computed the entropy for every distribution in every parameter class, we can actually plot a "meta-distribution" of entropies for a parameter class, as shown in Figure 4. As an example of one of the data points of Figure 4, consider the history context explored in the previous section. While it may be one of the most frequent, it also has the highest entropy at 9.141 bits, as shown by Table 4. This value not only confirms but quantifies the long-held intuition that PP-attachment requires more than just the local phrasal context; it is, *e.g.*, precisely why the PP-specific features of (Collins, 2000) were likely to be very helpful, as cases such as these are among the most difficult that the model must discriminate. In fact, of the top 50 of the highest-entropy

P_H	0.2516	$P_{TOP_{NT}}$	2.517
P_{subcat_L}	0.02342	P_{TOP_w}	2.853
P_{subcat_R}	0.2147		
P_M	1.121		
P_{M_w}	3.923		

Table 3: Average entropies for each parameter class.

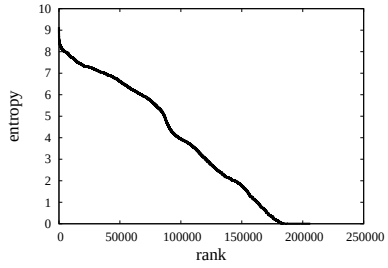


Figure 4: Entropy distribution for the P_{M_w} parameters.

[3] The astute reader will further note that the plots in Figure 3 both look bizarrely truncated with respect to low-frequency words. This is simply due to the fact that all words below a fixed frequency are generated as the `+UNKNOWN+` word.

[4] The decoder makes use of two additional parameter classes that jointly estimate the prior probability of a lexicalized nonterminal; however, these two parameter classes are not part of the generative model.

Back-off level	P_M min	max	avg	median	P_{M_w} min	max	avg	median
0	3.080E-10	4.351	1.128	0.931	4.655E-8	9.141	3.904	3.806
1	4.905E-7	4.254	0.910	0.667	2.531E-6	9.120	4.179	4.224
2	8.410E-4	3.501	0.754	0.520	0.002	8.517	3.182	2.451
Overall	3.080E-10	**4.351**	1.121	**0.917**	4.655E-8	**9.141**	3.922	**3.849**

Table 4: Entropy distribution statistics for P_M and P_{M_w}.

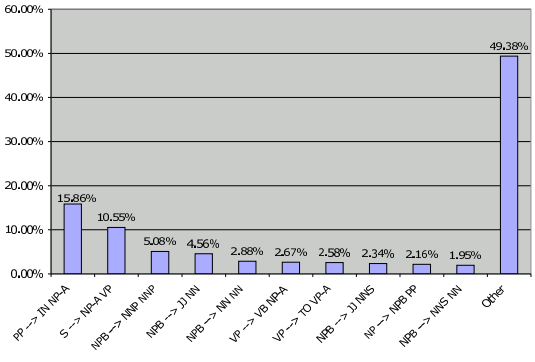


Figure 5: Total modifier word–generation entropy broken down by parent-head-modifier triple.

distributions from P_{M_w}, 25 involve the configuration `PP --> IN(IN/<prep>) NP-A(NN/...)`, where <prep> is some preposition whose tag is `IN`. Somewhat disturbingly, these are also some of the most frequent constructions.

To gauge roughly the importance of these high-frequency, high-entropy distributions, we performed the following analysis. Assume for the moment that every word-generation decision is roughly independent from all others (this is clearly not true, given head-propagation). We can then compute the total entropy of word-generation decisions for the entire training corpus via

$$H_{P_{M_w}} = \sum_{c \in P_{M_w}} f(c) \cdot H(c) \qquad (1)$$

where $f(c)$ is the frequency of some history context c and $H(c)$ is that context's entropy. The total modifier word-generation entropy for the corpus with the independence assumption is 3,903,224 bits. Of these, the total entropy for contexts of the form `PP` $\rightarrow$ `IN NP-A` is 618,640 bits, representing a sizable 15.9% of the total entropy, and the single largest percentage of total entropy of any parent-head-modifier triple (see Figure 5).

On the opposite end of the entropy spectrum, there are tens of thousands of P_{M_w} parameters with extremely low entropies, mostly having to do with extremely low-diversity, low-entropy part-of-speech tags, such as `DT`, `CC`, `IN` or `WRB`. Perhaps even more interesting is the number of distributions with *identical* entropies: of the 206,234 distributions, there are only 92,065 unique entropy values. Distributions with the same entropy are all candidates for removal from the model, because most of their probability mass resides in the back-off distribution. Many of these distributions are low- or one-count history contexts, justifying the common practice of removing transitions whose history count is below a certain threshold. This practice could be made more rigorous by relying on distributional similarity. Finally, we note that the most numerous low-entropy distributions (that are not trivial) involve generating right-modifier words of the head child of an `SBAR` parent. The model is able to learn these constructions extremely well, as one might expect.

5 Distributional similarity and bilexical statistics

We now return to the issue of bilexical statistics. As alluded to earlier, Gildea (2001) performed an experiment with his partial reimplementation of Collins' Model 1 in which he removed the maximal-context back-off level from P_{M_w}, which effectively removed all bilexical statistics from his model. Gildea observed that this change resulted in only a 0.5% drop in parsing performance. There were two logical possibilities for this behavior: either such statistics were not getting used due to sparse data problems, or they were not informative for some reason. The prevailing view of the NLP community had been that bilexical statistics were sparse, and Gildea (2001) adopted this view to explain his results. Subsequently, we duplicated Gildea's experiment with a complete emulation of Collins' Model 2, and found that when the decoder requested a smoothed estimate involving a bigram when testing on held-out data, it only received an estimate that made use of bilexical statistics a mere 1.49% of the time (Bikel, 2004). The conclusion was that the minuscule drop in performance from removing bigrams must have been due to the fact that they were barely able to be used. In other words, it appeared that bigram coverage was not nearly good enough for bigrams to have an impact on parsing

performance, seemingly confirming the prevailing view.

But the 1.49% figure does not tell the whole story. The parser pursues many incorrect and ultimately low-scoring theories in its search (in this case, using probabilistic CKY). So rather than asking how many times the decoder makes use of bigram statistics *on average*, a better question is to ask how many times the decoder can use bigram statistics *while pursuing the top-ranked theory*. To answer this question, we used our parser to *constrain-parse* its own output. That is, having trained it on Sections 02–21, we used it to parse Section 00 of the Penn Treebank (the canonical development test set) and then *re-parse* that section using its own highest-scoring trees (without lexicalization) as constraints, so that it only pursued theories consistent with those trees. As it happens, the number of times the decoder was able to use bigram statistics shot up to 28.8% overall, with a rate of 22.4% for `NPB` constituents.

So, bigram statistics *are* getting used; in fact, they are getting used more than 19 times as often when pursuing the highest-scoring theory as when pursuing any theory on average. And yet there is no disputing the fact that their use has a surprisingly small effect on parsing performance. The exploratory data analysis of §3.3 suggests an explanation for this perplexing behavior: the distributions that include the head word versus those that do not are so similar as to make almost no difference in terms of parse accuracy.

5.1 Distributional similarity

A useful metric for measuring distributional similarity, as explored by (Lee, 1999), is the *Jensen-Shannon divergence* (Lin, 1991):

$$JS(p \,\|\, q) = \frac{1}{2}\left[D\left(p \,\middle\|\, \text{avg}_{p,q}\right) + D\left(q \,\middle\|\, \text{avg}_{p,q}\right)\right] \tag{2}$$

where D is the Kullback-Leibler divergence (Cover and Thomas, 1991) and where $\text{avg}_{p,q} = \frac{1}{2}(p(A) + q(A))$ for an event A in the event space of at least one of the two distributions. One interpretation for the Jensen-Shannon divergence due to Slonim et al. (2002) is that it is related to the log-likelihood that "the two sample distributions originate by the most likely common source," relating the quantity to the "two-sample problem".

In our case, we have $p = p(y \mid x_1, x_2)$ and $q = p(y \mid x_1)$, where y is a possible future and x_1, x_2 are elements of a history context, with q representing a back-off distribution using less context. Therefore, whereas the standard JS formulation is agnostic with respect to its two distributions, and averages them in part to ensure that the quantity is defined over the entire space, we have the prior knowledge that one history context is a superset of the other, that $\langle x_1 \rangle$ is defined wherever $\langle x_1, x_2 \rangle$ is. In this case, then, we have a simpler, "one-sided" definition for the Jensen-Shannon divergence, but generalized to the multiple distributions that include an extra history component:

$$\begin{aligned} JS(p \,\|\, q) &= \\ &\sum_{x_2} p(x_2) \cdot D\left(p(y \mid x_1, x_2) \,\|\, p(y \mid x_1)\right) \\ &= E_{x_2} D\left(p(y \mid x_1, x_2) \,\|\, p(y \mid x_1)\right) \end{aligned} \tag{3}$$

An interpretation in our case is that this is the expected number of bits x_2 gives you when trying to predict y.[5] If we allow x_2 to represent an arbitrary amount of context, then the Jensen-Shannon divergence $JS_{b \leftarrow a} = JS(p_b \,\|\, p_a)$ can be computed for any two back-off levels, where a, b are back-off levels s.t. $b < a$ (meaning p_b is a distribution using more context than p_a). The actual value in bits of the Jensen-Shannon divergence between two distributions should be considered in relation to the number of bits of entropy of the more detailed distribution; that is, $JS_{b \leftarrow a}$ should be considered relative to $H(p_b)$. Having explored entropy in §4, we will now look at some summary statistics for JS divergence.

	min	max	avg.	median
$JS_{0 \leftarrow 1}$	2.729E-7	2.168	**0.1148**	**0.09672**
$JS_{1 \leftarrow 2}$	0.001318	1.962	0.6929	0.6986
$JS_{0 \leftarrow 2}$	0.001182	1.180	0.3774	0.3863

Table 5: Jensen-Shannon statistics for back-off parameters in P_{M_w}.

5.2 Results

We computed the quantity in Equation 3 for every parameter in P_{M_w} that used maximal context (contained a head word) and its associated parameter that did not contain the head word. The results are listed in Table 5. Note that, for this parameter class with a median entropy of 3.8 bits, we have a median JS divergence of only 0.097 bits. The distributions are so similar that the 28.8% of the time that the decoder uses an estimate based on a bigram, it might as well be using one that does not include the head word.

[5] Or, following from Slonim et al.'s interpretation, this quantity is the (negative of the) log-likelihood that all distributions that include an x_2 component come from a "common source" that does *not* include this component.

	≤ 40 words			
	§00		§23	
Model	LR	LP	LR	LP
m3	n/a	n/a	88.6	88.7
m2-emu	89.9	90.0	88.8	88.9
reduced	90.0	90.2	88.7	88.9
	all sentences			
Model	§00		§23	
m3	n/a	n/a	88.0	88.3
m2-emu	88.8	89.0	88.2	88.3
reduced	89.0	89.0	88.0	88.2

Table 6: Parsing results on Sections 00 and 23 with Collins' Model 3, our emulation of Collins' Model 2 and the reduced version at a threshold of 0.06. LR = labeled recall, LP = labeled precision.[6]

6 Distributional Similarity and Parameter Selection

The analysis of the previous two sections provides a window onto what types of parameters the parsing model is learning most and least well, and onto what parameters carry more and less useful information. Having such a window holds the promise of discovering new parameter types or features that would lead to greater parsing accuracy; such is the scientific, or at least, the forward-minded research perspective.

From a much more purely engineering perspective, one can also use the analysis of the previous two sections to identify individual parameters that carry little to no useful information and simply remove them from the model. Specifically, if p_b is a particular distribution and p_{b+1} is its corresponding back-off distribution, then one can remove all parameters p_b such that

$$\frac{JS(p_b \| p_{b+1})}{H(p_b)} < t,$$

where $0 < t < 1$ is some threshold. Table 6 shows the results of this experiment using a threshold of 0.06. To our knowledge, this is the first example of detailed parameter selection in the context of a generative lexicalized statistical parsing model. The consequence is a significantly smaller model that performs *with no loss of accuracy* compared to the full model.[6]

Further insight is gained by looking at the percentage of parameters removed from each parameter class. The results of (Bikel, 2004) suggested that the power of Collins-style parsing models did not lie primarily with the use of bilexical dependencies as was once thought, but in *lexico-structural* dependencies, that is, predicting syntactic structures conditioning on head words. The percentages of Table 7 provide even more concrete evidence of this assertion, for whereas nearly a third of the P_{M_w} parameters were removed, a much smaller fraction of parameters were removed from the P_{subcat_L}, P_{subcat_R} and P_M classes that generate structure conditioning on head words.

P_H	13.5%	P_{TOP_w}	0.023%
P_{subcat_L}	0.67%	P_M	10.1%
P_{subcat_R}	1.8%	P_{M_w}	29.4%

Table 7: Percentage of parameters removed from each parameter class for the 0.06-reduced model.

7 Discussion

Examining the lower-entropy P_{M_w} distributions revealed that, in many cases, the model was not so much learning how to disambiguate a given syntactic/lexical choice, but simply not having much to learn. For example, once a partially-lexicalized nonterminal has been generated whose tag is fairly specialized, such as `IN`, then the model has "painted itself into a lexical corner", as it were (the extreme example is `TO`, a tag that can only be assigned to the word `to`). This is an example of the "label bias" problem, which has been the subject of recent discussion (Lafferty et al., 2001; Klein and Manning, 2002). Of course, just because there is "label bias" does not necessarily mean there is a problem. If the decoder pursues a theory to a nonterminal/part-of-speech tag preterminal that has an extremely low entropy distribution for possible head words, then there is certainly a chance that it will get "stuck" in a potentially bad theory. This is of particular concern when a head word—which the top-down model generates at its highest point in the tree—influences an attachment decision. However, inspecting the low-entropy word-generation histories of P_{M_w} revealed that almost all such cases are when the model is generating a preterminal, and are thus of little to no consequence vis-a-vis syntactic disambiguation.

8 Conclusion and Future Work

With so many parameters, a lexicalized statistical parsing model seems like an intractable behemoth. However, as statisticians have long known, an excellent angle of attack for a mass of unruly data is exploratory data analysis. This paper presents some of the first data visualizations of parameters

[6] None of the differences between the Model 2–emulation results and the reduced model results is statistically significant.

in a parsing model, and follows up with a numerical analysis of properties of those distributions. In the course of this analysis, we have focused in on the question of bilexical dependencies. By constrain-parsing the parser's own output, and by hypothesizing and testing for distributional similarity, we have presented evidence that finally explains that (a) bilexical statistics are actually getting used with great frequency in the parse theories that will ultimately have the highest score, but (b) the distributions involving bilexical statistics are so similar to their back-off counterparts as to make them nearly indistinguishable insofar as making different parse decisions. Finally, our analysis has provided for the first time an effective way to do parameter selection with a generative lexicalized statistical parsing model.

Of course, there is still much more analysis, hypothesizing, testing and extrapolation to be done. A thorough study of the highest-entropy distributions should reveal new ways in which to use grammar transforms or develop features to reduce the entropy and increase parse accuracy. A closer look at the low-entropy distributions may reveal additional reductions in the size of the model, and, perhaps, a way to incorporate hard constraints without disturbing the more ambiguous parts of the model more suited to machine learning than human engineering.

9 Acknowledgements

Thanks to Mitch Marcus, David Chiang and Julia Hockenmaier for their helpful comments on this work. I would also like to thank Bob Moore for asking some insightful questions that helped prompt this line of research. Thanks also to Fernando Pereira, with whom I had invaluable discussions about distributional similarity. This work was supported in part by DARPA grant N66001-00-1-9815.

References

Daniel M. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of HLT2002*, San Diego, CA.

Daniel M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*. To appear.

E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavens, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Speech and Natural Language Workshop*, pages 306–311, Pacific Grove, California. Morgan Kaufmann Publishers.

Ezra Black, Frederick Jelinek, John Lafferty, David Magerman, Robert Mercer, and Salim Roukos. 1992a. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 5th DARPA Speech and Natural Language Workshop*, Harriman, New York.

Ezra Black, John Lafferty, and Salim Roukos. 1992b. Development and evaluation of a broad-coverage probabilistic grammar of english-language computer manuals. In *Proceedings of the 30th ACL*, pages 185–192.

Eugene Charniak. 2000. A maximum entropy–inspired parser. In *Proceedings of the 1st NAACL*, pages 132–139, Seattle, Washington, April 29 to May 4.

Michael John Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *International Conference on Machine Learning*.

Thomas Cover and Joy A. Thomas. 1991. *Elements of Information Theory*. John Wiley & Sons, Inc., New York.

Jason Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 340–345, Copenhagen, August.

Jason Eisner. 2000. Bilexical grammars and their cubic-time parsing algorithms. In Harry Bunt and Anton Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*, pages 29–62. Kluwer Academic Publishers, October.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, Pittsburgh, Pennsylvania.

Dan Klein and Christopher D. Manning. 2002. Conditional structure versus conditional estimation in NLP models. In *Proceedings of the 2002 Conference on Empirical Methods for Natural Language Processing*.

John Lafferty, Fernando Pereira, and Andrew McCallum. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Lillian Lee. 1999. Measures of distributional similarity. In *Proceedings of the 37th ACL*, pages 25–32.

Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

David Magerman. 1994. *Natural Language Parsing as Statistical Pattern Recognition*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Pennsylvania.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Noam Slonim, Nir Friedman, and Naftali Tishby. 2002. Unsupervised document classification using sequential information maximization. Technical Report 2002–19, Leibniz Center, The School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel.