Feature-Rich Log-Linear Lexical Model for Latent Variable PCFG Grammars

Zhongqiang Huang and **Mary Harper**
Department of Computer Science
University of Maryland, College Park
{zqhuang,mharper}@umd.edu

Abstract

Context-free grammars with latent annotations (PCFG-LA) have been found to be effective for parsing many languages; however, currently their lexical model may be subject to over-fitting and requires language engineering to handle out-of-vocabulary (OOV) words. Inspired by previous studies that have incorporated rich features into generative models, we propose to use a feature-rich log-linear lexical model to train PCFG-LA grammars that are more robust to rare and OOV words. The proposed lexical model has three advantages: over-fitting is alleviated via regularization, OOV words are modeled using rich features, and lexical features are exploited for grammar induction. Our approach results in significantly more accurate PCFG-LA grammars that are flexible to train for different languages (with test F scores of 90.5, 85.0, and 81.9 on WSJ, CTB6, and ATB, respectively).

1 Introduction

The latent variable approach of (Matsuzaki et al., 2005; Petrov et al., 2006) is capable of learning high accuracy context-free grammars directly from a raw treebank, and has achieved state-of-the-art parsing accuracies on multiple languages, outperforming many other parsers that are engineered for performance in a particular language (Petrov, 2009; Green and Manning, 2010). However, the lexical model of PCFG-LA grammars (responsible for emitting words from latent POS tags) is not designed to effectively handle OOV words universally. In fact, hand-crafted rules designed for English OOV words were used in the multi-language study of (Petrov, 2009) for non-English languages, leaving room for further improvement for each of the languages studied.

Huang and Harper (2009) and Attia et al. (2010) studied the impact of rare and OOV word handling for parsing with PCFG-LA grammars, especially for non-English languages. They both found that language-specific handling of OOV words significantly improves parsing performance. However, hand tailoring of the language-specific module with expert knowledge may produce suboptimal results, and would not be applicable to new languages. Petrov and Klein (2008) presented a discriminatively trained PCFG-LA model that makes use of rich morphological features for handling OOV words and obtained improved performance on some languages; however, this method was considerably less accurate than its strong generative counterpart on English WSJ.

Berg-Kirkpatrick et al. (2010) demonstrated that each generation step of a generative process can be modeled as a locally normalized log-linear model so that rich features can be incorporated for learning unsupervised models, e.g., POS induction. Inspired by their work, we propose a log-linear lexical model for generative PCFG-LA grammars. It maintains the advantages of generative models, while providing a principled way to: 1) alleviate over-fitting via regularization, 2) handle OOV words using rich features, and 3) exploit lexical features for grammar induction. The proposed approach produces significant improvements for all of the three studied languages.

The rest of the paper is structured as follows. We first review PCFG-LA grammars and issues related to the lexical model in Section 2, and then describe the proposed log-linear lexical model and the training methods in Sections 3 and 4, respectively. Experiments are presented in Section 5. Section 6 concludes this paper.

2 PCFG-LA Grammar

PCFG grammars with latent annotations (Matsuzaki et al., 2005; Petrov et al., 2006) augment

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 219–227,
Chiang Mai, Thailand, November 8 – 13, 2011. ©2011 AFNLP

the observed parse trees in the treebank with a latent variable at each tree node. Each latent variable effectively refines an observed category t into a set of latent subcategories $\{t_x | x = 1, \cdots, |t|\}$, where $|t|$ denotes the number of latent tags split from t. Each syntactic category in the original tree in Figure 1(a) is split into multiple latent subcategories, and that parse tree is decomposed into many derivation trees whose non-terminals are latent categories; Figure 1(b) depicts one such derivation tree, where each grammar rule expands a latent non-terminal category into a sequence of latent non-terminals and/or terminal words, e.g., VP-4→VBD-5 NP-6.

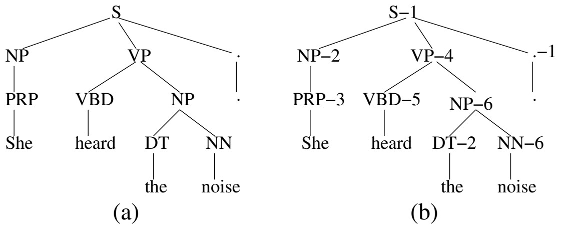

Figure 1: (a) treebank tree (b) derivation tree

The objective of PCFG-LA training is to induce a grammar with latent variables that maximizes the probability of the training trees. Given a PCFG-LA grammar with model parameter θ, $\mathcal{R}$ denotes the set of grammar rules, $\mathcal{D}(T)$ the set of derivation trees for parse tree T, and $\mathcal{R}(T)$ and $\mathcal{R}(D)$ the sets of rules comprising T and D, respectively. The probability of T under the grammar is computed as:

$$\mathrm{P}_\theta(T) = \sum_{D \in \mathcal{D}(T)} \mathrm{P}_\theta(D) = \sum_{D \in \mathcal{D}(T)} \prod_{r \in \mathcal{R}(D)} \mathrm{P}_\theta(r)$$

An EM-algorithm is used to optimize θ based on the training likelihood. The E-step computes the expected count e_r of rule r over the training set $\mathcal{T}$ under the current model parameter θ':

$$e_r \leftarrow \sum_{T \in \mathcal{T}} \sum_{r' \in \mathcal{R}(T)} \delta(r', r) \mathrm{P}_{\theta'}(r' | T) \qquad (1)$$

where $\delta(\cdot, \cdot)$ is an indicator function that returns 1 if the two operands are identical and 0 otherwise, and $\mathrm{P}_{\theta'}(r'|T)$ is the posterior probability of having (latent) rule r' in parse tree T. The M-step aims to maximize the intermediate objective:

$$l(\theta) = \sum_{r \in \mathcal{R}} e_r \log \mathrm{P}_\theta(r) \qquad (2)$$

which results in the following update formula for lexical rule probability $\theta_{t_x \rightarrow w} = \mathrm{P}_\theta(w|t_x)$:

$$\mathrm{P}_\theta(w|t_x) = \frac{e_{t_x,w}}{\sum_{w'} e_{t_x,w'}} \qquad (3)$$

where $e_{t_x,w}$ denotes the expected count of lexical rule $r{=}t_x{\rightarrow}w$. The phrasal rule probabilities are updated similarly.

In order to allocate the grammar complexity to where it is most needed, Petrov et al. (2006) developed a simple split-and-merge procedure. In every split-merge (SM) round, each latent category is first split into two, and the model is re-estimated using several rounds of EM iterations. A likelihood criterion is then used to merge back the least useful splits. The result is that categories, such as NP and VB, that occur frequently in different syntactic environments, are split more heavily than categories such as UH (interjection). This approach also creates a hierarchy of latent categories that enables efficient coarse-to-fine parsing (Petrov and Klein, 2007).

We next discuss two important issues related to the lexical model of PCFG-LA grammars: over-fitting and OOV word handling.

2.1 Over-fitting

As the number of latent annotations increases, a PCFG-LA grammar has an increasing power to fit the training data through EM training, leading to over-fitting. In order to counteract this behavior, Petrov et al. (2006) introduced a linear smoothing method to smooth lexical emission probabilities:

$$\begin{aligned} \bar{\mathrm{P}} &= \frac{1}{|t|} \sum_x \mathrm{P}_\theta(w|t_x) \\ \mathrm{P}_\theta(w|t_x) &\leftarrow \epsilon \bar{\mathrm{P}} + (1-\epsilon) \mathrm{P}_\theta(w|t_x) \end{aligned}$$

A similar smoothing method was used for phrasal rules.

While the above method has been found to be effective, Huang and Harper (2009) observed that rare words suffer more from over-fitting than frequent words and suggested tying rare words together when estimating their emission probabilities. Using their approach, all words with a frequency less than a threshold τ are mapped to symbol *rare*[1], and their emission probability $\mathrm{P}_\theta(w|t_x)$ is set in proportion to their co-occurrences with the surface POS tag:

$$\mathrm{P}_\theta(w|t_x) = \frac{c_{t,w}}{\sum_{w': c_{\cdot,w'} < \tau} c_{t,w'}} \mathrm{P}_\theta(\mathit{rare}|t_x)$$

[1] τ is tuned on the development set.

where $c_{\cdot,w}$ and $c_{t,w}$ are the observed counts of words and word/tag pairs, respectively, and $\mathrm{P}_\theta(rare|t_x)$ is a free parameter estimated by the EM algorithm. This constraint greatly reduces the number of free parameters and was found to significantly improve parsing accuracies.

2.2 OOV Handling

Since the lexical model can only generate words observed in the training data, a separate module is needed to handle OOV words that can appear in novel test sentences. A simple approach might be to estimate the emission probability of an OOV word w based on how likely it is that t_x generates a rare word in the training data:

$$\mathrm{P}_\theta(w|t_x) = \mathrm{P}_\theta(rare|t_x)$$

We call this type of approach the *simple* method[2].

A better approach would exploit the word formation process for the language being modeled. As with other generative English parsers, the PCFG-LA parser implementation of (Petrov et al., 2006) classifies OOV words into a set of OOV signatures based on the presence of features such as capital letters, digits, dashes, as well as a list of indicative suffixes (e.g., *-ing*, *-ion*, *-er*), and estimates the emission probability of an OOV word w given a latent tag t_x as:

$$\mathrm{P}_\theta(w|t_x) \propto \mathrm{P}_\theta(s|t_x)$$

where s is the OOV signature for w and $\mathrm{P}_\theta(s|t_x)$ is computed by $e_{t_x,s}/e_{t_x,\cdot}$.

While this approach performs well for English, the same OOV word handling module would not be adequate for other languages since they have different word formation processes, which should be exploited for better disambiguation of OOV words. For example, Huang and Harper (2009) improved Chinese parsing performance by estimating the emission probability of an OOV word using the geometric average of the emission probabilities of all of the characters ch_k in the word:

$$\mathrm{P}_\theta(w|t_x) = \sqrt[n]{\prod_{ch_k \in w, \mathrm{P}_\theta(ch_k|t_x) \neq 0} \mathrm{P}_\theta(ch_k|t_x)}$$

where $n = |\{ch_k \in w | \mathrm{P}_\theta(ch_k|t_x) \neq 0\}|$. As will be shown later in Section 5, handling Arabic OOV words in a similar way to Chinese produces improved parsing performance on Arabic[3]; however, the aforementioned language dependent OOV handling approaches are most likely suboptimal and designing a method for a new language could be nontrivial. We call this type of approach the *heuristic* method.

Researchers have exploited discriminative parsing models (Finkel et al., 2008; Petrov and Klein, 2008) to utilize naturally occurring overlapping features, including features for OOV handling. The discriminative version of the PCFG-LA grammar (Petrov and Klein, 2008) was found to be more accurate than its generative counterpart on some languages, partially due to its use of regularization and multi-scale grammars to alleviate data sparsity and rich features to improve OOV word handling. However, such a model is much slower to train and considerably less accurate on English WSJ than its strong generative counterpart. Hence, we will investigate a locally normalized log-linear lexical model to take advantage of rich features within the generative learning framework.

3 Log-Linear Lexical Model for PCFG-LA grammars

Instead of treating each $\mathrm{P}_\theta(w|t_x)$ as a free parameter of a multinomial distribution as in a standard PCFG-LA grammar, we first model the conditional probability of latent tag t_x given the surface POS tag t and word w using a log-linear model:

$$\mathrm{P}_\phi(t_x|t,w) = \frac{\exp\langle\phi, \mathbf{f}(t_x,w)\rangle}{\sum_{x'} \exp\langle\phi, \mathbf{f}(t_{x'},w)\rangle} \quad (4)$$

where $\mathbf{f}(t_x,w)$ represents the feature vector extracted from the pair (t_x,w), ϕ is the feature weight vector, and the denominator sums over all latent tags for POS tag t. This model is applicable to both known and OOV words as long as there are active features; otherwise, a uniform latent tag distribution would be assumed. We call this the *latent lexical model* as it deals with the distribution of latent tags.

The conditional probability of t_x given word w can then be expressed as:

$$\mathrm{P}_\theta(t_x|w) = \mathrm{P}_\theta(t_x,t|w) = \mathrm{P}_\phi(t_x|t,w)\mathrm{P}(t|w)$$

and finally the word emission probability given a latent tag can be computed via Bayes' rule:

$$\mathrm{P}_\theta(w|t_x) = \frac{\mathrm{P}_\phi(t_x|t,w)\mathrm{P}(t|w)\mathrm{P}(w)}{\sum_{w'} \mathrm{P}_\phi(t_x|t,w')\mathrm{P}(t|w')\mathrm{P}(w')} \quad (5)$$

[2] This method is used in the simple lexicon of the Berkeley parser.

[3] We use prefixes and suffixes up to three characters for handling Arabic OOV words.

This new lexical model is composed of the latent lexical model $\mathrm{P}_\phi(t_x|t,w)$ and two other parts: $\mathrm{P}(t|w)$ and $\mathrm{P}(w)$, which are computed differently for known and OOV words.

For words observed in the training data, both $\mathrm{P}(t|w)$ are $\mathrm{P}(w)$ are computed using the maximum-likelihood estimation (based on the observed training trees) so that $\mathrm{P}_\theta(w|t_x)$ forms a proper distribution of observed words during grammar induction.

For OOV words, we use a log-linear *OOV model* to estimate the POS tag distribution:

$$\mathrm{P}_\gamma(t|w) = \frac{\exp\langle\gamma, \mathbf{g}(t,w)\rangle}{\sum_{t'}\exp\langle\gamma, \mathbf{g}(t',w)\rangle} \tag{6}$$

where $\mathbf{g}(t,w)$ represents the feature vector extracted from the pair (t,w), γ is the feature weight vector, and the denominator sums over all POS tags with active features. The *simple* approach in Subsection 2.2 is used when no feature is active. $\mathrm{P}(w)$ is approximated by one over the number of training tokens. It should be noted that $\mathrm{P}_\gamma(t|w)$ may use different features than $\mathrm{P}_\phi(t_x|t,w)$.

Compared with modeling $\mathrm{P}_\theta(w|t_x)$ directly as a multinomial distribution, the new lexical model separates $\mathrm{P}(t|w)$ from $\mathrm{P}_\phi(t_x|t,w)$, offering three important advantages:

- The parameter ϕ of the latent lexical model $\mathrm{P}_\phi(t_x|t,w)$ can be smoothed through regularization to address data sparsity.
- Rich features can be utilized in the OOV model $\mathrm{P}_\gamma(t|w)$ to estimate POS tag distributions of OOV words for a variety of languages. This is important when working on new languages.
- Rich features can be utilized in the latent lexical model $\mathrm{P}_\phi(t_x|t,w)$ to guide the induction of latent POS tags.

The reader should note that Berg-Kirkpatrick et al. (2010) modeled $\mathrm{P}_\theta(w|t_x)$ directly using a log-linear model:

$$\mathrm{P}_\phi(w|t_x) = \frac{\exp\langle\phi, \mathbf{f}(t_x,w)\rangle}{\sum_{w'}\exp\langle\phi, \mathbf{f}(t_x,w')\rangle}$$

This would be problematic for our parsing model because it would not be trained to estimate the probability of OOV words given a latent tag. For parsing, we must model OOV words that can appear in previously unseen sentences. One might compute the numerator for an OOV word based on its features and divide it by a denominator approximated using the words in the training data, but such an estimate is inaccurate and results in poor performance in our preliminary experiments.

We also choose not to model $\mathrm{P}_\theta(t_x|w)$ directly using a log-linear model:

$$\mathrm{P}_\phi(t_x|w) = \frac{\exp\langle\phi, \mathbf{f}(t_x,w)\rangle}{\sum_{t'}\sum_{x'}\exp\langle\phi, \mathbf{f}(t'_{x'},w)\rangle}$$

and compute $\mathrm{P}_\theta(w|t_x)$ via Bayes' rule. Such a model cannot guarantee that the probability $\mathrm{P}_\theta(t|w)$ computed by $\sum_x \mathrm{P}_\theta(t_x|w)$ is equal to the maximum likelihood estimate, which is a reasonable constraint.

4 Model Training

The parameter θ for our parser model consists of ϕ for the log-linear latent lexical model, γ for the log-linear OOV model, and ψ for the phrasal rule expansion probabilities. The other parameters (e.g., $\mathrm{P}(t|w)$ and $\mathrm{P}(w)$ for known words and $\mathrm{P}(\mathit{rare}|t_x)$) can be computed based on observable or fractional counts once θ is determined.

γ of the OOV model is independent of the latent categories, and we simply use a gradient-based optimization approach to maximize the following objective:

$$l'(\gamma) = \sum_{t,w} c_{t,w} \log \mathrm{P}_\gamma(t|w) - \kappa'||\gamma||^2$$

where $c_{t,w}$ is the count of the pair (t,w) in the training data, and κ' is the regularization weight.

For parameters ψ and ϕ, we follow the split-merge training procedure in (Petrov et al., 2006) to induce latent categories. Given a set of latent categories, the goal is to find θ that maximizes the regularized training likelihood:

$$L(\theta) = \sum_{T\in\mathcal{T}} \log \mathrm{P}_\theta(T) - \kappa||\phi||^2 \tag{7}$$

where $\kappa||\phi||^2$ is the regularization term[4] for the feature weights of the latent lexical model.

The two optimization approaches described in (Berg-Kirkpatrick et al., 2010) can be extended naturally to our problem. One approach is EM-based with an E-step identical to Equation 1 in

[4] Both κ' and κ are tuned on the development set. We could also use L1 regularization and leave it to future work.

Section 2. The objective of the M-step becomes:

$$\begin{aligned} l(\theta) &= \sum_{w \rightarrow t_x \in \mathcal{R}_l} e_{t_x,w} \log \mathrm{P}_\phi(w|t_x) - \kappa||\phi||^2 \\ &\quad + \sum_{r \in \mathcal{R}_p} e_r \log \mathrm{P}_\psi(r) \end{aligned}$$

where we separate the set of rules $\mathcal{R}$ into lexical rules $\mathcal{R}_l$ and phrasal rules $\mathcal{R}_p$. The phrasal rule parameter ψ is updated as before by normalizing the expected rule counts and is smoothed in the same way as in (Petrov et al., 2006). The intermediate objective function $l(\phi)$ related to ϕ, i.e.,

$$l(\phi) = \sum_{w \rightarrow t_x \in \mathcal{R}_l} e_{t_x,w} \log \mathrm{P}_\phi(w|t_x) - \kappa||\phi||^2$$

can be optimized by a gradient descent optimization algorithm (we use LBFGS (Liu and Nocedal, 1989)). Its gradient has the following form:

$$\begin{aligned} \nabla l(\phi) &= \sum_{w \rightarrow t_x \in \mathcal{R}_l} e^*_{t_x,w} \Delta_{t_x,w}(\phi) - 2\kappa \cdot \phi \\ \Delta_{t_x,w}(\phi) &= \mathbf{f}(t_x, w) - \sum_{x'} \mathrm{P}_\phi(t_{x'}|w,t)\mathbf{f}(t_{x'}, w) \end{aligned}$$

where $e^*_{t_x,w} = e_{t_x,w} - e_{t_x,\cdot}\mathrm{P}_\phi(w|t_x)$.

It can be shown that $l(\phi)$ is not a concave function with respect to ϕ, but this created no problems in our experiments. It should be noted that if we set the regularization weight κ to 0, the maximum of $l(\phi)$ is achieved when $\mathrm{P}_\phi(w|t_x)$ is set to $e_{t_x,w}/e_{t_x,\cdot}$, which is identical to the update formula in Equation 3, and would thus be unable to use rich features. This is less of an issue when regularization takes effect as it favors common discriminative features to reduce the penalty term.

The second approach, which was found to outperform the EM-based approach in (Berg-Kirkpatrick et al., 2010), optimizes on the regularized log-likelihood (Equation 7) directly by updating both ψ and ϕ using a gradient descent approach. In order to convert this to an unconstrained optimization problem[5], we set each phrasal rule expansion probability ψ_i as the output of a log-linear model, i.e., $\psi_i = \exp(\psi'_i)/Z$ with Z being the normalization factor, and treat ψ' as the parameter for the phrasal rules to be optimized. The gradient of $L(\theta)$ with respect to ϕ turns out to be the same as in the first approach (Salakhutdinov et al., 2003). The gradient of $L(\theta)$ with respect to ψ' can be derived similarly. We omit the details here due to space limitations.

In the original EM-based training approach (Petrov et al., 2006), many of the rule expansion probabilities become very small and are pruned to dramatically reduce the grammar size. The phrasal rule probabilities computed from the log-linear model with parameter ψ' are not usually low enough to be pruned, due to the fact that a large decrease in ψ'_i results in a much smaller change in ψ_i when ψ_i is already relatively small. In order to address this problem, we combine the two optimization approaches together: first run rounds of EM-based optimization to initialize the grammar parameters and prune many of the useless phrasal rules, and then switch to the direct gradient descent optimization approach. This combined approach outperforms the standalone EM-based approach in our study and is used in the experiments reported in this paper.

5 Experiments

In this section, we will show the effect of rare word smoothing and OOV handling on the accuracy of the standard PCFG-LA grammars, and investigate how the proposed feature-rich lexical model addresses these problems. In what follows, we first describe the experimental data and then the results of the standard PCFG-LA grammars. We then describe the features and results of the PCFG-LA grammars with log-linear lexical models, and present some analyses. Finally, additional features are discussed and the final test results are compared with the literature.

5.1 Data & Setup

We experiment with three languages: English, Chinese, and Arabic. For English, we used the WSJ Penn Treebank (Marcus et al., 1999) and the commonly used data splits (Charniak, 2000). For Chinese, we used the Penn Chinese Treebank 6.0 (CTB6) (Xue et al., 2005) and the preparation steps and data splits in (Huang and Harper, 2009). For Arabic, we used the Penn Arabic Treebank (ATB) (Maamouri et al., 2009) and the preparation steps[6] and data splits in (Green and Manning, 2010; Chiang et al., 2006). Table 1 provides gross statistics for each treebank. As we can see, CTB6 and ATB both have a higher OOV rate than WSJ,

[5] The elements of ψ are constrained to form proper probability distributions.

[6] Except that clitic marks were removed, which results in about 0.3 degradation in F score (p.c.).

and hence have greater need for effective OOV handling.

	Statistics	Train	Dev	Test
English (WSJ)	#sents	39832	1700	2416
	#tokens	950.0k	40.1k	56.7k
	%oov_types	-	12.8%	13.2%
	%oov_tokens	-	2.8%	2.5%
Chinese (CTB6)	#sents	24416	1904	1975
	#tokens	678.8k	51.2k	52.9k
	%oov_types	-	20.6%	20.9%
	%oov_tokens	-	5.0%	5.3%
Arabic (ATB)	#sents	18818	2318	2313
	#tokens	597.9k	70.7k	70.1k
	%oov_types	-	15.6%	16.7%
	%oov_tokens	-	3.2%	3.4%

Table 1: Gross Statistics of the treebanks.

Due to the variability (caused by random initialization) among the grammars (Petrov, 2010), we train 10 grammars with different seeds in each experiment and report their average F score on the development set. The best grammar selected using the development set is used for evaluation on the test set.

5.2 Standard PCFG-LA Grammars

We first study the effect of rare word smoothing and OOV handling on the standard PCFG-LA grammars using our reimplementation of the Berkeley parser. The *no+simple* row in Table 2 represents the baseline, for which the grammars are trained without rare word smoothing described in Subsection 2.1 and OOV words are handled by the simple method described in Subsection 2.2. Each language-dependent heuristic-based OOV word handling method improves parsing accuracies, and the rare word smoothing method provides even greater improvement across the languages. Their combination results in further improvement. This confirms that both over-fitting and OOV words are issues to consider for training accurate PCFG-LA grammars.

Rare Word Smoothing	OOV	WSJ	CTB6	ATB
no	simple	89.86	82.52	79.12
no	heuristic	90.07	82.98	79.44
yes	simple	90.53	83.25	80.30
yes	heuristic	90.69	83.73	80.64

Table 2: The effect of rare word smoothing and OOV handling on parsing F scores evaluated on the respective development set.

5.3 Log-Linear Lexical Model

Here we investigate a core set of features that have proven effective for POS tagging to demonstrate the effectiveness of our model and its robustness across languages, and leave it to future work to include additional features as discussed in Subsection 5.5. Table 3 lists the templates we used to extract predicates on words. For the log-linear OOV model, we use the *full* feature set, i.e., (t, pred) pairs extracted using all of the predicates. For the log-linear latent lexical model, we experiment with two feature sets: 1) the *wid* feature set containing only (t_x, wid) pairs, which are the same as those used in the standard PCFG-LA grammars, 2) the *full* feature set using all of the predicates.

Predicate	Explanation
$\delta(w = \cdot)$	word identity (wid)
$\delta(\text{hasDigit}(w) = \cdot)$	contains a digit?
$\delta(\text{hasHyphen}(w) = \cdot)$	contains a hyphen?
$\delta(\text{initCap}(w) = \cdot)$	first letter capitalized?
$\delta(\text{prefix}_k(w) = \cdot)$	prefix of length $k \leq 3$
$\delta(\text{suffix}_k(w) = \cdot)$	suffix of length $k \leq 3$

Table 3: Predicate templates on word w.

We first evaluate the effectiveness of regularization and the log-linear OOV model by training the latent lexical model using the *wid* feature set with regularization and examining different OOV handling methods. As shown in Table 4, the *wid+simple* and *wid+heuristic* approaches[7] produce results comparable to the corresponding PCFG-LA grammars trained with rare word smoothing and respective OOV handling. This shows that regularizing the latent lexical model alleviates data sparsity, however, we will illustrate in Subsection 5.4 that this is achieved in a different way than rare word smoothing.

The log-linear OOV model using the *full* feature set results in improved parsing performance over all languages, with the most improvement seen on Arabic (0.71 F), followed by Chinese (0.28 F), confirming that the log-linear OOV model is more accurate than the heuristic approach, and can be flexibly used for different languages. The improvement on English is marginal possibly because the signature-based OOV features are sufficiently accurate for handling English unknown

[7] Training the latent lexical model using the *wid* feature set and handling OOV words using the *simple* or *heuristic* approach.

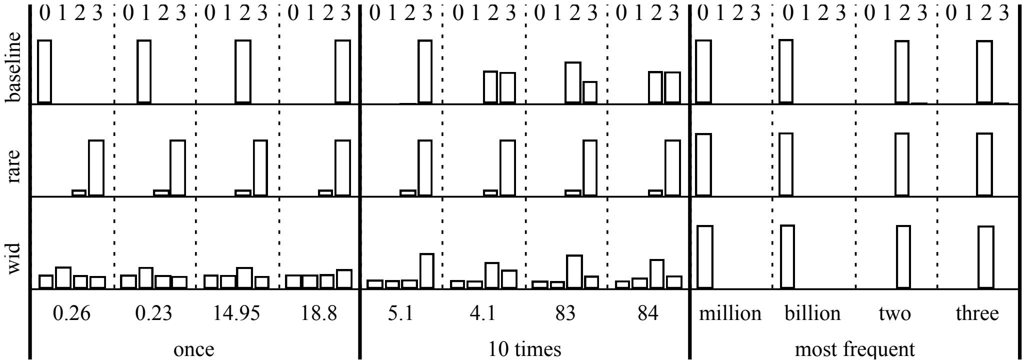

Figure 2: The conditional distribution P$(t_x|t, w)$ of latent tags for selected cardinal numbers (e.g., 0.26, million) that appear only once, 10 times, or frequently for standard PCFG-LA grammars trained with (labeled rare) or without (labeled baseline) rare word smoothing and for PCFG-LA grammars with regularized feature-rich lexical model using the *wid* feature set (labeled wid). The distribution is represented by the four bars separated by dotted vertical lines, and each bar represents the conditional probability of a latent tag.

words after years of expert crafting.

We next investigate the effect of training the latent lexical model using the *full* feature set. Compared with the *wid+full* model, the *full+full* model improves 0.38 F on Arabic and 0.27 F on Chinese, despite the fact that the additional features are very simple, mostly prefixes and suffixes of words. The improvement on English is again marginal possibly because the features do not provide such insights on fine-grained syntactic subcategories (e.g., suffix *-ed* is indicative of past tense verbs, but not their sub-categories). Admittedly, many of the features are noisy, but as we will show in Subsection 5.4, some of the features can guide the learning of the latent categories to reflect the similarity between syntactically similar words of the same POS type.

Compared with the baseline (*no+simple* in Table 2), the feature-rich *full+full* model significantly improves parsing F scores by 1.03, 1.66, and 2.67 absolute on English, Chinese, and Arabic, respectively.

Latent Lexical	OOV	WSJ	CTB6	ATB
wid	simple	90.54	83.18	80.32
wid	heuristic	90.71	83.63	80.70
wid	full	90.81	83.91	81.41
full	full	90.89	84.18	81.79

Table 4: The effect of features (wid vs. full) for training the latent lexical model and the OOV handling methods (simple, heuristic, or the log-linear model using the full feature set) on parsing performance (F score) on the development set.

5.4 Analysis

We examine in Figure 2 the effect of regularization and rare word smoothing on the learned rules by looking at the distribution P$(t_x|t, w)$ for PCFG-LA grammars trained in different ways[8]. For standard PCFG-LA grammars trained without rare word smoothing (labeled baseline), rare words have sparse distributions of latent tags, which are determined solely based on limited contexts and are thus not reliable. The rare word smoothing approach (labeled rare) collapses all rare words into a single token so that P$(t_x|t, w)$ = P$(t_x|t, rare)$ is identical for any rare word w. This constraint greatly reduces data sparsity; however, treating all rare words as one token could eliminate too much lexical information (e.g., the distribution of latent tags is the same for all rare cardinal numbers no matter whether they appear only once or 10 times). Regularization of the log-linear latent lexical model (labeled wid) favors a uniform distribution (zero penalty when all feature weights are zero). There is not much evidence to skew the distribution from uniform for rare words. However, when more evidence is available, the distribution becomes smoothly skewed to reflect the different syntactic preferences of the individual words, and it can eventually become as spiky as in the other approaches given sufficient evidence.

In order to provide some insights into why parsing accuracies are improved for Arabic and Chinese by using the *full* feature set when training the latent lexical model, we look at the country names

[8] For standard PCFG-LA grammars, P$(t_x|t, w)$ is simply computed by $e_{t_x,w}/e_{t,w}$; whereas, for the feature-rich lexical model, P$(t_x|t, w)$ is computed from the latent lexical model.

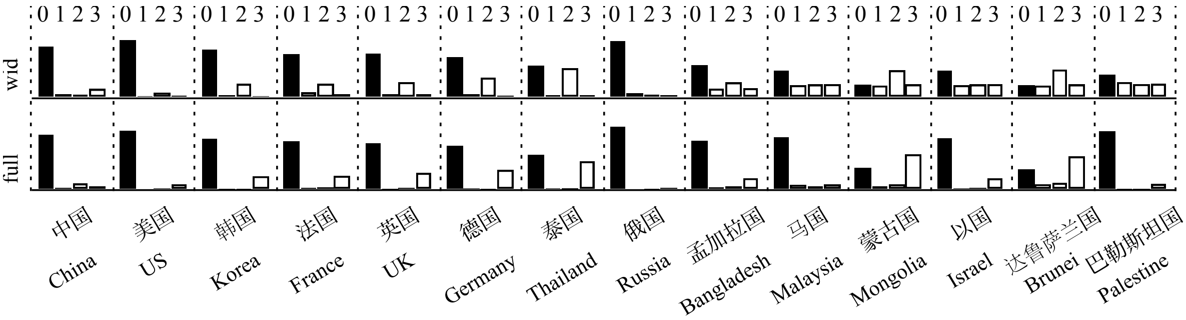

Figure 3: The conditional distribution $\mathrm{P}(t_x|t, w)$ of latent tags for selected country names (proper nouns) listed in order of decreasing frequency from the Chinese treebank (English translations are provided under Chinese names), after training using the *wid* and the *full* feature set, respectively. The distribution is represented by the four bars separated by dotted vertical lines, and each bar represents the conditional probability of a latent tag. The preferred latent tag for country names is highlighted in black.

that end with the character 国 (country) in the Chinese treebank. These names appear in similar contexts and would be expected to favor certain latent tag or tags; however, when training using the *wid* feature set, this is only true for the frequent names as shown in Figure 3. For the rare names, there is not much evidence to divert the distribution away from uniform. When training with the *full* feature set, the suffix1=国 predicate is active for all of those country names and has a large feature weight associated with the preferred latent tag. As a result, the distribution of latent tags for the rare names is skewed more toward the preferred latent tag due to strong evidence from that suffix feature.

5.5 Other Features

Our model supports any local features that can be extracted from the pair (t_x, w), including the language-dependent features studied in (Attia et al., 2010). In addition, features related to word semantics (e.g., using WordNet (Fellbaum, 1998)) or word clusters (e.g., using unsupervised clustering (Brown et al., 1992; Koo et al., 2008; Goyal and Daume, 2011)) might also be beneficial for modeling $\mathrm{P}_\phi(t_x|t, w)$ and/or $\mathrm{P}_\gamma(t|w)$. Features extracted from (t, w) could also be helpful for providing some smoothing effect across the latent tags. Moreover, it might be beneficial to perform feature selection prior to training. We leave this to future work.

5.6 Final Results

Table 5 compares the final test results of our best grammars (the *full+full* approach) with the literature[9]. Our PCFG-LA grammars with a feature-rich lexical model significantly outperform the standard PCFG-LA grammars of (Petrov and Klein, 2007) for all of the three languages, especially on Chinese (+1.6 F) and Arabic (+2.2 F).

TB	Parser	LP	LR	F
WSJ	Charniak (2000)	89.9	89.5	89.7
	Petrov and Klein (2007)	90.2	90.1	90.1
	Petrov and Klein (2008)	-	-	89.4
	Huang and Harper (2009)	90.4	89.9	90.1
	This Paper	**90.8**	**90.3**	**90.5**
CTB6	Charniak (2000)	80.5	79.5	80.0
	Petrov and Klein (2007)	84.0	82.9	83.4
	Huang and Harper (2009)	85.1	83.2	84.1
	This Paper	**85.9**	**84.2**	**85.0**
ATB	Petrov and Klein (2007)	80.5	78.9	79.7
	This Paper	**82.7**	**81.2**	**81.9**

Table 5: Final test set accuracies.

6 Conclusions

We have presented a feature-rich lexical model for PCFG-LA grammars to: 1) alleviate over-fitting via regularization, 2) handle OOV words using rich features, and 3) exploit lexical features for grammar induction. Experiments show that the proposed approach allows us to train more effective PCFG-LA grammars for more accurate and robust parsing of three different languages. It is expected that even more accurate parsers can be produced by using this approach together with self-training (Huang and Harper, 2009) and/or product models (Petrov, 2010; Huang et al., 2010).

Acknowledgments

This research was supported in part by NSF IIS-0703859. We would like to thank Spence Green for providing the processed Arabic Treebank data and lots of insightful suggestions.

[9] All of the parsers from the referenced papers are trained and evaluated using the data splits in our experiments.

References

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In *Proceedings of the North American Chapter of the Association for Computational Linguistics conference*.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic dialects. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Jenny Rose Finkel, Alex Kleeman, and Christopher D. Manning. 2008. Efficient, feature-based, conditional random field parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Amit Goyal and Hal Daume. 2011. Approximate scalable bounded space sketch for large data NLP. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the International Conference on Computational Linguistics*.

Zhongqiang Huang and Mary Harper. 2009. Self-training PCFG grammars with latent annotations across languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Zhongqiang Huang, Mary Harper, and Slav Petrov. 2010. Self-training with products of latent variable. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Terry Koo, Xavier Carrera, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*.

Mohamed Maamouri, Ann Bies, Sondos Krouna, Fatma Gaddeche, and Basma Bouziri. 2009. Penn Arabic treebank guidelines. Technical report, Linguistic Data Consortium, University of Pennsylvania.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor, 1999. *Treebank-3*. Linguistic Data Consortium, Philadelphia.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

Slav Petrov and Dan Klein. 2008. Sparse multi-scale grammars for discriminative latent variable parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Slav Petrov. 2009. *Coarse-to-fine natural language processing*. Ph.D. thesis, University of California at Bekeley.

Slav Petrov. 2010. Products of random latent variable grammars. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*.

Ruslan Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. 2003. Optimization with EM and expectation-conjugate-gradient. In *Proceedings of the International Conference on Machine Learning*.

Nianwen Xue, Fei Xia, Fu-dong Chiou, and Marta Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*.