Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure

Mark Johnson
Brown University
Mark_Johnson@Brown.edu

Abstract

Adaptor grammars (Johnson et al., 2007b) are a non-parametric Bayesian extension of Probabilistic Context-Free Grammars (PCFGs) which in effect learn the probabilities of entire subtrees. In practice, this means that an adaptor grammar learns the structures useful for generating the training data as well as their probabilities. We present several different adaptor grammars that learn to segment phonemic input into words by modeling different linguistic properties of the input. One of the advantages of a grammar-based framework is that it is easy to combine grammars, and we use this ability to compare models that capture different kinds of linguistic structure. We show that incorporating both unsupervised syllabification and collocation-finding into the adaptor grammar significantly improves unsupervised word-segmentation accuracy over that achieved by adaptor grammars that model only one of these linguistic phenomena.

1 Introduction

How humans acquire language is arguably the central issue in the scientific study of language. Human language is richly structured, but it is still hotly debated as to whether this structure can be learnt, or whether it must be innately specified. Computational linguistics can contribute to this debate by identifying which aspects of language can potentially be learnt from the input available to a child. Here we try to identify linguistic properties that convey information useful for learning to segment streams of phonemes into words. We show that simultaneously learning syllable structure and collocations improves word segmentation accuracy compared to models that learn these independently. This suggests that there might be a synergistic interaction in learning several aspects of linguistic structure simultaneously, as compared to learning each kind of linguistic structure independently.

Because learning collocations and word-initial syllable onset clusters requires the learner to be able to identify word boundaries, it might seem that we face a chicken-and-egg problem here. One of the important properties of the adaptor grammar inference procedure is that it gives us a way of learning these interacting linguistic structures simultaneously.

Adaptor grammars are also interesting because they can be viewed as directly inferring linguistic structure. Most well-known machine-learning and statistical inference procedures are parameter estimation procedures, i.e., the procedure is designed to find the values of a finite vector of parameters. Standard methods for learning linguistic structure typically try to reduce structure learning to parameter estimation, say, by using an iterative generate-and-prune procedure in which each iteration consists of a rule generation step that proposes new rules according to some scheme, a parameter estimation step that estimates the utility of these rules, and pruning step that removes low utility rules. For example, the Bayesian unsupervised PCFG estimation procedure devised by Stolcke (1994) uses a model-merging procedure to propose new sets of PCFG rules and a Bayesian version of the EM procedure to estimate their weights.

Proceedings of ACL-08: HLT, pages 398–406,
Columbus, Ohio, USA, June 2008.

Recently, methods have been developed in the statistical community for Bayesian inference of increasingly sophisticated non-parametric models. ("Non-parametric" here means that the models are not characterized by a finite vector of parameters, so the complexity of the model can vary depending on the data it describes). Adaptor grammars are a framework for specifying a wide range of such models for grammatical inference. They can be viewed as a nonparametric extension of PCFGs.

Informally, there seem to be at least two natural ways to construct non-parametric extensions of a PCFG. First, we can construct an infinite number of more specialized PCFGs by splitting or refining the PCFG's nonterminals into increasingly finer states; this leads to the iPCFG or "infinite PCFG" (Liang et al., 2007). Second, we can generalize over arbitrary subtrees rather than local trees in much the way done in DOP or tree substitution grammar (Bod, 1998; Joshi, 2003), which leads to adaptor grammars.

Informally, the units of generalization of adaptor grammars are entire subtrees, rather than just local trees, as in PCFGs. Just as in tree substitution grammars, each of these subtrees behaves as a new context-free rule that expands the subtree's root node to its leaves, but unlike a tree substitution grammar, in which the subtrees are specified in advance, in an adaptor grammar the subtrees, as well as their probabilities, are learnt from the training data. In order to make parsing and inference tractable we require the leaves of these subtrees to be terminals, as explained in section 2. Thus adaptor grammars are simple models of structure learning, where the subtrees that constitute the units of generalization are in effect new context-free rules learnt during the inference process. (In fact, the inference procedure for adaptor grammars described in Johnson et al. (2007b) relies on a PCFG approximation that contains a rule for each subtree generalization in the adaptor grammar).

This paper applies adaptor grammars to word segmentation and morphological acquisition. Linguistically, these exhibit considerable cross-linguistic variation, and so are likely to be learned by human learners. It's also plausible that semantics and contextual information is less important for their acquisition than, say, syntax.

2 From PCFGs to Adaptor Grammars

This section introduces adaptor grammars as an extension of PCFGs; for a more detailed exposition see Johnson et al. (2007b). Formally, an adaptor grammar is a PCFG in which a subset M of the nonterminals are *adapted*. An adaptor grammar generates the same set of trees as the CFG with the same rules, but instead of defining a fixed probability distribution over these trees as a PCFG does, it defines a distribution over distributions over trees. An adaptor grammar can be viewed as a kind of PCFG in which each subtree of each adapted nonterminal $A \in M$ is a potential rule, with its own probability, so an adaptor grammar is nonparametric if there are infinitely many possible adapted subtrees. (An adaptor grammar can thus be viewed as a tree substitution grammar with infinitely many initial trees). But any finite set of sample parses for any finite corpus can only involve a finite number of such subtrees, so the corresponding PCFG approximation only involves a finite number of rules, which permits us to build MCMC samplers for adaptor grammars.

A PCFG can be viewed as a set of recursively-defined mixture distributions G_A over trees, one for each nonterminal and terminal in the grammar. If A is a terminal then G_A is the distribution that puts all of its mass on the unit tree (i.e., tree consisting of a single node) labeled A. If A is a nonterminal then G_A is the distribution over trees with root labeled A that satisfies:

$$G_A = \sum_{A \to B_1 \ldots B_n \in R_A} \theta_{A \to B_1 \ldots B_n} \mathrm{TD}_A(G_{B_1}, \ldots, G_{B_n})$$

where R_A is the set of rules expanding A, $\theta_{A \to B_1, \ldots, B_n}$ is the PCFG "probability" parameter associated with the rule $A \to B_1 \ldots B_n$ and $\mathrm{TD}_A(G_{B_1}, \ldots, G_{B_n})$ is the distribution over trees with root label A satisfying:

$$\mathrm{TD}_A(G_1, \ldots, G_n)\left(\begin{array}{c} A \\ \diagup \quad \diagdown \\ t_1 \quad \ldots \quad t_n \end{array}\right) = \prod_{i=1}^{n} G_i(t_i).$$

That is, $\mathrm{TD}_A(G_1, \ldots, G_n)$ is the distribution over trees whose root node is labeled A and each subtree t_i is generated *independently* from the distribution G_i. This independence assumption is what makes a PCFG "context-free" (i.e., each subtree is independent given its label). Adaptor grammars relax

this independence assumption by in effect learning the probability of the subtrees rooted in a specified subset M of the nonterminals known as the *adapted nonterminals*.

Adaptor grammars achieve this by associating each adapted nonterminal $A \in M$ with a Dirichlet Process (DP). A DP is a function of a *base distribution* H and a *concentration parameter* α, and it returns a distribution over distributions $\text{DP}(\alpha, H)$. There are several different ways to define DPs; one of the most useful is the characterization of the conditional or sampling distribution of a draw from $\text{DP}(\alpha, H)$ in terms of the Polya urn or Chinese Restaurant Process (Teh et al., 2006). The Polya urn initially contains $\alpha H(x)$ balls of color x. We sample a distribution from $\text{DP}(\alpha, H)$ by repeatedly drawing a ball at random from the urn and then returning it plus an additional ball of the same color to the urn.

In an adaptor grammar there is one DP for each adapted nonterminal $A \in M$, whose base distribution H_A is the distribution over trees defined using A's PCFG rules. This DP "adapts" A's PCFG distribution by moving mass from the infrequently to the frequently occuring subtrees. An adaptor grammar associates a distribution G_A that satisfies the following constraints with each nonterminal A:

$$\begin{aligned} G_A &\sim \text{DP}(\alpha_A, H_A) && \text{if } A \in M \\ G_A &= H_A && \text{if } A \notin M \\ H_A &= \sum_{A \to B_1 \ldots B_n \in R_A} \theta_{A \to B_1 \ldots B_n} \text{TD}_A(G_{B_1}, \ldots, G_{B_n}) \end{aligned}$$

Unlike a PCFG, an adaptor grammar does not define a single distribution over trees; rather, each set of draws from the DPs defines a different distribution. In the adaptor grammars used in this paper there is no recursion amongst adapted nonterminals (i.e., an adapted nonterminal never expands to itself); it is currently unknown whether there are tree distributions that satisfy the adaptor grammar constraints for recursive adaptor grammars.

Inference for an adaptor grammar involves finding the rule probabilities $\boldsymbol{\theta}$ and the adapted distributions over trees $\boldsymbol{G}$. We put Dirichlet priors over the rule probabilities, i.e.:

$$\boldsymbol{\theta}_A \sim \text{DIR}(\boldsymbol{\beta}_A)$$

where $\boldsymbol{\theta}_A$ is the vector of probabilities for the rules expanding the nonterminal A and $\boldsymbol{\beta}_A$ are the corresponding Dirichlet parameters.

The applications described below require unsupervised estimation, i.e., the training data consists of terminal strings alone. Johnson et al. (2007b) describe an MCMC procedure for inferring the adapted tree distributions G_A, and Johnson et al. (2007a) describe a Bayesian inference procedure for the PCFG rule parameters $\boldsymbol{\theta}$ using a Metropolis-Hastings MCMC procedure; implementations are available from the author's web site.

Informally, the inference procedure proceeds as follows. We initialize the sampler by randomly assigning each string in the training corpus a random tree generated by the grammar. Then we randomly select a string to resample, and sample a parse of that string with a PCFG approximation to the adaptor grammar. This PCFG contains a production for each adapted subtree in the parses of the other strings in the training corpus. A final accept-reject step corrects for the difference in the probability of the sampled tree under the adaptor grammar and the PCFG approximation.

3 Word segmentation with adaptor grammars

We now turn to linguistic applications of adaptor grammars, specifically, to models of unsupervised word segmentation. We follow previous work in using the Brent corpus consists of 9790 transcribed utterances (33,399 words) of child-directed speech from the Bernstein-Ratner corpus (Bernstein-Ratner, 1987) in the CHILDES database (MacWhinney and Snow, 1985). The utterances have been converted to a phonemic representation using a phonemic dictionary, so that each occurrence of a word has the same phonemic transcription. Utterance boundaries are given in the input to the system; other word boundaries are not. We evaluated the f-score of the recovered word constituents (Goldwater et al., 2006b). Using the adaptor grammar software available on the author's web site, samplers were run for 10,000 epochs (passes through the training data). We scored the parses assigned to the training data at the end of sampling, and for the last two epochs we annealed at temperature 0.5 (i.e., squared the probability) during sampling in or-

	1	10	100	1000
U word	**0.55**	**0.55**	**0.55**	0.53
U morph	**0.46**	**0.46**	0.42	0.36
U syll	**0.52**	0.51	0.49	0.46
C word	0.53	0.64	0.74	**0.76**
C morph	0.56	0.63	**0.73**	0.63
C syll	0.77	0.77	**0.78**	0.74

Table 1: Word segmentation f-score results for all models, as a function of DP concentration parameter α. "U" indicates unigram-based grammars, while "C" indicates collocation-based grammars.

$$\text{Sentence} \rightarrow \text{Word}^+$$
$$\underline{\text{Word}} \rightarrow \text{Phoneme}^+$$

Figure 1: The unigram word adaptor grammar, which uses a unigram model to generate a sequence of words, where each word is a sequence of phonemes. Adapted nonterminals are underlined.

der to concentrate mass on high probability parses. In all experiments below we set $\boldsymbol{\beta} = \mathbf{1}$, which corresponds to a uniform prior on PCFG rule probabilities $\boldsymbol{\theta}$. We tied the Dirichlet Process concentration parameters $\boldsymbol{\alpha}$, and performed runs with $\alpha = 1, 10, 100$ and 1000; apart from this, no attempt was made to optimize the hyperparameters. Table 1 summarizes the word segmentation f-scores for all models described in this paper.

3.1 Unigram word adaptor grammar

Johnson et al. (2007a) presented an adaptor grammar that defines a unigram model of word segmentation and showed that it performs as well as the unigram DP word segmentation model presented by (Goldwater et al., 2006a). The adaptor grammar that encodes a unigram word segmentation model shown in Figure 1.

In this grammar and the grammars below, underlining indicates an adapted nonterminal. Phoneme is a nonterminal that expands to each of the 50 distinct phonemes present in the Brent corpus. This grammar defines a Sentence to consist of a sequence of Words, where a Word consists of a sequence of Phonemes. The category Word is adapted, which means that the grammar learns the words that occur in the training corpus. We present our adaptor grammars using regular expressions for clarity, but since our implementation does not handle regular expressions in rules, in the grammars actually used by the program they are expanded using new non-adapted nonterminals that rewrite in a uniform right-branching manner. That is, the adaptor grammar used by the program is shown in Figure 2.

$$\text{Sentence} \rightarrow \text{Words}$$
$$\text{Words} \rightarrow \text{Word}$$
$$\text{Words} \rightarrow \text{Word Words}$$
$$\underline{\text{Word}} \rightarrow \text{Phonemes}$$
$$\text{Phonemes} \rightarrow \text{Phoneme}$$
$$\text{Phonemes} \rightarrow \text{Phoneme Phonemes}$$

Figure 2: The unigram word adaptor grammar of Figure 1 where regular expressions are expanded using new unadapted right-branching nonterminals.

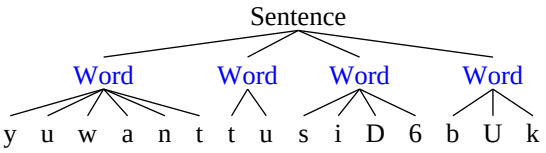

Figure 3: A parse of the phonemic representation of "you want to see the book" produced by unigram word adaptor grammar of Figure 1. Only nonterminal nodes labeled with adapted nonterminals and the start symbol are shown.

The unigram word adaptor grammar generates parses such as the one shown in Figure 3. With $\alpha = 1$ and $\alpha = 10$ we obtained a word segmentation f-score of 0.55. Depending on the run, between $1,100$ and $1,400$ subtrees (i.e., new rules) were found for Word. As reported in Goldwater et al. (2006a) and Goldwater et al. (2007), a unigram word segmentation model tends to undersegment and misanalyse collocations as individual words. This is presumably because the unigram model has no way to capture dependencies between words in collocations except to make the collocation into a single word.

3.2 Unigram morphology adaptor grammar

This section investigates whether learning morphology together with word segmentation improves word segmentation accuracy. Johnson et al. (2007a) presented an adaptor grammar for segmenting verbs into stems and suffixes that implements the DP-

$$\text{Sentence} \rightarrow \text{Word}^{+}$$
$$\underline{\text{Word}} \rightarrow \text{Stem}\ (\text{Suffix})$$
$$\underline{\text{Stem}} \rightarrow \text{Phoneme}^{+}$$
$$\underline{\text{Suffix}} \rightarrow \text{Phoneme}^{+}$$

Figure 4: The unigram morphology adaptor grammar, which generates each Sentence as a sequence of Words, and each Word as a Stem optionally followed by a Suffix. Parentheses indicate optional constituents.

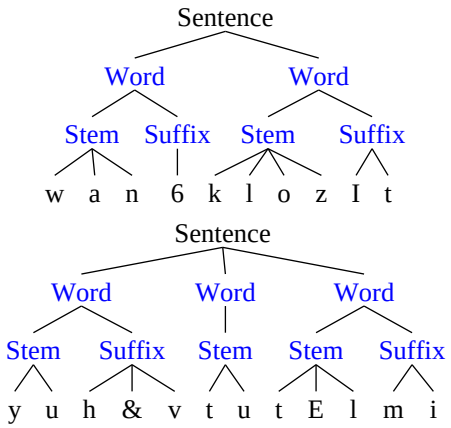

Figure 5: Parses of "wanna close it" and "you have to tell me" produced by the unigram morphology grammar of Figure 4. The first parse was chosen because it demonstrates how the grammar is intended to analyse "wanna" into a Stem and Suffix, while the second parse shows how the grammar tends to use Stem and Suffix to capture collocations.

based unsupervised morphological analysis model presented by Goldwater et al. (2006b). Here we combine that adaptor grammar with the unigram word segmentation grammar to produce the adaptor grammar shown in Figure 4, which is designed to simultaneously learn both word segmentation and morphology.

Parentheses indicate optional constituents in these rules, so this grammar says that a Sentence consists of a sequence of Words, and each Word consists of a Stem followed by an optional Suffix. The categories Word, Stem and Suffix are adapted, which means that the grammar learns the Words, Stems and Suffixes that occur in the training corpus. Technically this grammar implements a *Hierarchical* Dirichlet Process (HDP) (Teh et al., 2006) because the base distribution for the Word DP is itself constructed from the Stem and Suffix distributions, which are themselves generated by DPs.

This grammar recovers words with an f-score of only 0.46 with $\alpha = 1$ or $\alpha = 10$, which is considerably less accurate than the unigram model of section 3.1. Typical parses are shown in Figure 5. The unigram morphology grammar tends to misanalyse even longer collocations as words than the unigram word grammar does. Inspecting the parses shows that rather than capturing morphological structure, the Stem and Suffix categories typically expand to words themselves, so the Word category expands to a collocation. It may be possible to correct this by "tuning" the grammar's hyperparameters, but we did not attempt this here.

These results are not too surprising, since the kind of regular stem-suffix morphology that this grammar can capture is not common in the Brent corpus. It is possible that a more sophisticated model of morphology, or even a careful tuning of the Bayesian prior parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, would produce better results.

3.3 Unigram syllable adaptor grammar

PCFG estimation procedures have been used to model the supervised and unsupervised acquisition of syllable structure (Müller, 2001; Müller, 2002); and the best performance in unsupervised acquisition is obtained using a grammar that encodes linguistically detailed properties of syllables whose rules are inferred using a fairly complex algorithm (Goldwater and Johnson, 2005). While that work studied the acquisition of syllable structure from isolated words, here we investigate whether learning syllable structure together with word segmentation improves word segmentation accuracy. Modeling syllable structure is a natural application of adaptor grammars, since the grammar can learn the possible onset and coda clusters, rather than requiring them to be stipulated in the grammar.

In the unigram syllable adaptor grammar shown in Figure 7, Consonant expands to any consonant and Vowel expands to any vowel. This grammar defines a Word to consist of up to three Syllables, where each Syllable consists of an Onset and a Rhyme and a Rhyme consists of a Nucleus and a Coda. Following Goldwater and Johnson (2005), the grammar differentiates between OnsetI, which expands to word-initial onsets, and Onset,

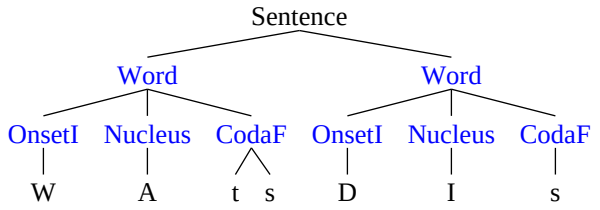

Figure 6: A parse of "what's this" produced by the unigram syllable adaptor grammar of Figure 7. (Only adapted non-root nonterminals are shown in the parse).

which expands to non-word-initial onsets, and between CodaF, which expands to word-final codas, and Coda, which expands to non-word-final codas. Note that we do not need to distinguish specific positions within the Onset and Coda clusters as Goldwater and Johnson (2005) did, since the adaptor grammar learns these clusters directly. Just like the unigram morphology grammar, the unigram syllable grammar also defines a HDP because the base distribution for Word is defined in terms of the Onset and Rhyme distributions.

The unigram syllable grammar achieves a word segmentation f-score of 0.52 at $\alpha = 1$, which is also lower than the unigram word grammar achieves. Inspection of the parses shows that the unigram syllable grammar also tends to misanalyse long collocations as Words. Specifically, it seems to misanalyse function words as associated with the content words next to them, perhaps because function words tend to have simpler initial and final clusters.

We cannot compare our syllabification accuracy with Goldwater's and others' previous work because that work used different, supervised training data and phonological representations based on British rather than American pronunciation.

3.4 Collocation word adaptor grammar

Goldwater et al. (2006a) showed that modeling dependencies between adjacent words dramatically improves word segmentation accuracy. It is not possible to write an adaptor grammar that directly implements Goldwater's bigram word segmentation model because an adaptor grammar has one DP per adapted nonterminal (so the number of DPs is fixed in advance) while Goldwater's bigram model has one DP per word type, and the number of word types is not known in advance. However it is possible for an adaptor grammar to generate a sentence as a sequence of *collocations*, each of which consists of a sequence of words. These collocations give the grammar a way to model dependencies between words.

$$
\begin{aligned}
\text{Sentence} &\rightarrow \text{Word}^+ \\
\underline{\text{Word}} &\rightarrow \text{SyllableIF} \\
\underline{\text{Word}} &\rightarrow \text{SyllableI SyllableF} \\
\underline{\text{Word}} &\rightarrow \text{SyllableI Syllable SyllableF} \\
\text{Syllable} &\rightarrow (\text{Onset})\ \text{Rhyme} \\
\text{SyllableI} &\rightarrow (\text{OnsetI})\ \text{Rhyme} \\
\text{SyllableF} &\rightarrow (\text{Onset})\ \text{RhymeF} \\
\text{SyllableIF} &\rightarrow (\text{OnsetI})\ \text{RhymeF} \\
\text{Rhyme} &\rightarrow \text{Nucleus}\ (\text{Coda}) \\
\text{RhymeF} &\rightarrow \text{Nucleus}\ (\text{CodaF}) \\
\underline{\text{Onset}} &\rightarrow \text{Consonant}^+ \\
\underline{\text{OnsetI}} &\rightarrow \text{Consonant}^+ \\
\underline{\text{Coda}} &\rightarrow \text{Consonant}^+ \\
\underline{\text{CodaF}} &\rightarrow \text{Consonant}^+ \\
\underline{\text{Nucleus}} &\rightarrow \text{Vowel}^+
\end{aligned}
$$

Figure 7: The unigram syllable adaptor grammar, which generates each word as a sequence of up to three Syllables. Word-initial Onsets and word-final Codas are distinguished using the suffixes "I" and "F" respectively; these are propagated through the grammar to ensure that these appear in the correct positions.

$$
\begin{aligned}
\text{Sentence} &\rightarrow \text{Colloc}^+ \\
\underline{\text{Colloc}} &\rightarrow \text{Word}^+ \\
\underline{\text{Word}} &\rightarrow \text{Phoneme}^+
\end{aligned}
$$

Figure 8: The collocation word adaptor grammar, which generates a Sentence as sequence of Colloc(ations), each of which consists of a sequence of Words.

With the DP concentration parameters $\alpha = 1000$ we obtained a f-score of 0.76, which is approximately the same as the results reported by Goldwater et al. (2006a) and Goldwater et al. (2007). This suggests that the collocation word adaptor grammar can capture inter-word dependencies similar to those that improve the performance of Goldwater's bigram segmentation model.

3.5 Collocation morphology adaptor grammar

One of the advantages of working within a grammatical framework is that it is often easy to combine

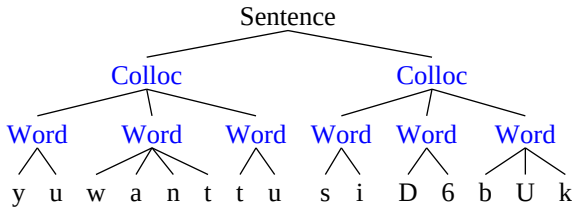

Figure 9: A parse of "you want to see the book" produced by the collocation word adaptor grammar of Figure 8.

$$\begin{aligned}
\text{Sentence} &\rightarrow \text{Colloc}^+ \\
\underline{\text{Colloc}} &\rightarrow \text{Word}^+ \\
\underline{\text{Word}} &\rightarrow \text{Stem (Suffix)} \\
\underline{\text{Stem}} &\rightarrow \text{Phoneme}^+ \\
\underline{\text{Suffix}} &\rightarrow \text{Phoneme}^+
\end{aligned}$$

Figure 10: The collocation morphology adaptor grammar, which generates each Sentence as a sequence of Colloc(ations), each Colloc as a sequence of Words, and each Word as a Stem optionally followed by a Suffix.

different grammar fragments into a single grammar. In this section we combine the collocation aspect of the previous grammar with the morphology component of the grammar presented in section 3.2 to produce a grammar that generates Sentences as sequences of Colloc(ations), where each Colloc consists of a sequence of Words, and each Word consists of a Stem followed by an optional Suffix, as shown in Figure 10.

This grammar achieves a word segmentation f-score of 0.73 at $\alpha = 100$, which is much better than the unigram morphology grammar of section 3.2, but not as good as the collocation word grammar of the previous section. Inspecting the parses shows

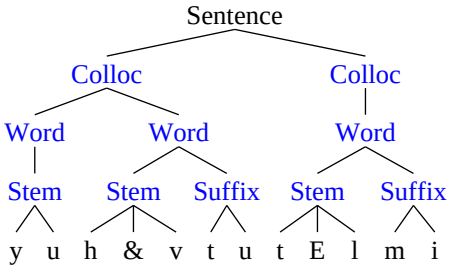

Figure 11: A parse of the phonemic representation of "you have to tell me" using the collocation morphology adaptor grammar of Figure 10.

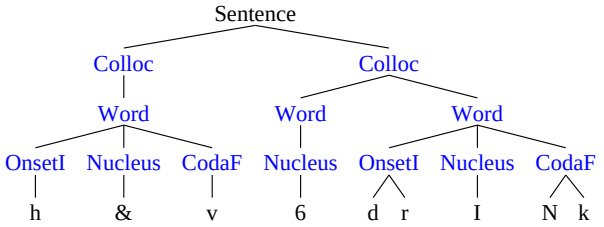

Figure 12: A parse of "have a drink" produced by the collocation syllable adaptor grammar. (Only adapted non-root nonterminals are shown in the parse).

that while the ability to directly model collocations reduces the number of collocations misanalysed as words, function words still tend to be misanalysed as morphemes of two-word collocations. In fact, some of the misanalyses have a certain plausibility to them (e.g., "to" is often analysed as the suffix of verbs such as "have", "want" and "like", while "me" is often analysed as a suffix of verbs such as "show" and "tell"), but they lower the word f-score considerably.

3.6 Collocation syllable adaptor grammar

The collocation syllable adaptor grammar is the same as the unigram syllable adaptor grammar of Figure 7, except that the first production is replaced with the following pair of productions.

$$\begin{aligned}
\text{Sentence} &\rightarrow \text{Colloc}^+ \\
\underline{\text{Colloc}} &\rightarrow \text{Word}^+
\end{aligned}$$

This grammar generates a Sentence as a sequence of Colloc(ations), each of which is composed of a sequence of Words, each of which in turn is composed of a sequence of Syll(ables).

This grammar achieves a word segmentation f-score of 0.78 at $\alpha = 100$, which is the highest f-score of any of the grammars investigated in this paper, including the collocation word grammar, which models collocations but not syllables. To confirm that the difference is significant, we ran a Wilcoxon test to compare the f-scores obtained from 8 runs of the collocation syllable grammar with $\alpha = 100$ and the collocation word grammar with $\alpha = 1000$, and found that the difference is significant at $p = 0.006$.

4 Conclusion and future work

This paper has shown how adaptor grammars can be used to study a variety of different linguistic hy-

potheses about the interaction of morphology and syllable structure with word segmentation. Technically, adaptor grammars are a way of specifying a variety of Hierarchical Dirichlet Processes (HDPs) that can spread their support over an unbounded number of distinct subtrees, giving them the ability to learn which subtrees are most useful for describing the training corpus. Thus adaptor grammars move beyond simple parameter estimation and provide a principled approach to the Bayesian estimation of at least some types of linguistic structure. Because of this, less linguistic structure needs to be "built in" to an adaptor grammar compared to a comparable PCFG. For example, the adaptor grammars for syllable structure presented in sections 3.3 and 3.6 learn more information about syllable onsets and codas than the PCFGs presented in Goldwater and Johnson (2005).

We used adaptor grammars to study the effects of modeling morphological structure, syllabification and collocations on the accuracy of a standard unsupervised word segmentation task. We showed how adaptor grammars can implement a previously investigated model of unsupervised word segmentation, the unigram word segmentation model. We then investigated adaptor grammars that incorporate one additional kind of information, and found that modeling collocations provides the greatest improvement in word segmentation accuracy, resulting in a model that seems to capture many of the same interword dependencies as the bigram model of Goldwater et al. (2006b).

We then investigated grammars that combine these kinds of information. There does not seem to be a straight forward way to design an adaptor grammar that models both morphology and syllable structure, as morpheme boundaries typically do not align with syllable boundaries. However, we showed that an adaptor grammar that models collocations and syllable structure performs word segmentation more accurately than an adaptor grammar that models either collocations or syllable structure alone. This is not surprising, since syllable onsets and codas that occur word-peripherally are typically different to those that appear word-internally, and our results suggest that by tracking these onsets and codas, it is possible to learn more accurate word segmentation.

There are a number of interesting directions for future work. In this paper all of the hyperparameters α_A were tied and varied simultaneously, but it is desirable to learn these from data as well. Just before the camera-ready version of this paper was due we developed a method for estimating the hyperparameters by putting a vague Gamma hyper-prior on each α_A and sampled using Metropolis-Hastings with a sequence of increasingly narrow Gamma proposal distributions, producing results for each model that are as good or better than the best ones reported in Table 1.

The adaptor grammars presented here barely scratch the surface of the linguistically interesting models that can be expressed as Hierarchical Dirichlet Processes. The models of morphology presented here are particularly naive—they only capture regular concatenative morphology consisting of one paradigm class—which may partially explain why we obtained such poor results using morphology adaptor grammars. It's straight forward to design an adaptor grammar that can capture a finite number of concatenative paradigm classes (Goldwater et al., 2006b; Johnson et al., 2007a). We'd like to learn the number of paradigm classes from the data, but doing this would probably require extending adaptor grammars to incorporate the kind of adaptive state-splitting found in the iHMM and iPCFG (Liang et al., 2007). There is no principled reason why this could not be done, i.e., why one could not design an HDP framework that simultaneously learns both the fragments (as in an adaptor grammar) and the states (as in an iHMM or iPCFG).

However, inference with these more complex models will probably itself become more complex. The MCMC sampler of Johnson et al. (2007a) used here is satifactory for small and medium-sized problems, but it would be very useful to have more efficient inference procedures. It may be possible to adapt efficient split-merge samplers (Jain and Neal, 2007) and Variational Bayes methods (Teh et al., 2008) for DPs to adaptor grammars and other linguistic applications of HDPs.

Acknowledgments

This research was funded by NSF awards 0544127 and 0631667.

References

N. Bernstein-Ratner. 1987. The phonology of parent-child speech. In K. Nelson and A. van Kleeck, editors, *Children's Language*, volume 6. Erlbaum, Hillsdale, NJ.

Rens Bod. 1998. *Beyond grammar: an experience-based theory of language*. CSLI Publications, Stanford, California.

Sharon Goldwater and Mark Johnson. 2005. Representational bias in unsupervised learning of syllable structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 112–119, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006a. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia, July. Association for Computational Linguistics.

Sharon Goldwater, Tom Griffiths, and Mark Johnson. 2006b. Interpolating between types and tokens by estimating power-law generators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 459–466, Cambridge, MA. MIT Press.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2007. Distributional cues to word boundaries: Context is important. In David Bamman, Tatiana Magnitskaia, and Colleen Zaller, editors, *Proceedings of the 31st Annual Boston University Conference on Language Development*, pages 239–250, Somerville, MA. Cascadilla Press.

Sonia Jain and Radford M. Neal. 2007. Splitting and merging components of a nonconjugate dirichlet process mixture model. *Bayesian Analysis*, 2(3):445–472.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007a. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 139–146, Rochester, New York, April. Association for Computational Linguistics.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007b. Adaptor Grammars: A framework for specifying compositional nonparametric Bayesian models. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 641–648. MIT Press, Cambridge, MA.

Aravind Joshi. 2003. Tree adjoining grammars. In Ruslan Mikkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 483–501. Oxford University Press, Oxford, England.

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697.

Brian MacWhinney and Catherine Snow. 1985. The child language data exchange system. *Journal of Child Language*, 12:271–296.

Karin Müller. 2001. Automatic detection of syllable boundaries combining the advantages of treebank and bracketed corpora training. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*.

Karin Müller. 2002. Probabilistic context-free grammars for phonology. In *Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 70–80, Philadelphia.

Andreas Stolcke. 1994. *Bayesian Learning of Probabilistic Language Models*. Ph.D. thesis, University of California, Berkeley.

Y. W. Teh, M. Jordan, M. Beal, and D. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581.

Yee Whye Teh, Kenichi Kurihara, and Max Welling. 2008. Collapsed variational inference for hdp. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA.