Unsupervised word segmentation for Sesotho using Adaptor Grammars

Mark Johnson
Brown University
Mark_Johnson@Brown.edu

Abstract

This paper describes a variety of non-parametric Bayesian models of word segmentation based on *Adaptor Grammars* that model different aspects of the input and incorporate different kinds of prior knowledge, and applies them to the Bantu language Sesotho. While we find overall word segmentation accuracies lower than these models achieve on English, we also find some interesting differences in which factors contribute to better word segmentation. Specifically, we found little improvement to word segmentation accuracy when we modeled contextual dependencies, while modeling morphological structure did improve segmentation accuracy.

1 Introduction

A Bayesian approach to learning (Bishop, 2006) is especially useful for computational models of language acquisition because we can use it to study the effect of different kinds and amounts of *prior knowledge* on the learning process. The Bayesian approach is agnostic as to what this prior knowledge might consist of; the prior could encode the kinds of rich universal grammar hypothesised by e.g., Chomsky (1986), or it could express a vague non-linguistic preference for simpler as opposed to more complex models, as in some of the grammars discussed below. Clearly there's a wide range of possible priors, and one of the exciting possibilities raised by Bayesian methods is that we may soon be able to empirically evaluate the potential contribution of different kinds of prior knowledge to language learning.

The Bayesian framework is surprisingly flexible. The bulk of the work on Bayesian inference is on *parametric models*, where the goal is to learn the value of a set of parameters (much as in Chomsky's Principles and Parameters conception of learning). However, recently Bayesian methods for *nonparametric inference* have been developed, in which the parameters themselves, as well as their values, are learned from data. (The term "nonparametric" is perhaps misleading here: it does not mean that the models have no parameters, rather it means that the learning process considers models with different sets of parameters). One can think of the prior as providing an infinite set of possible parameters, from which a learner selects a subset with which to model their language.

If one pairs each of these infinitely-many parameters with possible structures (or equivalently, rules that generate such structures) then these nonparametric Bayesian learning methods can learn the structures relevant to a language. Determining whether methods such as these can in fact learn linguistic structure bears on the nature vs. nurture debates in language acquisition, since one of the arguments for the nativist position is that there doesn't seem to be a way to learn structure from the input that children receive.

While there's no reason why these methods can't be used to learn the syntax and semantics of human languages, much of the work to date has focused on lower-level learning problems such as morphological structure learning (Goldwater et al., 2006b) and word segmentation, where the learner is given unsegmented broad-phonemic utterance transcriptions

Proceedings of the Tenth Meeting of the ACL Special Interest Group on Computational Morphology and Phonology, pages 20–27,
Columbus, Ohio, USA June 2008.

and has to identify the word boundaries (Goldwater et al., 2006a; Goldwater et al., 2007). One reason for this is that these problems seem simpler than learning syntax, where the non-linguistic context plausibly supplies important information to human learners. Virtually everyone agrees that the set of possible morphemes and words, if not infinite, is astronomically large, so it seems plausible that humans use some kind of nonparametric procedure to learn the lexicon.

Johnson et al. (2007) introduced *Adaptor Grammars* as a framework in which a wide variety of linguistically-interesting nonparametric inference problems can be formulated and evaluated, including a number of variants of the models described by Goldwater (2007). Johnson (2008) presented a variety of different adaptor grammar word segmentation models and applied them to the problem of segmenting Brent's phonemicized version of the Bernstein-Ratner corpus of child-directed English (Bernstein-Ratner, 1987; Brent, 1999). The main results of that paper were the following:

1. it confirmed the importance of modeling contextual dependencies above the word level for word segmentation (Goldwater et al., 2006a),
2. it showed a small but significant improvement to segmentation accuracy by learning the possible syllable structures of the language together with the lexicon, and
3. it found no significant advantage to learning morphological structure together with the lexicon (indeed, that model confused morphological and lexical structure).

Of course the last result is a null result, and it's possible that a different model would be able to usefuly combine morphological learning with word segmentation.

This paper continues that research by applying the same kinds of models to Sesotho, a Bantu language spoken in Southern Africa. Bantu languages are especially interesting for this kind of study, as they have rich productive agglutinative morphologies and relatively transparent phonologies, as compared to languages such as Finnish or Turkish which have complex harmony processes and other phonological complexities. The relative clarity of Bantu has inspired previous computational work, such as the algorithm for learning Swahili morphology by Hu et al. (2005). The Hu et al. algorithm uses a Minimum Description Length procedure (Rissanen, 1989) that is conceptually related to the nonparametric Bayesian procedure used here. However, the work here is focused on determining whether the word segmentation methods that work well for English generalize to Sesotho and whether modeling morphological and/or syllable structure improves Sesotho word segmentation, rather than learning Sesotho morphological structure per se.

The rest of this paper is structured as follows. Section 2 informally reviews adaptor grammars and describes how they are used to specify different Bayesian models. Section 3 describes the Sesotho corpus we used and the specific adaptor grammars we used for word segmentation, and section 5 summarizes and concludes the paper.

2 Adaptor grammars

One reason why Probabilistic Context-Free Grammars (PCFGs) are interesting is because they are very simple and natural models of hierarchical structure. They are parametric models because each PCFG has a fixed number of rules, each of which has a numerical parameter associated with it. One way to construct nonparametric Bayesian models is to take a parametric model class and let one or more of their components grow unboundedly.

There are two obvious ways to construct nonparametric models from PCFGs. First, we can let the number of nonterminals grow unboundedly, as in the *Infinite PCFG*, where the nonterminals of the grammar can be indefinitely refined versions of a base PCFG (Liang et al., 2007). Second, we can fix the set of nonterminals but permit the number of rules or productions to grow unboundedly, which leads to Adaptor Grammars (Johnson et al., 2007).

At any point in learning, an Adaptor Grammar has a finite set of rules, but these can grow unboundedly (typically logarithmically) with the size of the training data. In a word-segmentation application these rules typically generate words or morphemes, so the learner is effectively learning the morphemes and words of its language.

The new rules learnt by an Adaptor Grammar are

compositions of old ones (that can themselves be compositions of other rules), so it's natural to think of these new rules as tree fragments, where each entire fragment is associated with its own probability. Viewed this way, an adaptor grammar can be viewed as learning the tree fragments or constructions involved in a language, much as in Bod (1998). For computational reasons adaptor grammars require these fragments to consist of subtrees (i.e., their yields are terminals).

We now provide an informal description of Adaptor Grammars (for a more formal description see Johnson et al. (2007)). An adaptor grammar consists of terminals V, nonterminals N (including a start symbol S), initial rules R and rule probabilities p, just as in a PCFG. In addition, it also has a vector of *concentration parameters* $\boldsymbol{\alpha}$, where $\alpha_A \geq 0$ is called the *(Dirichlet) concentration parameter* associated with nonterminal A.

The nonterminals A for which $\alpha_A > 0$ are *adapted*, which means that each subtree for A that can be generated using the initial rules R is considered as a potential rule in the adaptor grammar. If $\alpha_A = 0$ then A is *unadapted*, which means it expands just as in an ordinary PCFG.

Adaptor grammars are so-called because they adapt both the subtrees and their probabilities to the corpus they are generating. Formally, they are Hierarchical Dirichlet Processes that generate a distribution over distributions over trees that can be defined in terms of stick-breaking processes (Teh et al., 2006). It's probably easiest to understand them in terms of their conditional or sampling distribution, which is the probability of generating a new tree T given the trees $(T_1, \ldots, T_n)$ that the adaptor grammar has already generated.

An adaptor grammar can be viewed as generating a tree top-down, just like a PCFG. Suppose we have a node A to expand. If A is unadapted (i.e., $\alpha_A = 0$) then A expands just as in a PCFG, i.e., we pick a rule $A \rightarrow \beta \in R$ with probability $p_{A \rightarrow \beta}$ and recursively expand β. If A is adapted and has expanded n_A times before, then:

1. A expands to a subtree σ with probability $n_\sigma/(n_A+\alpha_A)$, where n_σ is the number of times A has expanded to subtree σ before, and
2. A expands to β where $A \rightarrow \beta \in R$ with probability $\alpha_A\, p_{A \rightarrow \beta}/(n_A + \alpha_A)$.

Thus an adapted nonterminal A expands to a previously expanded subtree σ with probability proportional to the number n_σ of times it was used before, and expands just as in a PCFG (i.e., using R) with probability proportional to the concentration parameter α_A. This parameter specifies how likely A is to expand into a potentially new subtree; as n_A and n_σ grow this becomes increasingly unlikely.

We used the publically available adaptor grammar inference software described in Johnson et al. (2007), which we modified slightly as described below. The basic algorithm is a Metropolis-within-Gibbs or Hybrid MCMC sampler (Robert and Casella, 2004), which resamples the parse tree for each sentence in the training data conditioned on the parses for the other sentences. In order to produce sample parses efficiently the algorithm constructs a PCFG approximation to the adaptor grammar which contains one rule for each adapted subtree σ, and uses a Metropolis accept/reject step to correct for the difference between the true adaptor grammar distribution and the PCFG approximation. With the datasets described below less than 0.1% of proposal parses from this PCFG approximation are rejected, so it is quite a good approximation to the adaptor grammar distribution.

On the other hand, at convergence this algorithm produces a sequence of samples from the posterior distribution over adaptor grammars, and this posterior distribution seems quite broad. For example, at convergence with the most stable of our models, each time a sentence's parse is resampled there is an approximately 25% chance of the parse changing. Perhaps this is not surprising given the comparatively small amount of training data and the fact that the models only use fairly crude distributional information.

As just described, adaptor grammars require the user to specify a concentration parameter α_A for each adapted nonterminal A. It's not obvious how this should be done. Previous work has treated α_A as an adjustable parameter, usually tying all of the α_A to some shared value which is adjusted to optimize task performance (say, word segmentation accuracy). Clearly, this is undesirable.

Teh et al. (2006) describes how to learn the con-

centration parameters $\boldsymbol{\alpha}$, and we modified their procedure for adaptor grammars. Specifically, we put a vague $\mathrm{Gamma}(10, 0.1)$ prior on each α_A, and after each iteration through the training data we performed 100 Metropolis-Hastings resampling steps for each α_A from an increasingly narrow Gamma proposal distribution. We found that the performance of the models with automatically learned concentration parameters $\boldsymbol{\alpha}$ was generally as good as the models where $\boldsymbol{\alpha}$ was tuned by hand (although admittedly we only tried three or four different values for $\boldsymbol{\alpha}$).

3 Models of Sesotho word segmentation

We wanted to make our Sesotho corpus as similar as possible to one used in previous work on word segmentation. We extracted all of the non-child utterances from the LI–LV files from the Sesotho corpus of child speech (Demuth, 1992), and used the Sesotho gloss as our gold-standard corpus (we did not phonemicize them as Sesotho orthography is very close to phonemic). This produced 8,503 utterances containing 21,037 word tokens, 30,200 morpheme tokens and 100,113 phonemes. By comparison, the Brent corpus contains 9,790 utterances, 33,399 word tokens and 95,809 phonemes. Thus the Sesotho corpus contains approximately the same number of utterances and phonemes as the Brent corpus, but far fewer (and hence far longer) words. This is not surprising as the Sesotho corpus involves an older child and Sesotho, being an agglutinative language, tends to have morphologically complex words.

In the subsections that follow we describe a variety of adaptor grammar models for word segmentation. All of these models were given same Sesotho data, which consisted of the Sesotho gold-standard corpus described above with all word boundaries (spaces) and morpheme boundaries (hyphens) removed. We computed the f-score (geometric average of precision and recall) with which the models recovered the words or the morphemes annotated in the gold-standard corpus.

3.1 Unigram grammar

We begin by describing an adaptor grammar that simulates the unigram word segmentation model proposed by Goldwater et al. (2006a). In this model each utterance is generated as a sequence of words, and each word is a sequence of phonemes. This grammar contains three kinds of rules, including rules that expand the nonterminal Phoneme to all of the phonemes seen in the training data.

Model	word f-score	morpheme f-score
word	0.431	0.352
colloc	0.478	0.387
colloc2	0.467	0.389
word − syll	0.502	0.349
colloc − syll	0.476	0.372
colloc2 − syll	0.490	0.393
word − morph	0.529	0.321
word − smorph	0.556	0.378
colloc − smorph	0.537	0.352

Table 1: Summary of word and morpheme f-scores for the different models discussed in this paper.

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

Adapted non-terminals are indicated by underlining, so in the word grammar only the Word nonterminal is adapted. Our software doesn't permit regular expressions in rules, so we expand all Kleene stars in rules into right-recursive structures over new unadapted nonterminals. Figure 1 shows a sample parse tree generated by this grammar for the sentence:

u- e- nk- il- e kae
SM-OM-take-PERF-IN where
"You took it from where?"

This sentence shows a typical inflected verb, with a subject marker (glossed SM), an object marker (OM), perfect tense marker (PERF) and mood marker (IN). In order to keep the trees a managable size, we only display the root node, leaf nodes and nodes labeled with adapted nonterminals.

The word grammar has a word segmentation f-score of 43%, which is considerably below the 56% f-score the same grammar achieves on the Brent corpus. This difference presumably reflects the fact that Sesotho words are longer and more complex, and so segmentation is a harder task.

We actually ran the adaptor grammar sampler for

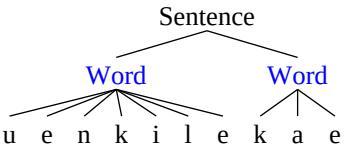

Figure 1: A sample (correct) parse tree generated by the word adaptor grammar for a Sesotho utterance.

the word grammar four times (as we did for all grammars discussed in this paper). Because the sampler is non-deterministic, each run produced a different series of sample segmentations. However, the average segmentation f-score seems to be very stable. The accuracies of the final sample of the four runs ranges between 42.8% and 43.7%. Similarly, one can compute the average f-score over the last 100 samples for each run; the average f-score ranges between 42.6% and 43.7%. Thus while there may be considerable uncertainty as to where the word boundaries are in any given sentence (which is reflected in fact that the word boundaries are very likely to change from sample to sample), the average accuracy of such boundaries seems very stable.

The final sample grammars contained the initial rules R, together with between 1,772 and 1,827 additional expansions for Word, corresponding to the cached subtrees for the adapted Word nonterminal.

3.2 Collocation grammar

Goldwater et al. (2006a) showed that incorporating a bigram model of word-to-word dependencies significantly improves word segmentation accuracy in English. While it is not possible to formulate such a bigram model as an adaptor grammar, Johnson (2008) showed that a similar improvement can be achieved in an adaptor grammar by explicitly modeling collocations or sequences of words. The colloc adaptor grammar is:

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

This grammar generates a Sentence as a sequence of Colloc(ations), where each Colloc(ation) is a sequence of Words. Figure 2 shows a sample parse tree generated by the colloc grammar. In terms of word segmentation, this grammar performs much worse

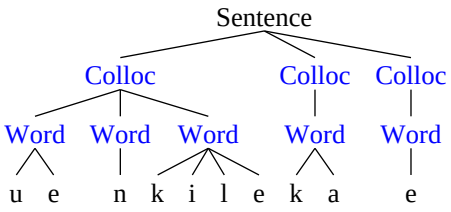

Figure 2: A sample parse tree generated by the colloc grammar. The substrings generated by Word in fact tend to be morphemes and Colloc tend to be words, which is how they are evaluated in Table 1.

than the word grammar, with an f-score of 27%.

In fact, it seems that the Word nonterminals typically expand to morphemes and the Colloc nonterminals typically expand to words. It makes sense that for a language like Sesotho, when given a grammar with a hierarchy of units, the learner would use the lower-level units as morphemes and the higher-level units as words. If we simply interpret the Word trees as morphemes and the Colloc trees as words we get a better word segmentation accuracy of 48% f-score.

3.3 Adding more levels

If two levels are better than one, perhaps three levels would be better than two? More specifically, perhaps adding another level of adaptation would permit the model to capture the kind of interword context dependencies that improved English word segmentation. Our colloc2 adaptor grammar includes the following rules:

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

This grammar generates sequences of Words grouped together in collocations, as in the previous grammar, but each Word now consists of a sequence of Morph(emes). Figure 3 shows a sample parse tree generated by the colloc2 grammar.

Interestingly, word segmentation f-score is 46.7%, which is slightly lower than that obtained by the simpler colloc grammar. Informally, it seems that when given an extra level of structure the colloc2 model uses it to describe structure internal

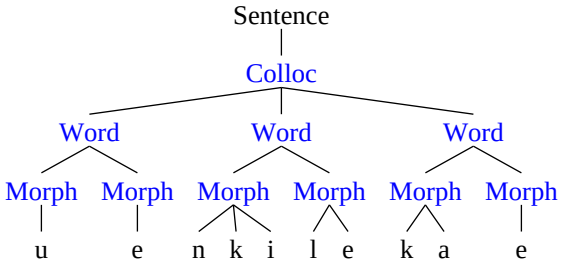

Figure 3: A sample parse tree generated by the colloc2 grammar.

to the word, rather than to capture interword dependencies. Perhaps this shouldn't be surprising, since Sesotho words in this corpus are considerably more complex than the English words in the Brent corpus.

4 Adding syllable structure

Johnson (2008) found a small but significant improvement in word segmentation accuracy by using an adaptor grammar that models English words as a sequence of syllables. The word – syll grammar builds in knowledge that syllables consist of an optional Onset, a Nuc(leus) and an optional Coda, and knows that Onsets and Codas are composes of consonants and that Nucleii are vocalic (and that syllabic consonsants are possible Nucleii), and learns the possible syllables of the language. The rules in the adaptor grammars that expand Word are changed to the following:

$$\begin{aligned}
\underline{\text{Word}} &\rightarrow \text{Syll}^{+} \\
\underline{\text{Syll}} &\rightarrow (\text{Onset})\,\text{Nuc}\,(\text{Coda}) \\
\underline{\text{Syll}} &\rightarrow \text{SC} \\
\underline{\text{Onset}} &\rightarrow \text{C}^{+} \\
\text{Nuc} &\rightarrow \text{V}^{+} \\
\text{Coda} &\rightarrow \text{C}^{+}
\end{aligned}$$

In this grammar C expands to any consonant and V expands to any vowel, SC expands to the syllablic consonants '*l*', '*m*' '*n*' and '*r*', and parentheses indicate optionality. Figure 4 shows a sample parse tree produced by the word – syll adaptor grammar (i.e., where Words are generated by a unigram model), while Figure 5 shows a sample parse tree generated by the corresponding colloc – syll adaptor grammar (where Words are generated as a part of a Collocation).

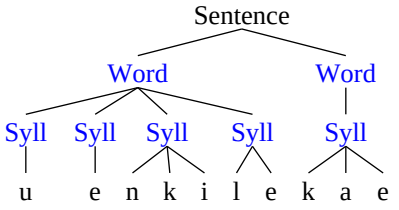

Figure 4: A sample parse tree generated by the word – syll grammar, in which Words consist of sequences of Syll(ables).

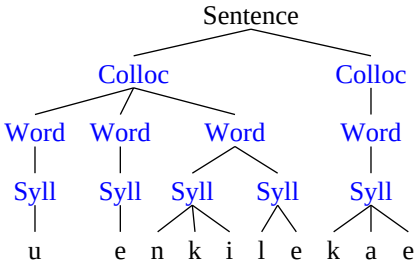

Figure 5: A sample parse tree generated by the colloc – syll grammar, in which Colloc(ations) consist of sequences of Words, which in turn consist of sequences of Syll(ables).

Building in this knowledge of syllable structure does improve word segmentation accuracy, but the best performance comes from the simplest word – syll grammar (with a word segmentation f-score of 50%).

4.1 Tracking morphological position

As we noted earlier, the various Colloc grammars wound up capturing a certain amount of morphological structure, even though they only implement a relatively simple unigram model of morpheme word order. Here we investigate whether we can improve word segmentation accuracy with more sophisticated models of morphological structure.

The word – morph grammar generates a word as a sequence of one to five morphemes. The relevant productions are the following:

$$\begin{aligned}
\underline{\text{Word}} &\rightarrow \text{T1}\,(\text{T2}\,(\text{T3}\,(\text{T4}\,(\text{T5})))) \\
\underline{\text{T1}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{T2}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{T3}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{T4}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{T5}} &\rightarrow \text{Phoneme}^{+}
\end{aligned}$$

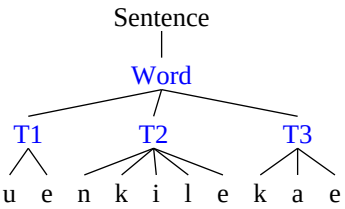

Figure 6: A sample parse tree generated by the word − morph grammar, in which Words consist of morphemes T1–T5, each of which is associated with specific lexical items.

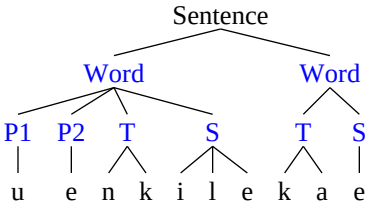

Figure 7: A sample parse tree generated by the word − smorph grammar, in which Words consist of up to five morphemes that satisfy prespecified ordering constraints.

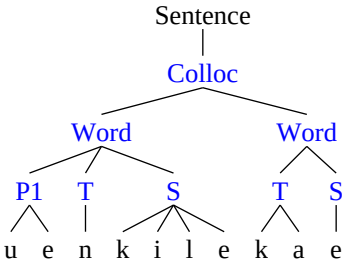

Figure 8: A sample parse tree generated by the colloc − smorph grammar, in which Colloc(ations) generate a sequence of Words, which in turn consist of up to five morphemes that satisfy prespecified ordering constraints.

While each morpheme is generated by a unigram character model, because each of these five morpheme positions is independently adapted, the grammar can learn which morphemes prefer to appear in which position. Figure 6 contains a sample parse generated by this grammar. Modifying the grammar in this way significantly improves word segmentation accuracy, achieving a word segmentation f-score of 53%.

Inspired by this, we decided to see what would happen if we built-in some specific knowledge of Sesotho morphology, namely that a word consists of a stem plus an optional suffix and zero to three optional prefixes. (This kind of information is often built into morphology learning models, either explicitly or implicitly via restrictions on the search procedure). The resulting grammar, which we call word − smorph, generates words as follows:

$$\begin{aligned}
\underline{\text{Word}} &\rightarrow (\text{P1}\,(\text{P2}\,(\text{P3})))\,\text{T}\,(\text{S}) \\
\underline{\text{P1}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{P2}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{P3}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{T}} &\rightarrow \text{Phoneme}^{+} \\
\underline{\text{S}} &\rightarrow \text{Phoneme}^{+}
\end{aligned}$$

Figure 7 contains a sample parse tree generated by this grammar. Perhaps not surprisingly, with this modification the grammar achieves the highest word segmentation f-score of any of the models examined in this paper, namely 55.6%.

Of course, this morphological structure is perfectly compatible with models which posit higher-level structure than Words. We can replace the Word expansion in the colloc grammar with one just given; the resulting grammar is called colloc − smorph, and a sample parse tree is given in Figure 8. Interestingly, this grammar achieves a lower accuracy than either of the two word-based morphology grammars we considered above.

5 Conclusion

Perhaps the most important conclusion to be drawn from this paper is that the methods developed for unsupervised word segmentation for English also work for Sesotho, despite its having radically different morphological structures to English. Just as with English, more structured adaptor grammars can achieve better word-segmentation accuracies than simpler ones. While we find overall word segmentation accuracies lower than these models achieve on English, we also found some interesting differences in which factors contribute to better word segmentation. Perhaps surprisingly, we found little improvement to word segmentation accuracy when we modeled contextual dependencies, even though these are most important in English. But including either morphological structure or syllable structure in the model improved word segmentation accu-

racy markedly, with morphological structure making a larger impact. Given how important morphology is in Sesotho, perhaps this is no surprise after all.

Acknowledgments

I'd like to thank Katherine Demuth for the Sesotho data and help with Sesotho morphology, my collaborators Sharon Goldwater and Tom Griffiths for their comments and suggestions about adaptor grammars, and the anonymous SIGMORPHON reviewers for their careful reading and insightful comments on the original abstract. 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.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

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

M. Brent. 1999. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34:71–105.

Noam Chomsky. 1986. *Knowledge of Language: Its Nature, Origin and Use*. Praeger, New York.

Katherine Demuth. 1992. Acquisition of Sesotho. In Dan Slobin, editor, *The Cross-Linguistic Study of Language Acquisition*, volume 3, pages 557–638. Lawrence Erlbaum Associates, Hillsdale, N.J.

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.

Sharon Goldwater. 2007. *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis, Brown University.

Yu Hu, Irina Matveeva, John Goldsmith, and Colin Sprague. 2005. Refining the SED heuristic for morpheme discovery: Another look at Swahili. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition*, pages 28–35, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. 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.

Mark Johnson. 2008. Using adaptor grammars to identifying synergies in the unsupervised acquisition of linguistic structure. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics*, Columbus, Ohio, June. Association for Computational Linguistics.

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.

Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Company, Singapore.

Christian P. Robert and George Casella. 2004. *Monte Carlo Statistical Methods*. Springer.

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