Note (May 30, 2008): The results in this paper for the HDP model were obtained using an implementation that was later discovered to contain a small bug. Updated results (which are qualitatively similar) can be found in “A Bayesian framework for word segmentation: Exploring the effects of context” (Goldwater et al., in submission). Please cite results from that paper in future publications.

Contextual Dependencies in Unsupervised Word Segmentation*

Sharon Goldwater and **Thomas L. Griffiths** and **Mark Johnson**
Department of Cognitive and Linguistic Sciences
Brown University
Providence, RI 02912
{Sharon_Goldwater,Tom_Griffiths,Mark_Johnson}@brown.edu

Abstract

Developing better methods for segmenting continuous text into words is important for improving the processing of Asian languages, and may shed light on how humans learn to segment speech. We propose two new Bayesian word segmentation methods that assume unigram and bigram models of word dependencies respectively. The bigram model greatly outperforms the unigram model (and previous probabilistic models), demonstrating the importance of such dependencies for word segmentation. We also show that previous probabilistic models rely crucially on suboptimal search procedures.

1 Introduction

Word segmentation, i.e., discovering word boundaries in continuous text or speech, is of interest for both practical and theoretical reasons. It is the first step of processing orthographies without explicit word boundaries, such as Chinese. It is also one of the key problems that human language learners must solve as they are learning language.

Many previous methods for unsupervised word segmentation are based on the observation that transitions between units (characters, phonemes, or syllables) within words are generally more predictable than transitions across word boundaries. Statistics that have been proposed for measuring these differences include “successor frequency” (Harris, 1954), “transitional probabilities” (Saffran et al., 1996), mutual information (Sun et al., 1998), “accessor variety” (Feng et al., 2004), and boundary entropy (Cohen and Adams, 2001).

While methods based on local statistics are quite successful, here we focus on approaches based on explicit probabilistic models. Formulating an explicit probabilistic model permits us to cleanly separate assumptions about the input and properties of likely segmentations from details of algorithms used to find such solutions. Specifically, this paper demonstrates the importance of contextual dependencies for word segmentation by comparing two probabilistic models that differ only in that the first assumes that the probability of a word is independent of its local context, while the second incorporates bigram dependencies between adjacent words. The algorithms we use to search for likely segmentations do differ, but so long as the segmentations they produce are close to optimal we can be confident that any differences in the segmentations reflect differences in the probabilistic models, i.e., in the kinds of dependencies between words.

We are not the first to propose explicit probabilistic models of word segmentation. Two successful word segmentation systems based on explicit probabilistic models are those of Brent (1999) and Venkataraman (2001). Brent’s Model-Based Dynamic Programming (MBDP) system assumes a unigram word distribution. Venkataraman uses standard unigram, bigram, and trigram language models in three versions of his system, which we refer to as n-gram Segmentation (NGS). Despite their rather different generative structure, the MBDP and NGS segmentation accuracies are very similar. Moreover, the segmentation accuracy of the NGS unigram, bigram, and trigram models hardly differ, suggesting that contextual dependencies are irrelevant to word segmentation. How-

*This work was partially supported by the following grants: NIH 1R01-MH60922, NIH RO1-DC000314, NSF IGERT-DGE-9870676, and the DARPA CALO project.

ever, the segmentations produced by both these methods depend crucially on properties of the search procedures they employ. We show this by exhibiting for each model a segmentation that is less accurate but more probable under that model.

In this paper, we present an alternative framework for word segmentation based on the Dirichlet process, a distribution used in nonparametric Bayesian statistics. This framework allows us to develop extensible models that are amenable to standard inference procedures. We present two such models incorporating unigram and bigram word dependencies, respectively. We use Gibbs sampling to sample from the posterior distribution of possible segmentations under these models.

The plan of the paper is as follows. In the next section, we describe MBDP and NGS in detail. In Section 3 we present the unigram version of our own model, the Gibbs sampling procedure we use for inference, and experimental results. Section 4 extends that model to incorporate bigram dependencies, and Section 5 concludes the paper.

2 NGS and MBDP

The NGS and MBDP systems are similar in some ways: both are designed to find utterance boundaries in a corpus of phonemically transcribed utterances, with known utterance boundaries. Both also use approximate online search procedures, choosing and fixing a segmentation for each utterance before moving onto the next. In this section, we focus on the very different probabilistic models underlying the two systems. We show that the optimal solution under the NGS model is the unsegmented corpus, and suggest that this problem stems from the fact that the model assumes a uniform prior over hypotheses. We then present the MBDP model, which uses a non-uniform prior but is difficult to extend beyond the unigram case.

2.1 NGS

NGS assumes that each utterance is generated independently via a standard n-gram model. For simplicity, we will discuss the unigram version of the model here, although our argument is equally applicable to the bigram and trigram versions. The unigram model generates an utterance u according to the grammar in Figure 1, so

$$P(u) = p_\$ (1 - p_\$)^{n-1} \prod_{j=1}^{n} P(w_j) \qquad (1)$$

$$\begin{array}{rll} 1 - p_\$ & U \rightarrow W\ U & \\ p_\$ & U \rightarrow W & \\ P(w) & W \rightarrow w & \quad \forall w \in \Sigma^* \end{array}$$

Figure 1: The unigram NGS grammar.

where u consists of the words $w_1 \ldots w_n$ and $p_\$$ is the probability of the utterance boundary marker \$. This model can be used to find the highest probability segmentation hypothesis h given the data d by using Bayes' rule:

$$P(h|d) \propto P(d|h)P(h)$$

NGS assumes a uniform prior $P(h)$ over hypotheses, so its goal is to find the solution that maximizes the likelihood $P(d|h)$.

Using this model, NGS's approximate search technique delivers competitive results. However, the true maximum likelihood solution is not competitive, since it contains no utterance-internal word boundaries. To see why not, consider the solution in which $p_\$ = 1$ and each utterance is a single 'word', with probability equal to the empirical probability of that utterance. Any other solution will match the empirical distribution of the data less well. In particular, a solution with additional word boundaries must have $1 - p_\$ > 0$, which means it wastes probability mass modeling unseen data (which can now be generated by concatenating observed utterances together).

Intuitively, the NGS model considers the unsegmented solution to be optimal because it ranks all hypotheses equally probable *a priori*. We know, however, that hypotheses that memorize the input data are unlikely to generalize to unseen data, and are therefore poor solutions. To prevent memorization, we could restrict our hypothesis space to models with fewer parameters than the number of utterances in the data. A more general and mathematically satisfactory solution is to assume a nonuniform prior, assigning higher probability to hypotheses with fewer parameters. This is in fact the route taken by Brent in his MBDP model, as we shall see in the following section.

2.2 MBDP

MBDP assumes a corpus of utterances is generated as a single probabilistic event with four steps:

1. Generate L, the number of lexical types.
2. Generate a phonemic representation for each type (except the utterance boundary type, \$).

3. Generate a token frequency for each type.
4. Generate an ordering for the set of tokens.

In a final deterministic step, the ordered tokens are concatenated to create an unsegmented corpus. This means that certain segmented corpora will produce the observed data with probability 1, and all others will produce it with probability 0. The posterior probability of a segmentation given the data is thus proportional to its prior probability under the generative model, and the best segmentation is that with the highest prior probability.

There are two important points to note about the MBDP model. First, the distribution over L assigns higher probability to models with fewer lexical items. We have argued that this is necessary to avoid memorization, and indeed the unsegmented corpus is not the optimal solution under this model, as we will show in Section 3. Second, the factorization into four separate steps makes it theoretically possible to modify each step independently in order to investigate the effects of the various modeling assumptions. However, the mathematical statement of the model and the approximations necessary for the search procedure make it unclear how to modify the model in any interesting way. In particular, the fourth step uses a uniform distribution, which creates a unigram constraint that cannot easily be changed. Since our research aims to investigate the effects of different modeling assumptions on lexical acquisition, we develop in the following sections a far more flexible model that also incorporates a preference for sparse solutions.

3 Unigram Model

3.1 The Dirichlet Process Model

Our goal is a model of language that prefers sparse solutions, allows independent modification of components, and is amenable to standard search procedures. We achieve this goal by basing our model on the *Dirichlet process* (DP), a distribution used in nonparametric Bayesian statistics. Our unigram model of word frequencies is defined as

$$\begin{aligned} w_i|G &\sim G \\ G|\alpha_0, P_0 &\sim \text{DP}(\alpha_0, P_0) \end{aligned}$$

where the *concentration parameter* α_0 and the *base distribution* P_0 are parameters of the model. Each word w_i in the corpus is drawn from a distribution G, which consists of a set of possible words (the lexicon) and probabilities associated with those words. G is generated from a DP(α_0, P_0) distribution, with the items in the lexicon being sampled from P_0 and their probabilities being determined by α_0, which acts like the parameter of an infinite-dimensional symmetric Dirichlet distribution. We provide some intuition for the roles of α_0 and P_0 below.

Although the DP model makes the distribution G explicit, we never deal with G directly. We take a Bayesian approach and integrate over all possible values of G. The conditional probability of choosing to generate a word from a particular lexical entry is then given by a simple stochastic process known as the Chinese restaurant process (CRP) (Aldous, 1985). Imagine a restaurant with an infinite number of tables, each with infinite seating capacity. Customers enter the restaurant and seat themselves. Let z_i be the table chosen by the ith customer. Then

$$P(z_i|\mathbf{z}_{-i}) = \begin{cases} \frac{n_k^{(\mathbf{z}_{-i})}}{i-1+\alpha_0} & 0 \leq k < K(\mathbf{z}_{-i}) \\ \frac{\alpha_0}{i-1+\alpha_0} & k = K(\mathbf{z}_{-i}) \end{cases} \quad (2)$$

where $\mathbf{z}_{-i} = z_1 \dots z_{i-1}$, $n_k^{(\mathbf{z}_{-i})}$ is the number of customers already sitting at table k, and $K(\mathbf{z}_{-i})$ is the total number of occupied tables. In our model, the tables correspond to (possibly repeated) lexical entries, having labels generated from the distribution P_0. The seating arrangement thus specifies a distribution over word tokens, with each customer representing one token. This model is an instance of the two-stage modeling framework described by Goldwater et al. (2006), with P_0 as the generator and the CRP as the adaptor.

Our model can be viewed intuitively as a *cache model*: each word in the corpus is either retrieved from a cache or generated anew. Summing over all the tables labeled with the same word yields the probability distribution for the ith word given previously observed words $\mathbf{w}_{-i}$:

$$P(w_i|\mathbf{w}_{-i}) = \frac{n_{w_i}^{(\mathbf{w}_{-i})}}{i-1+\alpha_0} + \frac{\alpha_0 P_0(w_i)}{i-1+\alpha_0} \quad (3)$$

where $n_w^{(\mathbf{w}_{-i})}$ is the number of instances of w observed in $\mathbf{w}_{-i}$. The first term is the probability of generating w from the cache (i.e., sitting at an occupied table), and the second term is the probability of generating it anew (sitting at an unoccupied table). The actual table assignments $\mathbf{z}_{-i}$ only become important later, in the bigram model.

There are several important points to note about this model. First, the probability of generating a particular word from the cache increases as more instances of that word are observed. This rich-get-richer process creates a power-law distribution on word frequencies (Goldwater et al., 2006), the same sort of distribution found empirically in natural language. Second, the parameter α_0 can be used to control how sparse the solutions found by the model are. This parameter determines the total probability of generating *any* novel word, a probability that decreases as more data is observed, but never disappears. Finally, the parameter P_0 can be used to encode expectations about the nature of the lexicon, since it defines a probability distribution across different novel words. The fact that this distribution is defined separately from the distribution on word frequencies gives the model additional flexibility, since either distribution can be modified independently of the other.

Since the goal of this paper is to investigate the role of context in word segmentation, we chose the simplest possible model for P_0, i.e. a unigram phoneme distribution:

$$P_0(w) = p_\#(1 - p_\#)^{n-1} \prod_{i=1}^{n} P(m_i) \qquad (4)$$

where word w consists of the phonemes $m_1 \ldots m_n$, and $p_\#$ is the probability of the word boundary $\#$. For simplicity we used a uniform distribution over phonemes, and experimented with different fixed values of $p_\#$.[1]

A final detail of our model is the distribution on utterance lengths, which is geometric. That is, we assume a grammar similar to the one shown in Figure 1, with the addition of a symmetric Beta($\frac{\tau}{2}$) prior over the probability of the U productions,[2] and the substitution of the DP for the standard multinomial distribution over the W productions.

3.2 Gibbs Sampling

Having defined our generative model, we are left with the problem of inference: we must determine the posterior distribution of hypotheses given our input corpus. To do so, we use Gibbs sampling, a standard Markov chain Monte Carlo method (Gilks et al., 1996). Gibbs sampling is an iterative procedure in which variables are repeatedly sampled from their conditional posterior distribution given the current values of all other variables in the model. The sampler defines a Markov chain whose stationary distribution is $P(h|d)$, so after convergence samples are from this distribution.

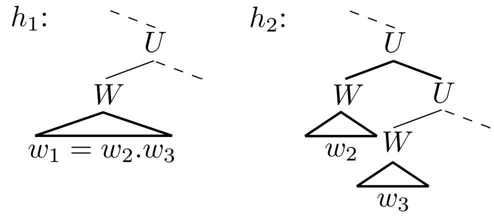

Figure 2: The two hypotheses considered by the unigram sampler. Dashed lines indicate possible additional structure. All rules except those in bold are part of h^-.

Our Gibbs sampler considers a single possible boundary point at a time, so each sample is from a set of two hypotheses, h_1 and h_2. These hypotheses contain all the same boundaries except at the one position under consideration, where h_2 has a boundary and h_1 does not. The structures are shown in Figure 2. In order to sample a hypothesis, we need only calculate the relative probabilities of h_1 and h_2. Since h_1 and h_2 are the same except for a few rules, this is straightforward. Let h^- be all of the structure shared by the two hypotheses, including n^- words, and let d be the observed data. Then

$$\begin{aligned} P(h_1|h^-, d) &= P(w_1|h^-, d) \\ &= \frac{n_{w_1}^{(h^-)} + \alpha_0 P_0(w_1)}{n^- + \alpha_0} \end{aligned} \qquad (5)$$

where the second line follows from Equation 3 and the properties of the CRP (in particular, that it is *exchangeable*, with the probability of a seating configuration not depending on the order in which customers arrive (Aldous, 1985)). Also,

$$\begin{aligned} &P(h_2|h^-, d) \\ &= P(r, w_2, w_3|h^-, d) \\ &= P(r|h^-, d) P(w_2|h^-, d) P(w_3|w_2, h^-, d) \\ &= \frac{n_r + \frac{\tau}{2}}{n^- + 1 + \tau} \cdot \frac{n_{w_2}^{(h^-)} + \alpha_0 P_0(w_2)}{n^- + \alpha_0} \\ &\quad \cdot \frac{n_{w_3}^{(h^-)} + I(w_2 = w_3) + \alpha_0 P_0(w_3)}{n^- + 1 + \alpha_0} \end{aligned} \qquad (6)$$

where n_r is the number of branching rules $r = U \rightarrow W\ U$ in h^-, and $I(.)$ is an indicator function taking on the value 1 when its argument is

[1] Note, however, that our model could be extended to learn both $p_\#$ and the distribution over phonemes.

[2] The Beta distribution is a Dirichlet distribution over two outcomes.

true, and 0 otherwise. The n_r term is derived by integrating over all possible values of $p_\$$, and noting that the total number of U productions in h^- is $n^- + 1$.

Using these equations we can simply proceed through the data, sampling each potential boundary point in turn. Once the Gibbs sampler converges, these samples will be drawn from the posterior distribution $P(h|d)$.

3.3 Experiments

In our experiments, we used the same corpus that NGS and MBDP were tested on. The corpus, supplied to us by Brent, consists of 9790 transcribed utterances (33399 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.

Because our Gibbs sampler is slow to converge, we used annealing to speed inference. We began with a temperature of $\gamma = 10$ and decreased γ in 10 increments to a final value of 1. A temperature of γ corresponds to raising the probabilities of h_1 and h_2 to the power of $\frac{1}{\gamma}$ prior to sampling.

We ran our Gibbs sampler for 20,000 iterations through the corpus (with $\gamma = 1$ for the final 2000) and evaluated our results on a single sample at that point. We calculated precision (P), recall (R), and F-score (F) on the word tokens in the corpus, where both boundaries of a word must be correct to count the word as correct. The induced lexicon was also scored for accuracy using these metrics (LP, LR, LF).

Recall that our DP model has three parameters: $\tau, p_\#$, and α_0. Given the large number of known utterance boundaries, we expect the value of τ to have little effect on our results, so we simply fixed $\tau = 2$ for all experiments. Figure 3 shows the effects of varying of $p_\#$ and α_0.[3] Lower values of $p_\#$ cause longer words, which tends to improve recall (and thus F-score) in the lexicon, but decrease token accuracy. Higher values of α_0 allow more novel words, which also improves lexicon recall,

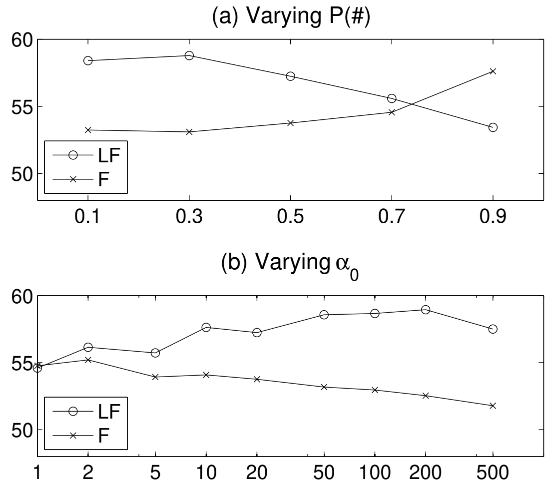

Figure 3: Word (F) and lexicon (LF) F-score (a) as a function of $p_\#$, with $\alpha_0 = 20$ and (b) as a function of α_0, with $p_\# = .5$.

but begins to degrade precision after a point. Due to the negative correlation between token accuracy and lexicon accuracy, there is no single best value for either $p_\#$ or α_0; further discussion refers to the solution for $p_\# = .5, \alpha_0 = 20$ (though others are qualitatively similar).

In Table 1(a), we compare the results of our system to those of MBDP and NGS.[4] Although our system has higher lexicon accuracy than the others, its token accuracy is much worse. This result occurs because our system often mis-analyzes frequently occurring words. In particular, many of these words occur in common collocations such as *what's that* and *do you*, which the system interprets as a single words. It turns out that a full 31% of the proposed lexicon and nearly 30% of tokens consist of these kinds of errors.

Upon reflection, it is not surprising that a unigram language model would segment words in this way. Collocations violate the unigram assumption in the model, since they exhibit strong word-to-word dependencies. The only way the model can capture these dependencies is by assuming that these collocations are in fact words themselves.

Why don't the MBDP and NGS unigram models exhibit these problems? We have already shown that NGS's results are due to its search procedure rather than its model. The same turns out to be true for MBDP. Table 2 shows the probabili-

[3] It is worth noting that all these parameters could be inferred. We leave this for future work.

[4] We used the implementations of MBDP and NGS available at http://www.speech.sri.com/people/anand/ to obtain results for those systems.

(a)	P	R	F	LP	LR	LF
NGS	**67.7**	**70.2**	**68.9**	52.9	51.3	52.0
MBDP	67.0	69.4	68.2	53.6	51.3	52.4
DP	61.9	47.6	53.8	**57.0**	**57.5**	**57.2**

(b)	P	R	F	LP	LR	LF
NGS	76.6	85.8	81.0	60.0	52.4	55.9
MBDP	77.0	86.1	81.3	60.8	53.0	56.6
DP	**94.2**	**97.1**	**95.6**	**86.5**	**62.2**	**72.4**

Table 1: Accuracy of the various systems, with best scores in bold. The unigram version of NGS is shown. DP results are with $p_{\#} = .5$ and $\alpha_0 = 20$. (a) Results on the true corpus. (b) Results on the permuted corpus.

Seg:	True	None	MBDP	NGS	DP
NGS	204.5	**90.9**	210.7	210.8	183.0
MBDP	208.2	321.7	217.0	218.0	**189.8**
DP	222.4	393.6	231.2	231.6	**200.6**

Table 2: Negative log probabilities (x 1000) under each model of the true solution, the solution with no utterance-internal boundaries, and the solutions found by each algorithm. Best solutions under each model are bold.

ties under each model of various segmentations of the corpus. From these figures, we can see that the MBDP model assigns higher probability to the solution found by our Gibbs sampler than to the solution found by Brent's own incremental search algorithm. In other words, Brent's model *does* prefer the lower-accuracy collocation solution, but his search algorithm instead finds a higher-accuracy but lower-probability solution.

We performed two experiments suggesting that our own inference procedure does not suffer from similar problems. First, we initialized our Gibbs sampler in three different ways: with no utterance-internal boundaries, with a boundary after every character, and with random boundaries. Our results were virtually the same regardless of initialization. Second, we created an artificial corpus by randomly permuting the words in the true corpus, leaving the utterance lengths the same. The artificial corpus adheres to the unigram assumption of our model, so if our inference procedure works correctly, we should be able to correctly identify the words in the permuted corpus. This is exactly what we found, as shown in Table 1(b). While all three models perform better on the artificial corpus, the improvements of the DP model are by far the most striking.

4 Bigram Model

4.1 The Hierarchical Dirichlet Process Model

The results of our unigram experiments suggested that word segmentation could be improved by taking into account dependencies between words. To test this hypothesis, we extended our model to incorporate bigram dependencies using a *hierarchical Dirichlet process* (HDP) (Teh et al., 2005). Our approach is similar to previous n-gram models using hierarchical Pitman-Yor processes (Goldwater et al., 2006; Teh, 2006). The HDP is appropriate for situations in which there are multiple distributions over similar sets of outcomes, and the distributions are believed to be similar. In our case, we define a bigram model by assuming each word has a different distribution over the words that follow it, but all these distributions are linked. The definition of our bigram language model as an HDP is

$$
\begin{aligned}
w_i | w_{i-1} = w, H_w &\sim H_w & \forall w \\
H_w | \alpha_1, G &\sim \mathrm{DP}(\alpha_1, G) & \forall w \\
G | \alpha_0, P_0 &\sim \mathrm{DP}(\alpha_0, P_0)
\end{aligned}
$$

That is, $P(w_i | w_{i-1} = w)$ is distributed according to H_w, a DP specific to word w. H_w is linked to the DPs for all other words by the fact that they share a common base distribution G, which is generated from another DP.[5]

As in the unigram model, we never deal with H_w or G directly. By integrating over them, we get a distribution over bigram frequencies that can be understood in terms of the CRP. Now, each word type w is associated with its own restaurant, which represents the distribution over words that follow w. Different restaurants are not completely independent, however: the labels on the tables in the restaurants are all chosen from a common base distribution, which is another CRP.

To understand the HDP model in terms of a grammar, we consider \$ as a special word type, so that w_i ranges over $\Sigma^* \cup \{\$\}$. After observing $\mathbf{w}_{-i}$, the HDP grammar is as shown in Figure 4,

[5] This HDP formulation is an oversimplification, since it does not account for utterance boundaries properly. The grammar formulation (see below) does.

$$
\begin{array}{llll}
P_2(w_i|\mathbf{w}_{-i},\mathbf{z}_{-i}) & U_{w_{i-1}} \rightarrow W_{w_i}\ U_{w_i} & \forall w_i \in \Sigma^*, \\
 & & \quad w_{i-1} \in \Sigma^* \cup \{\$\} \\
P_2(\$|\mathbf{w}_{-i},\mathbf{z}_{-i}) & U_{w_{i-1}} \rightarrow \$ & \forall w_{i-1} \in \Sigma^* \\
1 & W_{w_i} \rightarrow w_i & \forall w_i \in \Sigma^*
\end{array}
$$

Figure 4: The HDP grammar after observing $\mathbf{w}_{-i}$.

with

$$
P_2(w_i|h_{-i}) = \frac{n_{(w_{i-1},w_i)} + \alpha_1 P_1(w_i|h_{-i})}{n_{w_{i-1}} + \alpha_1} \quad (7)
$$

$$
P_1(w_i|h_{-i}) = \begin{cases} \frac{t_{\Sigma^*}+\frac{\tau}{2}}{t+\tau} \cdot \frac{t_{w_i}+\alpha_0 P_0(w_i)}{t_{\Sigma^*}+\alpha_0} & w_i \in \Sigma^* \\ \frac{t_\$+\frac{\tau}{2}}{t+\tau} & w_i = \$ \end{cases}
$$

where $h_{-i} = (\mathbf{w}_{-i}, \mathbf{z}_{-i})$; $t_\$, t_{\Sigma^*}$, and t_{w_i} are the total number of tables (across all words) labeled with \$, non-\$, and w_i, respectively; $t = t_\$ + t_{\Sigma^*}$ is the total number of tables; and $n_{(w_{i-1},w_i)}$ is the number of occurrences of the bigram (w_{i-1}, w_i). We have suppressed the superscript $(\mathbf{w}_{-i})$ notation in all cases. The base distribution shared by all bigrams is given by P_1, which can be viewed as a unigram backoff where the unigram probabilities are learned from the bigram table labels.

We can perform inference on this HDP bigram model using a Gibbs sampler similar to our unigram sampler. Details appear in the Appendix.

4.2 Experiments

We used the same basic setup for our experiments with the HDP model as we used for the DP model. We experimented with different values of α_0 and α_1, keeping $p_\# = .5$ throughout. Some results of these experiments are plotted in Figure 5. With appropriate parameter settings, both lexicon and token accuracy are higher than in the unigram model (dramatically so, for tokens), and there is no longer a negative correlation between the two. Only a few collocations remain in the lexicon, and most lexicon errors are on low-frequency words. The best values of α_0 are much larger than in the unigram model, presumably because all unique word types must be generated via P_0, but in the bigram model there is an additional level of discounting (the unigram process) before reaching P_0. Smaller values of α_0 lead to fewer word types with fewer characters on average.

Table 3 compares the optimal results of the HDP model to the only previous model incorporating bigram dependencies, NGS. Due to search, the performance of the bigram NGS model is not much different from that of the unigram model. In contrast, our HDP model performs far better than our DP model, leading to the highest published accuracy for this corpus on both tokens and lexical items. Overall, these results strongly support our hypothesis that modeling bigram dependencies is important for accurate word segmentation.

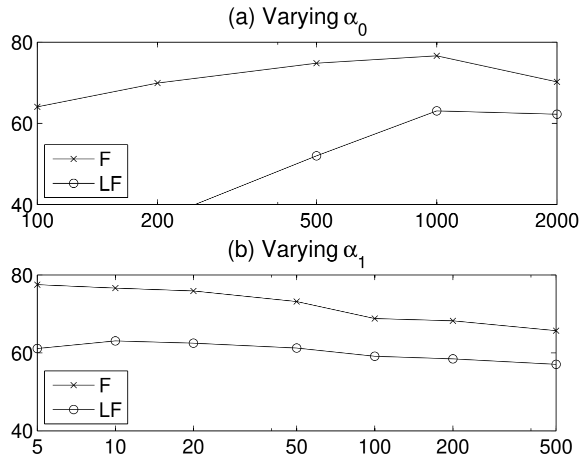

Figure 5: Word (F) and lexicon (LF) F-score (a) as a function of α_0, with $\alpha_1 = 10$ and (b) as a function of α_1, with $\alpha_0 = 1000$.

	P	R	F	LP	LR	LF
NGS	68.1	68.6	68.3	54.5	57.0	55.7
HDP	**79.4**	**74.0**	**76.6**	**67.9**	**58.9**	**63.1**

Table 3: Bigram system accuracy, with best scores in bold. HDP results are with $p_\# = .5$, $\alpha_0 = 1000$, and $\alpha_1 = 10$.

5 Conclusion

In this paper, we have introduced a new model-based approach to word segmentation that draws on techniques from Bayesian statistics, and we have developed models incorporating unigram and bigram dependencies. The use of the Dirichlet process as the basis of our approach yields sparse solutions and allows us the flexibility to modify individual components of the models. We have presented a method of inference using Gibbs sampling, which is guaranteed to converge to the posterior distribution over possible segmentations of a corpus.

Our approach to word segmentation allows us to investigate questions that could not be addressed satisfactorily in earlier work. We have shown that the search algorithms used with previous models of word segmentation do not achieve their ob-

$$
\begin{aligned}
P(h_1|h^-,d) &= \frac{n_{(w_l,w_1)}+\alpha_1 P_1(w_1|h^-,d)}{n_{w_l}+\alpha_1}\cdot\frac{n_{(w_1,w_r)}+I(w_l=w_1=w_r)+\alpha_1 P_1(w_r|h^-,d)}{n_{w_1}+1+\alpha_1}\\
P(h_2|h^-,d) &= \frac{n_{(w_l,w_2)}+\alpha_1 P_1(w_2|h^-,d)}{n_{w_l}+\alpha_1}\cdot\frac{n_{(w_2,w_3)}+I(w_l=w_2=w_3)+\alpha_1 P_1(w_3|h^-,d)}{n_{w_2}+1+\alpha_1}\cdot\\
&\qquad \frac{n_{(w_3,w_r)}+I(w_l=w_3,w_2=w_r)+I(w_2=w_3=w_r)+\alpha_1 P_1(w_r|h^-,d)}{n_{w_3}+1+I(w_2=w_4)+\alpha_1}
\end{aligned}
$$

Figure 6: Gibbs sampling equations for the bigram model. All counts are with respect to h^-.

jectives, which has led to misleading results. In particular, previous work suggested that the use of word-to-word dependencies has little effect on word segmentation. Our experiments indicate instead that bigram dependencies can be crucial for avoiding under-segmentation of frequent collocations. Incorporating these dependencies into our model greatly improved segmentation accuracy, and led to better performance than previous approaches on all measures.

References

D. Aldous. 1985. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, pages 1–198. Springer, Berlin.

C. Antoniak. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2:1152–1174.

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.

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

P. Cohen and N. Adams. 2001. An algorithm for segmenting categorical timeseries into meaningful episodes. In *Proceedings of the Fourth Symposium on Intelligent Data Analysis*.

H. Feng, K. Chen, X. Deng, and W. Zheng. 2004. Accessor variety criteria for chinese word extraction. *Computational Lingustics*, 30(1).

W.R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. 1996. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk.

S. Goldwater, T. Griffiths, and M. Johnson. 2006. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, Cambridge, MA. MIT Press.

Z. Harris. 1954. Distributional structure. *Word*, 10:146–162.

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

J. Saffran, E. Newport, and R. Aslin. 1996. Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35:606–621.

M. Sun, D. Shen, and B. Tsou. 1998. Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of COLING-ACL*.

Y. Teh, M. Jordan, M. Beal, and D. Blei. 2005. Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA.

Y. Teh. 2006. A Bayesian interpretation of interpolated kneser-ney. Technical Report TRA2/06, National University of Singapore, School of Computing.

A. Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):351–372.

Appendix

To sample from the posterior distribution over segmentations in the bigram model, we define h_1 and h_2 as we did in the unigram sampler so that for the corpus substring s, h_1 has a single word ($s = w_1$) where h_2 has two ($s = w_2.w_3$). Let w_l and w_r be the words (or \$) preceding and following s. Then the posterior probabilities of h_1 and h_2 are given in Figure 6. $P_1(.)$ can be calculated exactly using the equation in Section 4.1, but this requires explicitly tracking and sampling the assignment of words to tables. For easier and more efficient implementation, we use an approximation, replacing each table count t_{w_i} by its expected value $E[t_{w_i}]$. In a DP(α, P), the expected number of CRP tables for an item occurring n times is $\alpha \log \frac{n+\alpha}{\alpha}$ (Antoniak, 1974), so

$$E[t_{w_i}] = \alpha_1 \sum_j \log \frac{n_{(w_j,w_i)}+\alpha_1}{\alpha_1}$$

This approximation requires only the bigram counts, which we must track anyway.