Running Head: "Online Bayesian Word Segmentation"

Article Type: Full Article

Title: Online Learning Mechanisms for Bayesian Models of Word Segmentation

Authors: Lisa Pearl, Sharon Goldwater, Mark Steyvers

Affiliations:

Lisa Pearl: Department of Cognitive Sciences, University of California, Irvine

Sharon Goldwater: School of Informatics, University of Edinburgh

Mark Steyvers: Department of Cognitive Sciences, University of California, Irvine

Full Address for Correspondence:

Lisa Pearl

Department of Cognitive Sciences

3151 Social Science Plaza

University of California

Irvine, CA 92697-5100

Phone: 949-824-0156

Fax: 949-824-2307

Email: lpearl@uci.edu

Abstract

There has been growing interest in computational-level models of human cognition, which often analyze acquisition problems in terms of optimal Bayesian behavior. Human behavior appears consistent with the predictions of Bayesian ideal learners in a range of domains, including language (e.g., Griffiths & Tenenbaum, 2005; Xu & Tenenbaum, 2007). These ideal learner models aim to explain why humans behave as they do given the task and data they encounter, but typically avoid some questions addressed by more traditional psychological models, such as *how* the observed behavior is produced given constraints on memory and processing. Here, we use the task of word segmentation as a case study for investigating these questions within a Bayesian framework. We consider some limitations of the infant learner, and develop several online learning algorithms that take these limitations into account. Each algorithm can be viewed as a different method of approximating the same ideal learner. When tested on corpora of English child-directed speech, we find that the constrained learner's behavior depends non-trivially on how the learner's limitations are implemented. Interestingly, sometimes biases that are helpful to an ideal learner hinder a constrained learner, and in a few cases, constrained learners perform equivalently or better than the ideal learner. This suggests that the transition from a computational-level solution for acquisition to an algorithmic-level one is not straightforward.

Key words: algorithmic level, Bayesian models, computational level, English, ideal learning, online learning, processing limitations, word segmentation

1. Introduction

Language acquisition can be thought of as an induction problem, where the child observes some finite set of linguistic data, and must generalize beyond those data to form a more abstract representation of the language that can be used to produce and understand novel forms. In recent years, there has been growing interest and success in examining induction problems in many areas of cognition using a rational analysis approach (Oaksford & Chater, 1998), often through the use of Bayesian models (Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010; Griffiths, Kemp, & Tenenbaum, 2008; Tenenbaum, Griffiths, & Kemp, 2006). These models are typically used to examine problems at Marr's (1982) *computational level* of analysis, asking what the goal of the computation is and the general strategy by which it might be solved. They are ideal learners, which solve the induction problem optimally given particular assumptions about internal representation and the information available to the learner. Researchers have found that human behavior accords with that of the models in a number of domains, including language (e.g., Frank, Goodman, & Tenenbaum, 2009; Griffiths & Tenenbaum, 2005; Tenenbaum & Griffiths, 2001; Xu & Tenenbaum, 2007). These results are useful for showing that humans can integrate data optimally in many cases, but due to their focus on the computational level of explanation, the models tell us very little about the actual processes and mechanisms by which humans might achieve these behaviors - what Marr (1982) termed the *algorithmic level* of analysis. Indeed, one of the main criticisms of the Bayesian approach is that its frequent neglect of algorithmic-level explanations is unsatisfying to those who are interested in the processes by which humans make their inductive leaps (McClelland, Botvinick, Noelle, Plaut, Rogers, Seidenberg, & Smith, 2010). Given the limitations on human cognitive abilities (e.g., memory and processing), what kinds of algorithms might actually be used to compute the solutions that are optimal under a Bayesian model? What kinds of approximations might be required, and will different approximations lead to different results?

In this paper, we begin to address these very general questions by using the task of word segmentation by human infants as a case study. Our work can be viewed as one instance of a recent trend towards examining cognitively plausible implementations of Bayesian models (Sanborn, Griffiths, & Navarro, in press; Shi, Griffiths, Feldman, & Sanborn, in press). The problem of word segmentation is useful for our purposes because previous work has already analyzed the behaviors of ideal learners on this task, characterizing the differences between learners making different assumptions about the nature of the input data. Specifically (as discussed further in section 2.2), ideal learners who assume that words are predictive of other words (implemented using a *bigram* model) are more successful than learners who make the simpler *unigram* assumption that words are independent units (Goldwater, Griffiths, and Johnson, 2009).

The results of Goldwater, Griffiths, and Johnson (2009) (henceforth GGJ) were obtained by implementing their ideal learners using an algorithm that stored the entire corpus in memory. Here we ask whether more cognitively plausible algorithms, which take account of human memory and processing limitations, yield similar results (either quantitatively or qualitatively). We develop three different algorithms, where each algorithm can be viewed as a different method of approximating the same ideal learner. Like GGJ's ideal learners, our learners are unsupervised: their input consists of strings of phonemes with no word boundaries marked (except for utterance boundaries), and no initial lexicon of known words. Unlike GGJ's ideal learners, our learners use online (incremental) processing algorithms, which assume that the learner can only store and process a limited amount of data at once. When tested on a corpus of

English child-directed speech (Bernstein-Ratner, 1984), we find that the modeled learner's behavior depends non-trivially on how the learner's processing limitations are implemented. In particular, not all of the constrained learners exhibit the same qualitative differences between unigram and bigram versions as do the ideal learners. In some cases, the constrained unigram learners actually perform better than the ideal unigram learners, a behavior we discuss in light of Newport's "Less is More" hypothesis about human language acquisition (Newport, 1990). These results show that different kinds of cognitive limitations (constraints on the kinds of hypotheses learners entertain – e.g., whether or not words are predictive – and constraints on memory and processing) can interact in surprising and non-trivial ways. In particular, learners with restricted hypothesis spaces (here, unigram learners) may actually benefit from having restricted processing powers as well. Also, although the online learners we explore here are less successful than the most powerful learner we tested (the ideal bigram learner), we find they are able to utilize the statistical information in the data quite well, achieving comparable performance to other recent models of word segmentation, and far better performance than a simple transitional probability learner (Saffran et al., 1996).

2. Statistical Word Segmentation

Word segmentation is the task of identifying word boundaries in fluent speech, and is one of the first problems that infants must solve during language acquisition. A number of weak cues to word boundaries are present in fluent speech, and there is evidence that infants are able to use many of these, including phonotactics (Mattys et al., 1999), allophonic variation (Jusczyk et al., 1999b), metrical (stress) patterns (Morgan et al., 1995; Jusczyk et al., 1999c), effects of coarticulation (Johnson and Jusczyk, 2001), and statistical regularities among sequences of syllables (Saffran et al., 1996). With the exception of the last cue, all these cues are language-dependent, in the sense that the way the cue relates to word boundaries differs between languages. For example, English words are most often stressed on the initial syllable, while in other languages stress might be more typical on the penultimate or final syllable. Similarly, phonotactic constraints differ between languages, with legal words in one language being illegal in others. Thus, one would normally assume that in order to use these cues, infants must have already learned some of the words in the language in order to identify the dominant stress patterns and phonotactics (though see Blanchard, Heinz, & Golinkoff (2010) for one way language-specific phonotactics might be learned at the same time the initial segmentation problem is being solved). Since the point of word segmentation is to identify words in the first place, needing to know words in order to learn segmentation cues creates a chicken-and-egg problem. Fortunately, language-independent cues, such as statistical regularities between syllables or phonemes, can help infants out of this problem by allowing them to identify words (statistically coherent sound sequences) or word boundaries (statistically incoherent sound sequences) without already knowing what some words are. Infants appear to use these regularities earlier than other kinds of cues (Thiessen & Saffran, 2003), which suggests that strategies exploiting regularities in syllable or phoneme sequences can indeed provide the initial bootstrapping step for word segmentation. Consequently, although most of the other cues mentioned above are also statistical in nature, research (especially computational research) into statistical word learning has tended to focus on the use of syllable and phoneme regularities.

Before describing the models we will be exploring here, we first briefly review some other unsupervised models of word segmentation. A complete review of previous work is beyond the scope of this paper; instead we describe only some of the most recent models, which we will use for comparison to our own, and refer the reader to Goldwater (2006) for additional references.

2.1 Recent Statistical Word Segmentation Models

2.1.1 WordEnds

WordEnds (Fleck, 2008) is an unsupervised learning algorithm that operates in batch mode over sequences of phonemes. It focuses on boundaries (rather than words), and works by estimating the probabilities of different phoneme sequences occurring at word beginnings and endings, using these to identify locations that are likely to be word boundaries. An initial estimate of the probabilities is made by looking at the sequences that occur at utterance boundaries. These initial probabilities are used to hypothesize new word boundaries, which are then used to update the learner's estimates of probable word beginnings and endings, and the process continues iteratively. As a final cleanup measure, the learner merges hypothesized words together if the merged word occurs frequently enough in the existing segmentation. Although lexical items are used in the last step of the algorithm, WordEnds is not primarily word-based, in that it does not try to optimize a lexicon or explicitly model entire words or their relationships to each other (the phoneme sequences to the right and left of a word boundary are assumed to be independent, so the learner does not assume one word is predictive of the next).

2.1.2 Bootstrap Voting Experts

Hewlett and Cohen (2009) introduced a learning model called Bootstrap Voting Experts that works by chunking together sequences of phonemes that have low internal entropy and high boundary entropy. Thus, like WordEnds, this learner uses phonemic probability information, but no explicit model of words or a lexicon. Also like WordEnds, the learner iterates over the corpus in batch mode, improving subsequent segmentation hypotheses by information gained from previous segmentation hypotheses. Using a sliding window of a fixed length, two "voting experts" use accumulated entropy knowledge to vote whether a boundary should be inserted between two phonemes; if the number of votes exceeds a pre-determined threshold, the learner inserts a word boundary.

2.1.3 PHOCUS

Blanchard, Heinz, & Golinkoff (2010) created PHOCUS (PHOnotactic CUe Segmenter), an online learner that couples statistical word learning with phonotactic constraints. In particular, Blanchard et al. demonstrate a way in which language-specific phonotactic constraints, realized as likely and unlikely phonemic sequences, can be learned at the same time that segmentation is being attempted and an explicit lexicon is built. The learner uses an online algorithm, processing one utterance at a time and starting with an initially empty lexicon. The learner considers all possible segmentations of each utterance and chooses the one that is most probable, as computed by multiplying together the probabilities of each hypothesized word. The newly segmented words are then added to the lexicon (or, if they already exist, their counts are incremented). The relative frequency of each word in the lexicon is used as its probability when segmenting future utterances, and phonotactic probabilities are computed based on the current lexical items. Possible words that are not in the lexicon are assigned probabilities using these phonotactic probabilities. Note that when the learner starts out, no lexical items exist, so utterances will not be segmented, but added to the lexicon whole. However, since some utterances are individual words, they allow the learner to begin to find boundaries in later utterances.

This model is the most similar to those we introduce below, in that it uses an online algorithm and computes probabilities of different segmentations based on a learned lexicon. The main differences are in exactly how the probabilities are computed and the use of phonotactic probabilities in the PHOCUS learner. The best-performing variants of this learner also include domain-specific universally applicable knowledge about words: Namely, well-formed words must have a least one syllabic sound, and the learner knows which sounds are syllabic.
In addition, some of our learners assume that words are predictive of each other, whereas PHOCUS does not.

2.2 Bayesian word segmentation

The starting point of our research is the work of Goldwater, Griffiths, and Johnson (2007) (GGJ), which provides a Bayesian learning analysis of how statistical information could be used by infants to begin to segment words from continuous speech. It is a computational-level approach which defines the goal of learning as identifying the optimal segmentation of the input corpus from the space of all possible segmentations. Each segmentation implicitly defines a lexicon (the set of word types occurring in the segmentation); the learner decides which segmentation is optimal based on the words in the learned lexicon and their frequencies.

In the language of Bayesian analysis, the learner seeks to identify an explanatory hypothesis that both accounts for the observed data and conforms to prior expectations about what a reasonable hypothesis should look like. GGJ develop two models within a Bayesian framework where the learner is presented with some data *d* (a corpus of phonemically transcribed utterances, where each utterance is an unsegmented sequence of phonemes)[1] and seeks a hypothesis *h* (a segmentation of the corpus into a sequence of words) that both explains the data (i.e., concatenating together the words in *h* forms *d*) and has high prior probability. The optimal solution is the hypothesis with the highest probability, given the data:

(1) $P(h \mid d) \propto P(d \mid h)P(h)$

The learner determines the posterior probability of *h* having observed *d* based on *P(d | h)* – the likelihood of *d* being observed if *h* was true – and *P(h)* – the prior probability of *h*. Since a hypothesis is only a sequence of words, if the hypothesis sequence matches the observed sequence of phonemes, the likelihood is 1; if the hypothesis sequence does not match the observed sequence, the likelihood is 0. For example, hypotheses consistent with the observation sequence *lookatthedoggie* (we use orthographic rather than phonemic transcriptions here for clarity) include *lookatthedoggie, look at the doggie, lo oka t th edo ggie,* and *l o o k a t t h e d o g g i e*. Inconsistent hypotheses, for which *P(d|h)* = 0, include *i like pizza, a b c,* and *lookatthat*.

Since the likelihood is either 0 or 1, all of the work in the models is done by the prior distribution over hypotheses. For GGJ, the prior of *h* encodes the intuitions that words should be relatively short, and the lexicon should be relatively small. In addition, each of the two models encodes a different expectation about word behavior: in the *unigram* model, the learner assumes that words are statistically independent (i.e. context is not predictive); in the *bigram* model, words are assumed to be predictive units.

To encode these intuitions mathematically, GGJ use a model based on the *Dirichlet Process* from nonparametric Bayesian statistics (Ferguson, 1973), which can be summarized as follows. Imagine that the sequence of words $w_1 \ldots w_n$ in *h* is generated sequentially using a probabilistic generative process. In the unigram model, the identity of the *i*th word is chosen according to

(2) $P(w_i = w | w_1 \dots w_{i-1}) = \frac{n_{i-1}(w) + \alpha P_0(w)}{i-1+\alpha}$

where $n_{i-1}(w)$ is the number of times w has occurred in the previous i-1 words, α is a parameter of the model, and P_0 is a *base distribution* specifying the probability that a novel word will consist of the phonemes $x_1 \dots x_m$:

(3) $P_0(w = x_1 \dots x_m) = \prod_{j=1}^{m} P(x_j)$

The equation in (2) enforces the preference for a small lexicon by stating that the probability of a word is approximately proportional to the number of times that word has occurred previously. Thus, hypotheses where a small number of words occur frequently will be preferred over those with larger lexicons, where each word occurs less often. In addition, the first time a word appears in the sequence, $n_{i-1}(w) = 0$, so the probability of the word is completely determined by the equation in (3). Since (3) is a product of the phonemes in the word, words with fewer phonemes (i.e., shorter words) will be preferred. The GGJ model also includes a geometric distribution over utterance lengths, to account for the fact that the corpus consists of individual utterances. A more detailed description of both the unigram model and the bigram model (below) can be found in the original paper.

The bigram model, which is based on a *hierarchical Dirichlet Process* (Teh at al, 2006), is conceptually similar to the unigram model except that it tracks not only the frequencies of individual words, but also the frequencies of pairs of words. Just as the unigram model prefers hypotheses where a small number of words appear with high frequency, the bigram model prefers hypotheses where a small number of bigrams appear with high frequency (in addition to the assumptions of the unigram model). The model is defined as follows:

(4) $P(w_i = w | w_{i-1} = w', w_1 \ldots w_{i-2}) = \frac{n_{i-1}(w', w) + \beta P_1(w)}{n_{i-1}(w') + \beta}$

(5) $P_1(w_i = w) = \frac{b_{i-1}(w) + \gamma P_0(w)}{b_{i-1} + \gamma}$

where $n_{i-1}(w', w)$ is the number of times the bigram (*w',w*) has occurred in the first *i*-1 words, $b_{i-1}(w)$ is the number of times *w* has occurred as the second word of a bigram, b_{i-1} is the total number of bigrams, and β and γ are model parameters. The preference for hypotheses with relatively few distinct bigrams is enforced in the equation in (4), by making a bigram's probability approximately proportional to the number of times it has occurred before. This is analogous to the equation in (2) for the unigram model. When a new bigram is created, its probability is determined by the equation in (5), which assigns higher probability to new bigrams that use words that already occur in many other bigrams (i.e., the model assumes that a few words create bigrams very promiscuously, while most do not).

2.3. Ideal and Constrained Bayesian Inference

2.3.1. Ideal learners

To evaluate the performance of both the unigram and bigram Bayesian models in an ideal learner framework, GGJ used Gibbs sampling, a stochastic search procedure often used for ideal learner inference problems. Gibbs sampling, a type of Markov chain Monte Carlo procedure, is a batch algorithm that iterates over the corpus multiple times. Gibbs samplers are guaranteed to converge, which means that after a number of initial iterations (usually called "burn-in"), each iteration produces a sample from the posterior distribution of the model in question (here, either the unigram or bigram GGJ model). This convergence guarantee is what makes these samplers popular for ideal learner problems, since it means that the true posterior of the model can be examined without the effects of additional constraints imposed by the learning algorithm.

During each iteration of GGJ's Gibbs sampler, every possible boundary location (position between two phonemes) in the corpus is considered in turn. At each location *b*, the probability that *b* is a boundary is computed, given the current boundary locations in the rest of the corpus (details of this computation can be found in GGJ; critically, it is based on the equations defining the Bayesian model and thus on the lexicon and frequencies implicit in the current segmentation). Then the segmentation is updated by inserting or removing a boundary at *b* according to this probability, and the learner moves on to the remaining boundary locations. Pseudocode for this algorithm is shown in (6).

(6) Pseudocode for Gibbs sampler (Ideal Learner)

```
Randomly initialize all word boundaries in corpus
For i = 1 to number of iterations
      For each possible boundary location b in corpus
            (1) Compute p, the probability that b is a boundary (b = 1)
            given the current segmentation of the rest of the corpus
            (2) With probability p, set b to 1; else set b to 0
```

GGJ found that in order to converge to a good approximation of the posterior, the Gibbs sampler required 20000 iterations (i.e., each possible boundary in the corpus was sampled 20000 times), with $\alpha = 20$ for the unigram models, and $\beta = 10$, $\gamma = 3000$ for the bigram models.

Due to the convergence guarantees noted above, this algorithm is well-suited to the computational-level analysis that GGJ were interested in, allowing them to ask what kinds of segmentations would be learned by ideal learners with different assumptions about the nature of language. GGJ discovered that an ideal learner that is biased to heed context (the bigram model) achieves far more successful segmentation than one that is not (the unigram model). Moreover, a unigram ideal learner will severely undersegment the corpus, identifying common collocations as single words (e.g., *you want* segmented as *youwant*), most likely because the only way a unigram learner can capture strong word-to-word dependencies is to assume those words are actually a single word. This tells us about the expected behavior in learners who are able to make optimal use of their input – that is, what in principle are the useful biases for humans to use, given the available data.

Turning to the algorithmic level of analysis, however, the GGJ learner is clearly less satisfactory, since the Gibbs sampling algorithm requires the learner to store the entire corpus in memory, and also to perform a significant amount of processing (recall that each boundary in the corpus is sampled 20000 times). In the following section, we describe three algorithms that make more cognitively plausible assumptions about memory and processing. These algorithms will allow us to investigate how such memory and processing limitations might affect the learner's ability to achieve the optimal solution to the segmentation task (i.e., the solution found by the ideal learners in GGJ).

2.3.2. Constrained learners

To simulate limited resources, all the learning algorithms we present operate in an online fashion, so that processing occurs one utterance at a time rather than over the entire corpus simultaneously. Under GGJ's Bayesian model, the only information necessary to compute the probability of any particular segmentation of an utterance is the number of times each word (or bigram, in the case of the bigram model) has occurred in the model's current estimation of the segmentation. Thus, in each of our online learners, the lexicon counts are updated after processing each utterance (and in the case of one learner, during the processing of each utterance as well). The primary differences between our algorithms lie in the additional details of how resource limitations are implemented, and whether the learner is assumed to sample segmentations from the posterior distribution or choose the most probable segmentation.

2.3.2.1 Dynamic Programming Maximization

We first tried to find the most direct translation of the ideal learner to an online learner that must process utterances one at a time, such that the *only* limitation is that utterances must be processed one at a time. One idea for this is an algorithm we call Dynamic Programming Maximization (DPM), which processes each utterance as a whole, using dynamic programming (specifically the Viterbi algorithm) to efficiently compute the highest-probability segmentation of that utterance given the current lexicon.[2] It then adds the words from that segmentation to the lexicon and moves to the next utterance. This algorithm is the only one of our three that has been previously applied to word segmentation (Brent, 1999). Pseudocode for this learner is shown in (7).

(7) Pseudocode for DPM Learner

```
initialize lexicon (initially empty)
For u = 1 to number of utterances in corpus
      (1) Use Viterbi algorithm to compute the highest probability
      segmentation of utterance u, given the current lexicon
      (2) Add counts of segmented words to lexicon
```

2.3.2.2 Dynamic Programming Sampling

We then created a variant that is similar to DPM, but instead of choosing the most probable segmentation of each utterance conditioned on the current lexicon, it chooses a segmentation based on how probable that segmentation is. This algorithm, called Dynamic Programming Sampling (DPS), computes the probabilities of all possible segmentations using the forward pass of the forward-backward algorithm, and then uses a backward pass to sample from the distribution over segmentations. Pseudocode for this learner is shown in (8); the backward sampling pass is an application of the general method described in Johnson, Griffiths, and Goldwater (2007).

(8) Pseudocode for DPS learner

```
initialize lexicon (initially empty)
For u = 1 to number of utterances in corpus
      (1) Use Forward algorithm to compute probabilities of all
      possible segmentations of utterance u, given the current lexicon
      (2) Sample segmentation, based on probability of the segmentation
      (3) Add counts of segmented words to lexicon
```

2.3.2.3 Decayed Markov Chain Monte Carlo

We also examined a learning algorithm that recognizes that human memory decays over time and so focuses processing resources more on recent data than on data heard further in the past (a recency effect). We implemented this using a Decayed Markov Chain Monte Carlo (DMCMC) algorithm (Marthi et al., 2002), which processes an utterance by probabilistically sampling *s* word boundaries from all the utterances encountered so far. The sampling process is similar to Gibbs sampling, except that the learner only has the information available from the utterances encountered so far to inform its decision, rather than information derived from processing the entire corpus.

The probability that a particular potential boundary *b* is sampled is given by the exponentially decaying function b_a^{-d}, where b_a is the number of potential boundary locations between *b* and the end of the current utterance, and *d* is the decay rate. Thus, the further *b* is from the end of the current utterance, the less likely it is to be sampled. The exact probability is based on the decay rate *d*. For example, suppose *d* was 1, and there are 5 potential boundaries that have been encountered so far. The probabilities for sampling each boundary are shown in Table 1.

[Insert Table 1 approximately here: Likelihood of sampling a given boundary in DMCMC]

After each boundary sample is completed, the learner updates the lexicon. Pseudocode for this learner is shown in (9).

(9) Pseudocode for DMCMC learner

```
initialize lexicon (initially empty)
```

```
For u = 1 to number of utterances in corpus
      Randomly initialize word boundaries for utterance u.
      For s = 1 to number of samples to be taken per utterance
            (1) Probabilistically sample one potential boundary from
            utterance u or earlier, based on decay rate d (has bias to
            sample more recent boundaries) and decide whether a word
            boundary should be placed there
            (2) Update lexicon if boundary changed (inserted or
            deleted)
```

We note that one main difference between the DMCMC learner and the Ideal learner is that the Ideal learner samples every boundary from the corpus on each iteration, rather than being restricted to a certain number from the current utterance or earlier. The Ideal learner thus has knowledge of future utterances when making its decisions about the current utterance and/or previous utterances, while the DMCMC learner does not. In addition, restricting the number of samples in the DMCMC learner means that it requires less processing power than the Ideal learner.

We examined a number of different decay rates, ranging from 2.0 down to 0.125. To give a sense of what these really mean for the DMCMC learner, Table 2 shows the probability of sampling a boundary within the current utterance assuming the learner could sample a boundary from any utterances that occurred within the last 30 minutes of verbal interaction (i.e., this includes child-directed speech as well as any silences or pauses in the input stream). Calculations are based on samples from the alice2.cha file from the Bernstein corpus, where an utterance occurs on average every 3.5 seconds. As we can see, the lower decay rates cause the learner to look further back in time, and thus require the learners to have a stronger memory in order to successfully complete the boundary decision process.

[Insert Table 2 approximately here: Probability of sampling a boundary from the current utterance, based on decay rate]

The DMCMC learner has some similarity to previous work on probabilistic human memory, such as Anderson & Schooler (2000). Specifically, Anderson & Schooler argue for a rational model of human memory that calculates a "need" probability for accessing words, which is approximately how likely humans are to need to retrieve that word. The higher a word's need probability, the more likely a human is to remember it. The need probability is estimated based on statistics of the linguistic environment. Anderson & Schooler demonstrate that the need probability estimated from a number of sources, including child-directed speech, appears to follow a power law distribution with respect to how much time has elapsed since the word was last mentioned. Our DMCMC learner, when doing its constrained inference, effectively calculates a need probability for potential word boundaries – this is the sampling probability calculated for a given boundary, which is derived from an exponential decay function. Potential word boundaries further in the past are less likely to be needed for inference, and so are less likely to be retrieved by our DMCMC learner.

3. Bayesian Model Results

3.1 The data set

We tested the GGJ Ideal learner and our three constrained learners on data from the

Bernstein corpus (Bernstein-Ratner, 1984) from the CHILDES database (MacWhinney 2000). We used the phonemic transcription of this corpus that has become standard for testing word segmentation models (Brent, 1999).[3] The phonemically transcribed corpus contains 9790 child-directed speech utterances (33399 tokens, 1321 types, average utterance length = 3.4 words, average word length = 2.9 phonemes) See Table 3 for sample transcriptions and Appendix Figure 1 for the phonemic alphabet used. Unlike previous work, we used cross-validation to evaluate our models, splitting the corpus into five randomly generated training sets (~8800 utterances each) and separate test sets (~900 utterances each), where each training and test set were non-overlapping subsets of the data set used by GGJ. We used separate training and test sets to examine the modeled learner's ability to generalize to new data it has not seen before (and been iterating over, in the case of the Ideal learner). Specifically, we wanted to test if the lexicon the learner inferred was useful beyond the immediate dataset it trained on. Temporal order of utterances was preserved in the training and test sets, such that utterances in earlier parts of each set appeared before utterances in later parts of each set.

[insert Table 3 approximately here: Samples of Bernstein corpus.]

3.2 Performance measures

We assessed the performance of these different learners, based on precision and recall over word tokens, word boundaries, and lexicon items, where precision is # correct/# found and recall is # correct/# found in the gold standard. To demonstrate how these measures gauge performance differently, let us consider the evaluation of the utterances "*look at the doggie*" and "*look at the kitty*", which are translated into phoneme characters as "lUk &t D6 dOgi" and “lUk &t D6 kIti”. Suppose the algorithm decided the best segmentation was “lUk&t D6 dOgi” and “lUk&t D6kIti”. For word tokens, precision is 2/5, while recall is 2/8; for word boundaries (utterance-initial and utterance-final boundaries are excluded), precision is 3/3, while recall is 3/6; for lexicon items, precision is 2/4, while recall is 2/5.

3.3 Performance

Table 4 reports the scores for each Bayesian learner, along with results (where available) from other statistical learners discussed previously. F scores (F = (2 * precision * recall)/(precision + recall)) are also provided. Note that the results for the WordEnds, Bootstrap Voting Experts (BVE), and PHOCUS learners are not strictly comparable to our own results, since they do not use a separate training and test set. Instead, they trained over the entire Bernstein corpus and were evaluated by how well they segmented that corpus. In addition, PHOCUS performance was computed after discarding the first 1000 utterances (see Blanchard et al. (2010) for discussion as to why). The results for the transitional probability learner are from an implementation that computes transitional probabilities based on an initial sweep through the corpus (technically making it a batch algorithm; online versions are possible but would require some form of smoothing to avoid zero-probability transitions). The learner then inserts boundaries at each transitional probability minimum, as suggested by Saffran, Aslin, & Newport (1996). (One could segment instead at a fixed transitional probability threshold; we found no threshold that worked better than using the minimum.)[4]

As for the parameters of our own learners, all DMCMC learners have *s* = 20000 (20000 samples per utterance), as we found this gave the best segmentation performance. While this may still seem like a lot of processing, this learner nonetheless takes 89% fewer samples than the ideal learner in GGJ, which is a substantial savings in processing resources. In addition, the DMCMC unigram learners fared best with $d = 1.0$, while the DMCMC bigram learners fared best with $d = 0.25$. Figure 1 shows the F scores over word tokens for each of the Bayesian learners (both unigram and bigram variants) while Figure 2 shows the F scores over lexicon items.

[Insert Table 4 approximately here: Average performance of different learners on the test sets.]

[Insert Figure 1 approximately here: Word token F-scores for each of the learners]
[Insert Figure 2 approximately here: Lexicon item F-scores for each of the learners]

A few observations: First, while there is considerable variation in the performance of our constrained learners, all of them out-perform a transitional probability learner operating over phonemes (compare Bayesian learner results to TransProb learner results in Table 4). In addition, our best Bayesian learners compare favorably to other statistical learning algorithms, particularly with respect to their scores over lexicon items. For example, though the batch-learning WordEnds model achieves comparable boundary scores (BF) to many of our online Bayesian learners, its lexicon score (LF) is much lower than most of our online Bayesian learners (LF WordEnds = 36.6, LF all learners but Bigram DPS: 51.8 – 65.0). Similarly, while PHOCUS achieves comparable token scores to our best online Bayesian learner (TF PHOCUS: 75.8, TF Bigram DMCMC: 73.0), its lexicon score is lower (LF PHOCUS: 54.5, LF Bigram DMCMC: 62.6). Thus, our online Bayesian learners seem better able to extract a reliable lexicon from the available data than other recent statistical learners, including one (PHOCUS) that relies on domain-specific knowledge about word well-formedness.

Second, when we examine the impact of the unigram and bigram assumptions on word token performance, we find that the bigram learners do not always benefit from assuming words are predictive of other words. While the Ideal, DPM and DMCMC learners do (bigram F > unigram F, Ideal: $p < .001$, DPM: $p = .046$, DMCMC: $p = .002$), the DPS learner is harmed by this bias (unigram F > bigram F: $p < .001$). This is also true for the lexicon F scores: While the Ideal and DPM learners are helped (bigram F > unigram F, Ideal: $p < .001$, DPM: $p = .002$), the DPS and DMCMC learners are harmed (unigram F > bigram F, DPS: $p < .001$, DMCMC: $p = .006$).[5]

Third, when comparing our ideal learner to our constrained learners, we find – somewhat unexpectedly – that some of our constrained learners are performing equivalently or *better* than their ideal counterparts. For example, when we look at word token F-scores for our bigram learners, the DMCMC learner seems to be performing equivalently to the Ideal learner (DMCMC ≠ Ideal: $p = 0.144$). Among the unigram learners, our DPM and DMCMC learners are equally out-performing the Ideal learner (DPM > Ideal: $p < .001$, DMCMC > Ideal: $p < .001$, DPM ≠ DMCMC: $p = 0.153$) and the DPS is performing equivalently to the Ideal learner (Ideal ≠ DPS: $p = 0.136$). Turning to the lexicon F-scores, the results look a bit more expected for the bigram learners: The Ideal learner is out-performing the constrained learners (Ideal > DPM: $p < .001$, Ideal > DPS: $p < .001$, Ideal > DMCMC: $p < .001$). However, among the unigram learners we again find something unexpected: the DMCMC learner is out-performing the Ideal learner (DMCMC > Ideal: $p = .006$). The Ideal learner is still out-performing the other two constrained learners, however (Ideal > DPM: $p = .031$, Ideal > DPS: $p < .001$).

Fourth, GGJ found that both their ideal learners tended to undersegment (putting multiple words together into one word), though the unigram learner did so more than the bigram learner (see Table 5 for examples).

[insert Table 5 approximately here: GGJ ideal model performance: Unigram vs. Bigram]

One way to gauge whether undersegmentation is occurring is to look at the boundary precision and recall scores. When boundary precision is higher than boundary recall, undersegmentation is occurring; when the reverse is true, the model is oversegmenting (splitting single words into more than one word). If we examine Table 4, we can see that (as found by GGJ) the Ideal learners are undersegmenting, with the bigram model doing so less than the unigram model. Looking at our constrained learners, we can see that the unigram DMCMC learner is also undersegmenting. However, every other constrained model is oversegmenting, with the DPS learners being the most blatant oversegmenters; the bigram DMCMC learner appears to be oversegmenting the least.

We also examined performance on the first and last words in utterances, as compared to performance over the entire utterance, based on work by Seidl & Johnson (2006) who found that 7-month-olds are better at segmenting words that are either utterance-initial or utterance-final (see Seidl & Johnson (2006) for detailed discussion on why this might be). If our models are reasonable reflections of human behavior, we hope to find that their performance on the first and last words is better than their performance over the entire utterance. Moreover, they should perform equally on the first and last words in order to match infant behavior. Figures 3 and 4 show word token F-scores for unigram and bigram learners, respectively, for whole utterances, first words, and last words. Table 6 shows the significance test scores for comparing first word, last word, and whole utterance performance for each of the learners.

[Insert Figure 3 approximately here: Performance of Bayesian unigram learners on whole utterances, first words, and last words]

[Insert Figure 4 approximately here: Performance of Bayesian bigram learners on whole utterances, first words, and last words]

[Insert Table 6 approximately here: Significance test scores]

Looking first to the Bayesian unigram learners, we find that the DPM and DMCMC learners match infant behavior best by improving equally on first and last words, compared to whole utterances. The Ideal learner improves on both first and last words, but improves more for last words than for first words, making its performance slightly different than infants'. The DPS learner only achieves better performance for first words, making its performance even more different from infants'. Turning to the Bayesian bigram learners, we find that only the DPM and DPS learners are matching infants by improving equally on first and last word performance, compared to whole utterance performance. Both the Ideal and DMCMC learners only improve for first words, and not for last words.

4. Discussion

Through these simulations, we have made several interesting discoveries. First, though none of our constrained learners out-performed the best ideal learner (the bigram learner) on all measures, our constrained learners still were able to extract statistical information from the available data well enough to out-perform learners that segment by tracking transitional probability. Since transitional probability strategies have historically been strongly associated with the idea of "cognitively plausible statistical learning" in models of human language acquisition (e.g., Saffran et al., 1996; Saffran, 2001; Perruchet & Desaulty, 2008; Pelucchi, Hay, & Saffran, 2009), our result underscores how statistical learning can be considerably more successful than is sometimes thought when only transitional probability learners are considered. In addition, our online Bayesian learners also out-performed several recent statistical models of word segmentation with respect to identifying a reliable lexicon, while performing comparably at token and word boundary identification. Our results suggest that even with limitations on memory and processing, a learning strategy that focuses explicitly on identifying words in the input and optimizing a lexicon (as all our learners here do) may work better than one that focuses on identifying boundaries (as transitional probability learners and some recent statistical learning models do).

Second, we discovered that a bias that was helpful for the ideal learner – to assume words are predictive units – is not always helpful for constrained learners. This suggests that we must be careful in transferring the solutions we find for ideal learners to learners who have constraints on their memory and processing the way that humans do. In this case, we speculate that the reason some of our constrained learners do not benefit from the bigram assumption has to do with the algorithm's ability to search the hypothesis space; when tracking bigrams instead of just individual words, the learner's hypothesis space is much larger. It may be that some constrained learners do not have sufficient processing resources to find the optimal solution (and perhaps to recover from mistakes made early on). However, not all constrained learners suffer from this. There were constrained learners that benefited from the bigram assumption, which suggests less processing power may be required than previously thought to converge on good word segmentations. In particular, if we examine the DMCMC learner, we can decrease the number of samples per utterance to simulate a decrease in processing power. Table 7 shows the F-scores by word tokens for both the unigram and bigram DMCMC learner with varying samples per utterance. Though performance does degrade when processing power is more limited, these learners still out-perform the best phonemic transition probability learner variant we identified (which had scores around 38 for word tokens), even when sampling only 0.057% as much as the ideal learner. Moreover, the bigram assumption continues to be helpful, even with very little processing power available for the DMCMC learner.

[Put Table 7 approximately here: Performance on test set 1 for DMCMC learners]

If we constrain the ideal learner so it can only sample as often as the DMCMC learner does, we find that its performance is not nearly as good as the DMCMC learner (see Table 8). The ideal bigram learner, in particular, suffers noticeably. This suggests that there is something significant gained by the DMCMC learner's method of approximated inference (to be discussed further below).

[Put Table 8 approximately here: Performance on test set 1 for DMCMC learners and ideal learners that only sample as much as the DMCMC learners do.]

Turning to the more general comparison of the ideal learner to the constrained learners, we made a surprising discovery – namely that some of our constrained unigram learners out-performed the ideal learner. This is somewhat counterintuitive, as one might naturally assume that less processing power would lead to equivalent if not worse performance.

To rule out the possibility that these results are an artifact of this particular corpus, we tested our learners on a larger corpus of English, the Pearl-Brent derived corpus available through CHILDES (MacWhinney 2000). This corpus contains child-directed speech to children between 8 months and 9 months old, consisting of 28,391 utterances (96,920 word tokens, 3,213 word types, average words per utterance: 3.4, average phonemes per word: 3.6). In Table 9, we report the learners' performance on five test sets generated from this corpus (these were generated the same way as the ones from the Bernstein-Ratner corpus were). The same surprising performance trend appears, where the DMCMC unigram learner is out-performing the Ideal unigram learner – though only with respect to tokens and word boundaries, and not with respect to lexicon items.

[Insert Table 9 approximately here: Average performance of different learner on five test sets from the Pearl-Brent derived corpus.]

We subsequently looked at the errors being made by both the ideal and the DMCMC unigram learners on these English corpora, and discovered a potential cause for the surprising behavior. It turns out that the ideal learner makes many more undersegmentation errors on highly frequent bigrams consisting of short words (e.g., *can you, do you, it's a*) while the DMCMC learner does not undersegment these bigrams. When the DMCMC learner does make errors on frequent items that are different from the errors the ideal learner makes, it tends to oversegment (e.g., *doggies* segmented as *doggie s*). If we survey the errors made by each learner for items occurring 7 or more times in the first test set of each English corpus and which are not shared (i.e., only one learner made the error), we find the DMCMC learner's additional errors are far fewer than the ideal learner's additional errors (as show below in Table 10).

[Insert Table 10 approximately here: Analysis of unshared errors made by the ideal and DMCMC unigram learners for items occurring 7 or more times in the first test set of each corpus.]

A possible explanation for this error pattern is related to the ideal learner's increased processing capabilities. Specifically, the ideal learner is granted the memory capacity to survey the entire corpus for frequency information and update its segmentation hypotheses for utterances occurring early in the corpus at any point during learning. This allows the ideal unigram learner to notice that certain short items (e.g., actual words like *it's* and *a*) appear very frequently together. Given that it cannot represent this mutual occurrence any other way, it will decide to make these items a single lexical item; moreover, it can fix its previous "errors" that it made earlier during learning when it thought these were two separate lexical items. In contrast, the DMCMC learner does not have this omniscience about item frequency in the corpus, nor as much ability to fix "errors" made earlier in the learning process. This results in the DMCMC learner leaving these short items as separate, particularly when encountered in earlier utterances. As they then continue to exist in the lexicon as separate lexical items, undersegmentation errors do not occur nearly as much.

In summary, more processing power and memory capacity does appear to hurt the inference process of the ideal unigram learner. This behavior is similar to Newport (1990)'s

"Less is More" hypothesis for human language acquisition, which proposes that limited processing abilities are advantageous for tasks like language acquisition because they selectively focus the learner's attention. With this selective focus, children are better able to home in on the correct components for language since they do not consider as much complex information. Transferring this idea to our unigram learners, the more limited inference process of the DMCMC learner focuses its attention only on the current frequency information and does not allow it to view the frequency of the corpus as a whole. Coupled with this learner's more limited ability to correct its initial hypotheses about lexicon items, this leads to superior segmentation performance. We note, however, that this superior performance is mainly due to the unigram learner's inability to capture word sequence predictiveness; when it sees items appearing together, it has no way to capture this behavior except by assuming these items are actually one word. Thus, the ideal unigram learner's additional knowledge causes it to commit more undersegmentation errors. The bigram learner, on the other hand, does not have this problem – and indeed we do not see the DMCMC bigram learner out-performing the ideal bigram learner.

Turning to general undersegmentation behavior, we also discovered that the tendency to undersegment the corpus depends on how constraints are implemented in our learners, as well as whether the learners assume words are predictive or not. According to Peters (1983), English children tend to make errors that indicate undersegmentation rather than oversegmentation, so perhaps learners that undersegment are a better match for children's behavior. Here, the Bayesian learners that undersegmented on the English data were both of the ideal learners as well as the unigram DMCMC learner.

Another finding is that models differ on their ability to match infant word segmentation behavior at utterance edges (the first word and the last word). Some of our constrained models are in fact better able to match infant behavior on this measure than our ideal models. Seidl and Johnson (2006) review a number of proposed explanations of why utterance edges are easier, including perceptual/prosodic salience, cognitive biases to attend more to edges (including recency effects), or the pauses at utterance boundaries. In our results, we find that all of the models find utterance-initial words easier to segment, and most of them also find utterance-final words easier. Since none of the algorithms include models of perceptual salience, our results suggest that this explanation is probably unnecessary to account for the edge effect, especially for utterance-initial words. Rather, it seems simpler to assume that the pauses at utterance boundaries make segmentation easier by eliminating the ambiguity of one of the two boundaries of the word.

However, if this were the only effect at utterance edges, then we would expect all of our models to find both initial and final words easier. In fact, some of them, including the ideal bigram learner, find only initial words easier. This finding suggests that some other statistical property of final words actually make them more difficult than initial words for (at least some) purely statistical learners. For example, the words and phrases that end sentences (often nouns or verbs) may be more variable or infrequent than the words that start sentences (often pronouns or determiners). Since utterance-final words seem to be at least as easy for infants as utterance-initial words, a recency effect could be playing an important role here. However, in light of the varying results of the different models, further analysis of the statistical properties of utterance-final versus utterance-initial words is warranted before drawing any strong conclusions.

5. Conclusions & Future Work

One moral of this investigation is that a simple intuition about human cognition, such as having memory and processing limitations, can be cashed out multiple ways in online learning algorithms. Here, we examined limitations such as processing utterances incrementally and implementing recency effects with exponential decay functions. Having explored several learner instantiations incorporating this intuition, we find that the learning assumptions or biases that work best depend on how limitations are implemented. And in fact, some biases that are helpful for an ideal learner, such as using context to guide hypotheses, may hinder a constrained learner with more limited memory and processing resources. On a related note, if the learner does not use word context, having less memory and processing resources may in fact be beneficial.

We view these investigations as a first step towards understanding how to translate computational-level solutions into algorithmic-level ones for language acquisition, as there are clearly other ways of implementing constrained algorithms. It is also useful to ask if the effects discovered here are robust, and persist across different languages. Moreover, we can take further inspiration from what is known about the representations infants attend to, and allow our algorithms to have knowledge of syllables (Jusczyk et al. 1999a), to track stressed and unstressed phonemes/syllables separately (Curtin, Mintz, & Christiansen (2005), Pelucchi, Hay, & Saffran (2009)), and to have additional prior phonotactic knowledge argued to be universal in human language (e.g., Blanchard et al. (2010)).

The transition from a computational-level solution for an acquisition problem to the algorithmic-level approximation may not necessarily be straightforward. By integrating what we know of the human ability to utilize available statistical information with what we know of human limitations, we can come to understand how infants accomplish the things they do.

Appendix. Phoneme encoding of corpus.

[Put Appendix Figure 1 approximately here: Phoneme encoding.]

Acknowledgements.

We would like to thank the audiences at the PsychoComputational Models of Human Language workshop in 2009, BUCLD 34, three anonymous reviewers, Alexander Clark, William Sakas, Tom Griffiths, and Michael Frank. We would also like to give a special thanks to Jim White for his insight about the differences in performance between the ideal and online Bayesian learners. This work was supported by NSF grant BCS-0843896 to the first author and CORCL grant MI 14B-2009-2010 to the first and third authors.

End Notes

[1] Note that the units over which these models operate are phonemes, rather than phonetic features (Christiansen et al., 1998) or syllables (Swingley, 2005). This is not uncontroversial, as it makes the model insensitive to feature-based similarity between sounds and abstracts away from many details of phonetic and acoustic variation. However, it was chosen based on the available input corpora, and to facilitate comparison with other word segmentation models.

[2] Technically, the probabilities computed by the Viterbi algorithm (and the forward algorithm used by the DPS model) are only an approximation of the true probabilities of the segmented utterances, since they are based on the contents of the lexicon not including any of the words in the current utterance. Equation (2) shows that in the true posterior, the words segmented at the beginning of the utterance will affect the probabilities of the words at the end of the utterance.

[3] We note that the statistical strategies we explore here are meant to be an initial bootstrapping method for children to break into word segmentation – as such, we believe testing these strategies on a corpus this size is not unreasonable. However, see additional results from a larger English corpus in the Discussion section that follow the same trends observed in the Bernstein corpus.

[4] We note that our transitional probability learner achieves comparable or better performance compared to other reported transitional probability learner implementations: The one in Brent (1999) operated over phonemes and had token F-scores in the 40s and lexicon precision near 15, while the one in Gambell & Yang (2006) operated over syllables and had token F-scores near 30.

[5] All p-values reported above and below were calculated by comparing 5 runs of each of the mentioned learners against each other in a two-tailed t-test analysis.

References

Anderson, J.R., & Schooler, L.J. (2000). The adaptive nature of memory. In E. Tulving & F.I.M. Craik (Eds.), *The Oxford Handbook of Memory*, 557-570. Oxford University Press.

Bernstein-Ratner, N. (1984). Patterns of vowel Modification in motherese. *Journal of Child Language. 11,* 557-578.

Blanchard, D., Heinz, J., & Golinkoff, R. (2010). Modeling the contribution of phonotactic cues to word segmentation. *Journal of Child Language, 27,* 487-511.

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

Christiansen, M., Allen, J., and Seidenberg, M. (1998). Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes, 13,* 221-268.

Curtin, S., Mintz, T., & Christansen, M. (2005). Stress changes the representational landscape: evidence from word segmentation in infants. *Cognition*, *96*, 233-262.

Ferguson, T. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics, 1*, 209-230.

Fleck, M. (2008). Lexicalized phonotactic word segmentation. *Proceedings of the Association for Computational Linguistics,* 130-138.

Frank, M. C., Goodman, N. D., & Tenenbaum, J. (2009). Using speakers' referential intentions to model early cross-situational word learning. *Psychological Science, 20*, 579-585.

Gambell, T. & Yang, C. (2006). Word Segmentation: Quick but not Dirty. Manuscript. Yale University.

Griffiths, T. L., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. B. (2010). Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive Sciences, 14, 357-364.*

Griffiths, T. L., Kemp, C., & Tenenbaum, J. B. (2008). Bayesian models of cognition. In Ron Sun (ed.), *The Cambridge handbook of computational cognitive modeling*. Cambridge University Press.

Griffiths, T. L., & Tenenbaum, J. B. (2005). Structure and strength in causal induction. *Cognitive Psychology, 51,* 354-384.

Goldwater (2006). *Nonparametric Bayesian models of lexical acquisition*. Ph.D. thesis, Brown University.

Goldwater, S., Griffiths, T., and Johnson, M. (2007). Distributional Cues to Word Boundaries: Context is Important, In Caunt-Nulton, H., Kulatilake, S., and Woo, I. (eds), *BUCLD 31: Proceedings of the 31st annual Boston University Conference on Language Development,* Somerville, MA: Cascadilla Press, 239-250.

Hewlett, D. & Cohen, P. (2009). Bootstrap voting experts. *Proceedings of the twenty-first international joint conference on artificial intelligence (IJCAI-09)*, 1071-1076. Available at http://ijcai.org/papers09/contents.php.

Jackson-Maldonado, D. & Thal, D. (1993). Lenguaje y Cognición en los Primeros Años de Vida. Project funded by the John D. and Catherine T. Mac Arthur Foundation and CONACYT, Mexico.

Johnson, E. & Jusczyk, P. (2001). Word segmentation by 8-month-olds: When speech cues count more than statistics. *Journal of Memory and Language, 44*, 548-567.

Johnson, M., Griffiths, T., and Goldwater, S. (2007). Bayesian inference for PCFGs via Markov Cain Monte Carlo. *Proceedings of the meeting of the North American Association for Computational Linguistics.*

Jusczyk, P., Goodman, M., and Baumann, A. (1999a). Nine-month-olds' attention to sound similarities in syllables, *Journal of Memory & Language, 40*, 62–82.

Jusczyk, P., Hohne, E., and Baumann, A. (1999b) Infants' sensitivity to allophonic cues for word segmentation. *Perception and Psychophysics, 61*, 1465-1476.

Juszcyk, P., Houston, D., and Newsome, M. (1999c). The beginnings of word segmentation in English-learning infants. *Cognitive Psychology, 39,* 159-207.

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk.* Mahwah, NJ: Lawrence Erlbaum Associates.

Marr, D. (1982). *Vision*. San Francisco: Freeman.

Marthi, B., Pasula, H., Russell, S, & Peres, Y. et al. 2002. Decayed MCMC Filtering. In *Proceedings of 18th UAI*, 319-326.

Mattys, S., Jusczyk, P., Luce, P., and Morgan, J. (1999). Phonotactic and prosodic effects on word segmentation in infants. *Cognitive Psychology, 38*, 465-494.

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T.T., Seidenberg, M. S., and Smith, L. B. (2010). Letting Structure Emerge: Connectionist and Dynamical Systems Approaches to Understanding Cognition. *Trends in Cognitive Sciences, 14,*, 348-356.

Morgan, J., Bonamo, K., and Travis, L. (1995). Negative evidence on negative evidence. *Developmental Psychology, 31*, 180-197.

Newport, E. (1990). Maturational constraints on language learning. *Cognitive Science, 14,* 11-28.

Oaksford, M. & Chater, N. (1998). *Rational models of cognition*. Oxford, England: Oxford University Press.

Pelucchi, B., Hay, J., & Saffran, J. (2009). Learning in reverse: Eight-month-old infants track backward transitional probabilities. *Cognition, 113,* 244-247.

Perruchet, P., & Desaulty, S. (2008). A role for backward transitional probabilities in word segmentation? *Memory and Cognition, 36,* 1299–1305.

Peters, A. (1983). *The Units of Language Acquisition, Monographs in Applied Psycholinguistics,* New York: Cambridge University Press.

Saffran, J., Aslin, R., & Newport, E. (1996). Statistical learning by 8-month-olds. *Science*, *274*, 1926-928.

Saffran, J. R. (2001). The use of predictive dependencies in language learning. *Journal of Memory and Language, 44,* 493–513.

Sanborn, A. N., Griffiths, T. L., & Navarro, D. J. (in press). Rational approximations to rational models: Alternative algorithms for category learning. *Psychological Review*.

Seidl, A. & Johnson, E. (2006). Infant word segmentation revisited: edge alignment facilitates target extraction. *Developmental Science, 9:6,* 565-573.

Shi, L., Griffiths, T. L., Feldman, N. H., & Sanborn, A. N. (in press). Exemplar models as a mechanism for performing Bayesian inference. *Psychonomic Bulletin & Review*.

Swingley, D. (2005). Statistical clustering and contents of the infant vocabulary. *Cognitive Psychology, 50,* 86-132.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association, 101(476),* 1566-1581.

Tenenbaum, J. & Griffiths, T. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences, 24,* 629-641.

Tenenbaum, J., Griffiths, T., & Kemp, C. (2006). Theory-based models of inductive learning and reasoning. *Trends in Cognitive Sciences, 10,* 309-318.

Thiessen, E., & Saffran, J. R. (2003). When cues collide: Use of stress and statistical cues to word boundaries by 7- to 9-month-old infants. *Developmental Psychology, 39*, 706–716.

Xu, F. & Tenenbaum, J. B. (2007). Word learning as Bayesian inference. *Psychological Review*, *114*, 245-272.

Figure 1. Word token F-scores for each of the learners, averaged over the test sets.

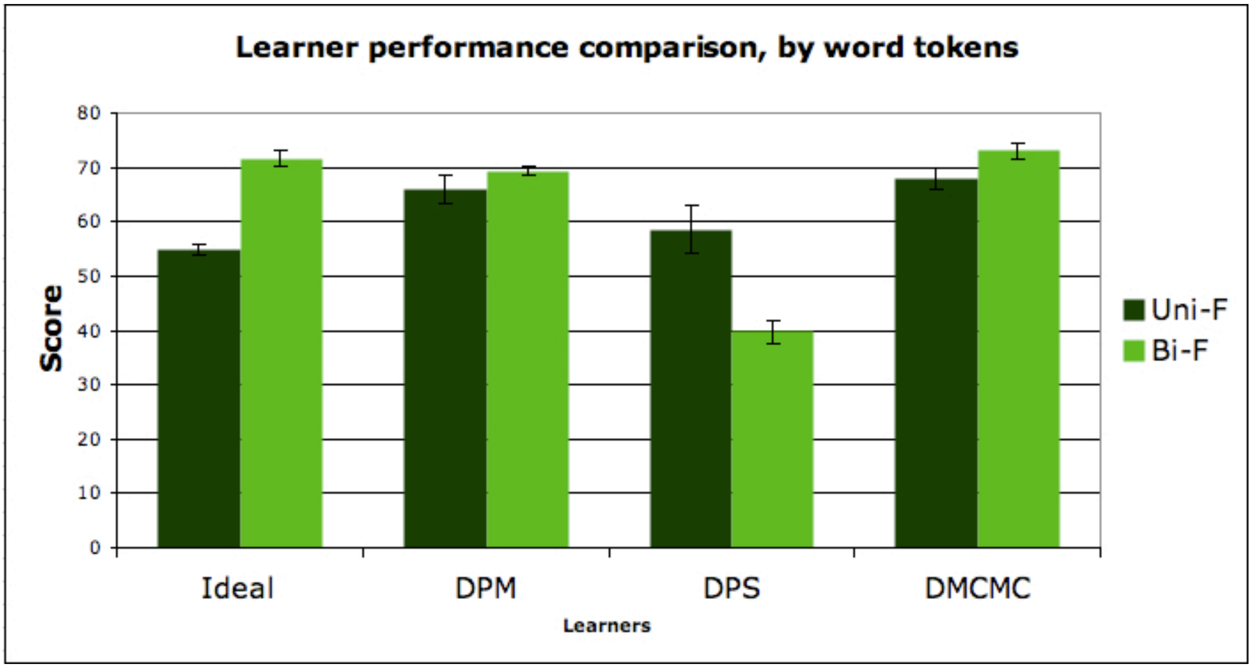


Figure 2. Lexicon item F-scores for each of the learners, averaged over the test sets.

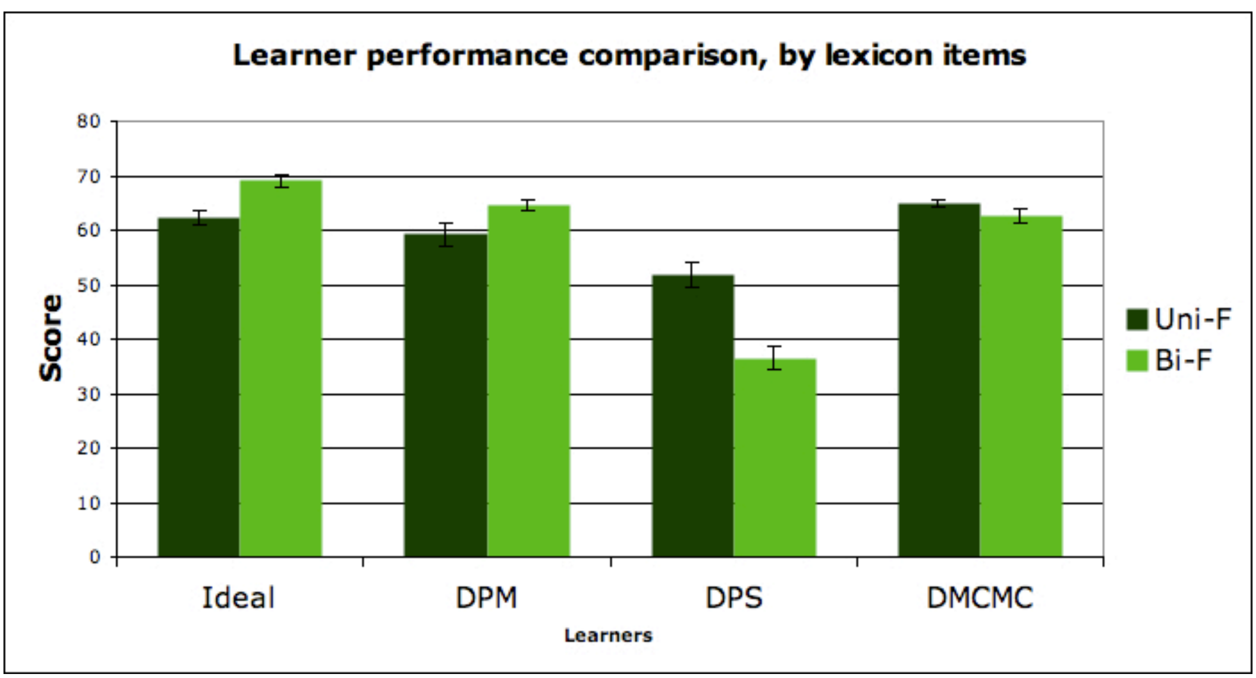


Figure 3. Performance of unigram learners on whole utterances, first words, and last words.

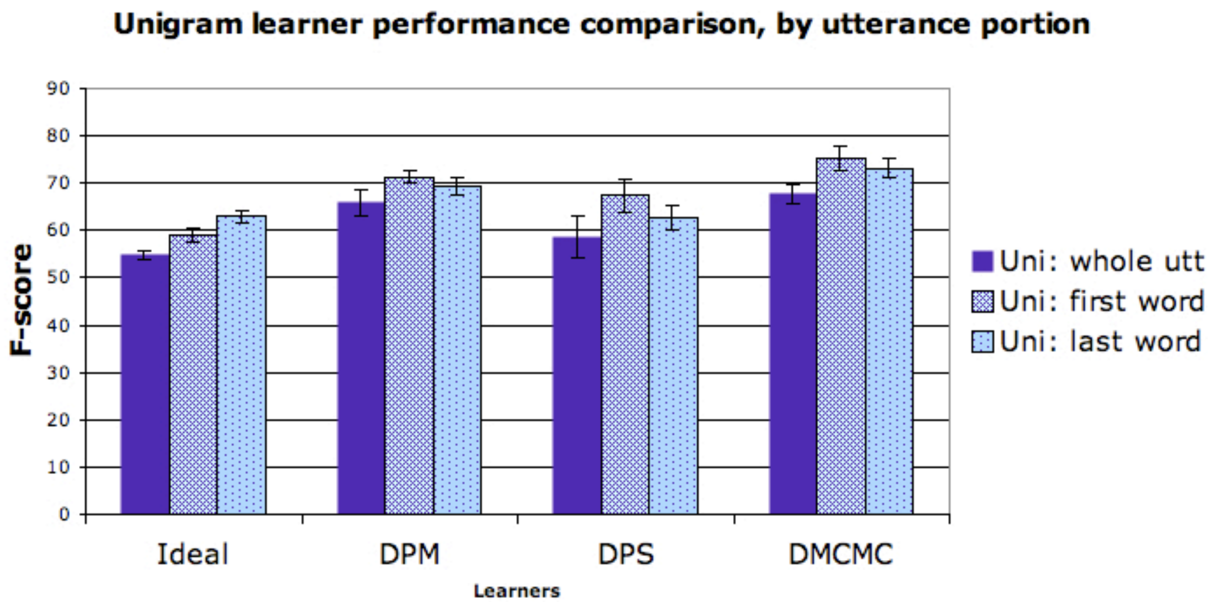
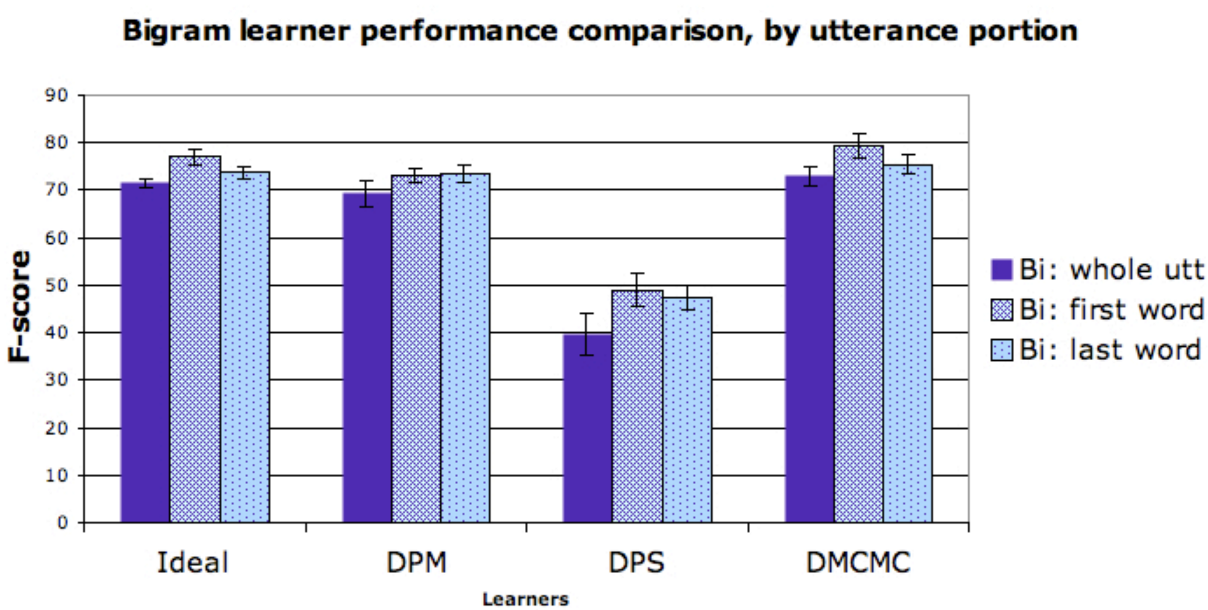


Figure 4. Performance of bigram learners on whole utterances, first words, and last words.



Appendix Figure 1. Phoneme encoding. Taken with permission from Goldwater, Griffiths, & Johnson (2007).

Consonants

ASCII	Ex.	ASCII	Ex.
D	THe	h	Hat
G	Jump	k	Cut
L	bottLe	l	Lamp
M	rhythM	m	Man
N	siNG	n	Net
S	SHip	p	Pipe
T	THin	r	Run
W	WHen	s	Sit
Z	aZure	t	Toy
b	Boy	v	View
c	CHip	w	We
d	Dog	y	You
f	Fox	z	Zip
g	Go	~	buttON

Vowels

ASCII	Ex.
&	thAt
6	About
7	bOY
9	flY
A	bUt
E	bEt
I	bIt
O	lAW
Q	bOUt
U	pUt
a	hOt
e	bAY
i	bEE
o	bOAt
u	bOOt

Rhotic Vowels

ASCII	Ex.
#	ARe
%	fOR
(	hERE
)	lURE
*	hAIR
3	bIRd
R	buttER

Table 1. Likelihood of sampling a given boundary in DMCMC, $d = 1$. The relative probability of a given boundary being sampled is the decay probability b_a^{-d} divided by the sum of the decay probabilities for all boundary positions under consideration (in this example, five boundary positions).

Boundary position	b_a^{-d}	Relative probability
end – 1	$1^{-1} = 1/1 = 1.00$	1.00/(Σ(probs)) = 0.44
end – 2	$2^{-1} = 1/2 = 0.50$	0.50/(Σ(probs)) = 0.22
end – 3	$3^{-1} = 1/3 = 0.33$	0.33/(Σ(probs)) = 0.15
end – 4	$4^{-1} = 1/4 = 0.25$	0.25/(Σ(probs)) = 0.11
end – 5	$5^{-1} = 1/5 = 0.20$	0.20/(Σ(probs)) = 0.08

Table 2. Probability of sampling a boundary from the current utterance, based on decay rate.

decay rate	Probability of sampling within current utterance
2.0	0.942
1.5	0.772
1.0	0.323
0.75	0.125
0.50	0.036
0.25	0.009
0.125	0.004

Table 3. Samples of Bernstein corpus.

English orthography	Phonemic transcription
you want to see the book	yu want tu si D6 bUk
look there's a boy with his hat	lUk D*z 6 b7 wIT hIz h&t
and a doggie	&nd 6 dOgi
you want to look at this	yu want tu lUk &t DIs

Table 4. Average performance of different learners on the five test sets, along with published results from other recent statistical learners where available and the results from a transitional probability learner. Note that the PHOCUS results are from the "3s" implementation, which performed best on the corpus. Precision (P), recall (R), and F-score (F) over word tokens (T), word boundaries (B), and lexicon items (L) resulting from the chosen word segmentation are shown. Standard deviations are shown in parentheses where available.

Bayesian Unigram Learners (words are not predictive)									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
GGJ-Ideal	63.2 (0.99)	48.4 (0.80)	54.8 (0.85)	92.8 (0.67)	62.1 (0.42)	74.4 (0.42)	54.0 (0.92)	73.6 (1.89)	62.3 (1.30)
DPM	63.7 (2.82)	68.4 (2.68)	65.9 (2.73)	77.2 (1.86)	85.3 (1.67)	81.0 (1.64)	61.9 (2.17)	56.9 (2.07)	59.3 (2.09)
DPS	55.0 (4.82)	62.6 (3.99)	58.5 (4.45)	70.4 (3.73)	84.21 (1.79)	76.7 (2.85)	54.8 (1.64)	49.2 (3.14)	51.8 (2.2)
DMCMC	71.2 (1.57)	64.7 (2.31)	67.8 (1.97)	88.8 (0.89)	77.2 (2.17)	82.6 (1.53)	61.0 (1.18)	69.6 (0.43)	65.0 (0.67)
Bayesian Bigram Learners (words are predictive)									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
GGJ-Ideal	74.5 (1.41)	68.8 (1.53)	71.5 (1.46)	90.1 (0.75)	80.4 (1.01)	85.0 (0.82)	65.0 (1.19)	73.5 (1.71)	69.1 (1.15)
DPM	67.5 (1.13)	71.3 (0.74)	69.4 (0.90)	80.4 (0.96)	86.8 (0.63)	83.5 (0.57)	66.0 (1.00)	63.2 (1.46)	64.5 (1.05)
DPS	34.2 (2.16)	47.6 (2.16)	39.8 (2.13)	54.9 (1.40)	85.3 (2.07)	66.8 (1.00)	39.0 (2.02)	34.4 (2.42)	36.5 (2.19)
DMCMC	72.0	74.0	73.0	84.1	87.4	85.7	61.1	64.2	62.6

	(1.24)	(1.76)	(1.43)	(0.98)	(1.47)	(0.94)	(1.41)	(1.35)	(1.17)
Comparison Learners									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
WordEnds			70.7	94.6	73.7	82.9			36.6
BVE	79.1	79.4	79.3	92.8	90.5	91.6			
PHOCUS	77.7	74.0	75.8	89.7	83.6	86.5	47.3	64.0	54.5
TransProb	34.3 (0.88)	42.7 (0.83)	38.0 (0.87)	52.8 (1.22)	71.1 (1.00)	60.6 (1.15)	24.3 (0.55)	39.7 (1.1)	30.1 (0.70)

Table 5. GGJ ideal learner model performance: Unigram vs. Bigram. Segmentations are shown in their English orthographic form, and undersegmentations are italicized.

Unigram Model	Bigram Model
youwant to see *thebook*	you want to see the book
look theres *aboy*	look theres a boy
with his hat	with his hat
and *adoggie*	and a doggie
you *wantto lookatthis*	you want to *lookat* this
lookatthis	*lookat* this
havea drink	have a drink
okay now	okay now
whatsthis	whats this
whatsthat	whats that
whatisit	*whatis* it
look *canyou* take *itout*	look *canyou* take it out

Table 6. Significance test scores (two tailed t-test) for comparisons between first word, last word, and whole utterance performance across the five test sets. Non-significant differences are italicized.

	first ≠ whole	last ≠ whole	last ≠ first
Unigram Models (Words are not predictive)			
GGJ – Ideal	.001	< .001	.003
DPM	.008	.046	*.108*
DPS	.008	*.130*	.038
DMCMC	< .001	.003	*.207*
Bigram Models (Words are predictive)			
GGJ – Ideal	< .001	*.157*	.050
DPM	< .001	.005	*.730*
DPS	< .001	.003	*.372*
DMCMC	.002	*.069*	.029

Table 7. Performance on test set 1 for DMCMC learners with varying samples per utterance. Learners were tested with the decay rate that yielded the best performance at 20000 samples per utterance (unigram = 1, bigram = 0.25). F-scores over word tokens are shown, as well as the processing comparison to the ideal learner (as measured by number of samples taken).

s	20000	10000	5000	2500	1000	500	250	100
% Ideal learner samples	11.0	5.7	2.8	1.4	0.57	0.28	0.14	0.057
Unigram, $d = 1$	69.3	68.5	65.5	63.5	63.4	60.0	56.9	51.1
Bigram, $d = 0.25$	74.9	71.8	68.3	66.1	64.6	61.2	59.9	60.9

Table 8. Performance on test set 1 for DMCMC learners and ideal learners that only sample approximately as much as the DMCMC learners do. DMCMC learners sampled 20000 times per utterance with decay rate = 1 for the Unigram learner and 0.25 for the Bigram learner. Ideal learners made 2000 iterations over the corpus, sampling every potential boundary once each iteration.

Unigram Learners (words are not predictive)									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
GGJ-Ideal	49.5	44.6	46.9	71.4	61.2	65.9	34.1	51.7	41.1
DMCMC	72.1	66.8	69.3	88.3	79.1	83.4	62.8	69.8	66.1
Bigram Learners (words are predictive)									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
GGJ-Ideal	29.9	35.2	32.3	50.3	63.0	56.0	25.5	48.7	33.4
DMCMC	73.9	76.0	74.9	85.2	88.7	86.9	63.2	64.2	63.7

Table 9. Average performance of different learners on five test sets from the Pearl-Brent derived corpus. Precision (P), recall (R), and F-score (F) over word tokens (T), word boundaries (B), and lexicon items (L) resulting from the chosen word segmentation are shown. Standard deviations are shown in parentheses.

Unigram Models (words are not predictive)									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
GGJ-Ideal	62.4 (0.52)	48.1 (0.67)	54.3 (0.62)	92.0 (0.33)	62.1 (0.53)	74.2 (0.39)	50.0 (0.76)	69.9 (1.10)	58.3 (0.84)
DPM	53.6 (3.15)	66.1 (2.19)	59.2 (2.79)	66.7 (2.37)	88.5 (0.54)	76.1 (1.61)	60.9 (1.79)	38.5 (1.70)	47.2 (1.81)
DPS	46.3 (5.48)	61.6 (3.66)	52.8 (4.87)	60.9 (4.61)	89.5 (1.33)	72.4 (3.10)	51.4 (3.15)	28.5 (4.22)	36.6 (4.29)
DMCMC	67.5 (1.71)	61.0 (3.92)	64.1 (2.80)	86.3 (1.24)	74.5 (4.08)	79.9 (1.96)	53.8 (3.11)	61.0 (2.47)	57.2 (2.82)
Bigram Models (words are predictive)									
	TP	**TR**	**TF**	**BP**	**BR**	**BF**	**LP**	**LR**	**LF**
GGJ-Ideal	70.4 (1.03)	68.3 (0.75)	69.4 (0.89)	85.6 (0.78)	82.0 (0.31)	83.7 (0.53)	60.5 (0.98)	65.5 (0.67)	62.9 (0.80)
DPM	61.9 (1.58)	70.3 (0.97)	65.9 (1.27)	75.2 (1.16)	89.6 (0.83)	81.7 (0.61)	61.0 (0.79)	48.9 (1.01)	54.3 (0.68)
DPS	32.3 (4.99)	48.4 (4.73)	38.7 (5.10)	52.8 (3.47)	90.5 (0.95)	66.6 (2.63)	37.6 (1.62)	23.7 (1.69)	29.1 (1.74)
DMCMC	69.2 (1.19)	73.1 (0.96)	71.1 (1.08)	81.1 (0.89)	87.6 (0.49)	84.2 (0.69)	52.7 (1.41)	53.0 (1.37)	52.8 (1.34)

Table 10. Analysis of unshared errors made by the ideal and DMCMC unigram learners for items occurring 7 or more times in the first test set of each corpus.

Corpus	Ideal learner (undersegmentation)	DMCMC learner (oversegmentation)
Bernstein-Ratner	749	62
Pearl-Brent	1671	185