A Joint Learning Model of Word Segmentation, Lexical Acquisition, and Phonetic Variability

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the Conference on Empirical Methods in Natural Language Processing

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Abstract

We present a cognitive model of early lexical acquisition which jointly performs word segmentation and learns an explicit model of phonetic variation. We define the model as a Bayesian noisy channel; we sample segmentations and word forms simultaneously from the posterior, using beam sampling to control the size of the search space. Compared to a pipelined approach in which segmentation is performed first, our model is qualitatively more similar to human learners. On data with variable pronunciations, the pipelined approach learns to treat syllables or morphemes as words. In contrast, our joint model, like infant learners, tends to learn multiword collocations. We also conduct analyses of the phonetic variations that the model learns to accept and its patterns of word recognition errors, and relate these to developmental evidence.

1 Introduction

By the end of their first year, infants have acquired many of the basic elements of their native language. Their sensitivity to phonetic contrasts has become language-specific (Werker and Tees, 1984), and they have begun detecting words in fluent speech (Jusczyk and Aslin, 1995; Jusczyk et al., 1999) and learning word meanings (Bergelson and Swingley, 2012). These developmental cooccurrences lead some researchers to propose that phonetic and word learning occur jointly, each one informing the other (Swingley, 2009; Feldman et al., 2013). Previous computational models capture some aspects of this joint learning problem, but typically simplify the problem considerably, either by assuming an unrealistic degree of phonetic regularity for word segmentation (Goldwater et al., 2009) or assuming pre-segmented input for phonetic and lexical acquisition (Feldman et al., 2009; Feldman et al., in press; Elsner et al., 2012). This paper presents, to our knowledge, the first broad-coverage model that learns to segment phonetically variable input into words, while simultaneously learning an explicit model of phonetic variation that allows it to cluster together segmented tokens with different phonetic realizations (e.g., [ju] and [jI]) into lexical items (/ju/).

We base our model on the Bayesian word segmentation model of Goldwater et al. (2009) (henceforth GGJ), using a noisy-channel setup where phonetic variation is introduced by a finite-state transducer (Neubig et al., 2010; Elsner et al., 2012). This integrated model allows us to examine how solving the word segmentation problem should affect infants’ strategies for learning about phonetic variability and how phonetic learning can allow word segmentation to proceed in ways that mimic the idealized input used in previous models.

In particular, although the GGJ model achieves high segmentation accuracy on phonemic (non-variable) input and makes errors that are qualitatively similar to human learners (tending to undersegment the input), its accuracy drops considerably on phonetically noisy data and it tends to oversegment rather than undersegment. Here, we demonstrate that when the model is augmented to account for phonetic variability, it is able to learn common phonetic changes.
and by doing so, its accuracy improves and its errors return to the more human-like undersegmentation pattern. In addition, we find small improvements in lexicon accuracy over a pipeline model that segments first and then performs lexical-phonetic learning (Elsner et al., 2012). We analyze the model’s phonetic and lexical representations in detail, drawing comparisons to experimental results on adult and infant speech processing. Taken together, our results support the idea that a Bayesian model that jointly performs word segmentation and phonetic learning provides a plausible explanation for many aspects of early phonetic and word learning in infants.

2 Related Work

Nearly all computational models used to explore the problems addressed here have treated the learning tasks in isolation. Examples include models of word segmentation from phonemic input (Christiansen et al., 1998; Brent, 1999c; Venkataraman, 2001; Swingley, 2005) or phonetic input (Fleck, 2008; Rytting, 2007; Daland and Pierrehumbert, 2011; Boruta et al., 2011), models of phonetic clustering (Vallabha et al., 2007; Varadarajan et al., 2008; Dupoux et al., 2011) and phonological rule learning (Peperkamp et al., 2006; Martin et al., 2013).

Elsner et al. (2012) present a model that is similar to ours, using a noisy channel model implemented with a finite-state transducer to learn about phonetic variability while clustering distinct tokens into lexical items. However (like the earlier lexical-phonetic learning model of Feldman et al. (2009; in press)) their model assumes known word boundaries, so to perform both segmentation and lexical-phonetic learning, they use a pipeline that first segments using GGJ and then applies their model to the results.

Neubig et al. (2010) also present a transducer-based noisy channel model that performs joint inference on two out of the three tasks we consider here; their model assumes fixed probabilities for phonetic changes (the noise model) and jointly infers the word segmentation and lexical items, as in our ‘oracle’ model below (though unlike our system their model learns from phone lattices rather than a single transcription). They evaluate only on phone recognition, not scoring the inferred lexical items.

Recently, Börschinger et al. (2013) did present a

![Figure 1: The graphical model for our system (Eq. 1-4). Note that the $s_i$ are not distinct observations; they are concatenated together into a continuous sequence of characters which constitute the observations.](image-url)

joint learner for segmentation, phonetic learning, and lexical clustering, but the model and inference are tailored to investigate word-final /t/-deletion, rather than aiming for a broad coverage system as we do.

3 Model

We follow several previous models of lexical acquisition in adopting a Bayesian noisy channel framework (Eq. 1-4; Fig. 1). The model has two components: a source distribution $P(X)$ over utterances without phonetic variability $X$, i.e., intended forms (Elsner et al., 2012) and a channel or noise distribution $T(S|X)$ that translates them into the observed surface forms $S$. The boundaries between surface forms are then deterministically removed so that the actual observations are just the unsegmented string of characters in the surface forms.

$$G_0 | \alpha_0, p_{\text{stop}} \sim DP(\alpha_0, \text{Geom}(p_{\text{stop}}))$$

$$G_x | G_0, \alpha_1 \sim DP(\alpha_1, G_0)$$

$$X_i | X_{i-1} \sim G_{X_{i-1}}$$

$$S | X; \theta \sim T(S|X; \theta)$$

The source model is an exact copy of GGJ\(^1\); to generate the intended-form word sequences $X$, we

\(^1\)We use their best reported parameter values: $\alpha_0 = 3000, \alpha_1 = 100, p_{\text{stop}} = .2$ and for unigrams, $\alpha_0 = 20$. 

\[G_0 | \alpha_0, p_{\text{stop}} \sim DP(\alpha_0, \text{Geom}(p_{\text{stop}}))\]

\[G_x | G_0, \alpha_1 \sim DP(\alpha_1, G_0)\]

\[X_i | X_{i-1} \sim G_{X_{i-1}}\]

\[S | X; \theta \sim T(S|X; \theta)\]
sample a random language model from a hierarchical Dirichlet process \( \text{Teh et al., 2006} \) with character strings as atoms. To do so, we first draw a unigram distribution \( G_0 \) from a Dirichlet process prior whose base distribution generates intended form word strings by drawing each phone in turn until the stop character is drawn (with probability \( p_{\text{stop}} \)). Then, for each possible context word \( x \), we draw a conditional distribution on words following that context \( G_x = P(X_i = \bullet | X_{i-1} = x) \) using \( G_0 \) as a prior. Finally, we sample word sequences \( x_1 \ldots x_n \) from the bigram model.

The channel model is a finite transducer with parameters \( \theta \) which independently rewrites single characters from the intended string into characters of the surface string. We use MAP point estimates of these parameters; single characters (without \( n \)-gram context) are used for computational efficiency. Also for efficiency, the transducer can insert characters into the surface string, but cannot delete characters from the intended string. As in several previous phonological models \( \text{Dreyer et al., 2008}; \text{Hayes and Wilson, 2008} \) the probabilities are learned using a feature-based log-linear model. For features, we use all the unigram features from \( \text{Elsner et al., 2012} \) which check faithfulness to voicing, place and manner of articulation (for example, for \( k \rightarrow g \), active features are \text{faith-manner, faith-place, output-g and voiceless-to-voiced} \).

Below, we present two methods for learning the transducer parameters \( \theta \). The oracle transducer is estimated using the gold-standard word segmentations and intended forms for the dataset; it represents the best possible approximation under our model of the actual phonetics of the dataset. We can also estimate the transducer using the \text{EM} algorithm. We first initialize a simple transducer by putting small weights on the faithfulness features to encourage phonologically plausible changes. With this initial model, we begin running the sampler used to learn word segmentations. After several hundred sampler iterations, we start re-estimating the transducer by maximum likelihood after each iteration. We regularize our estimates by adding 200 pseudocounts for the rewrite \( x \rightarrow \) during training (rather than regularizing the weights for particular features). We also show segment only results for a model without the transducer component (i.e., \( S = X \)); this recovers the GGJ baseline.

4 Inference

Inference for this model is complicated for two reasons. First, the hypothesis space is extremely large. Since we allow the input string to be probabilistically lengthened, we cannot be sure how long it is, nor which characters it contains. Second, our hypotheses about nearby characters are highly correlated due to lexical effects. When deciding how to interpret \( \text{[\text{wan}]} \), if we posit that the intended vowel is \( /\alpha/ \), the word is likely to be \( /\text{wan}/ \) “one” and the next word begins with \( /t/ \); if instead we posit that the vowel is \( /\alpha/ \), the word is probably \( /\text{wan}/ \) “want”. Thus, inference methods that change only one character at a time are unlikely to mix well. Since they cannot simultaneously change the vowel and resegment the \( /t/ \), they must pass through a low-probability intermediate state to get from one state to the other, so will tend to get stuck in a bad local minimum. A Gibbs sampler which inserts or deletes a single segment boundary in each step \( \text{Goldwater et al., 2009} \) suffers from this problem.

\( \text{Mochihashi et al., 2009} \) describe an inference method with higher mobility: a block sampler for the GGJ model that samples from the posterior over analyses of a whole utterance at once. This method encodes the model as a large HMM, using dynamic programming to select an analysis. We encode our own model in the same way, constructing the HMM and composing it with the transducer \( \text{Mohri, 2004} \) to form a larger finite-state machine which is still amenable to forward-backward sampling.

4.1 Finite-state encoding

Following \( \text{Mochihashi et al., 2009} \) and \( \text{Neubig et al., 2010} \), we can write the original GGJ model as a Hidden Semi-Markov model. States in the HMM, written \( \text{ST: } [w] [C] \), are labeled with the previous word \( w \) and the sequence of characters \( C \) which have so far been incorporated into the current word. To produce a word boundary, we transition from \( \text{ST: } [w] [C] \) to \( \text{ST: } [C] \) with probability \( P(x_i = C | x_{i-1} = w) \). We can also add the next character \( s \) to the current word, transitioning from \( \text{ST: } [w] [C] \) to \( \text{ST: } [w] [C : s] \), at no cost (since the full cost of the word is paid at its boundary, there
To recognize the current word, we transition to \( ST: [C] [u] \) with probability \( P(x_{i} = C|x_{i-1} = u) \). To parse a new surface character \( s \) by positing intended character \( x \) (note that \( x \) might be \( \epsilon \)), we transition to \( ST: [w] [C : x] [S : s] \) with probability \( T(s|x) \). (As above, we pay no cost for our choice of \( x \), which is paid for when we recognize the word; however, we must pay for \( s \).) For efficiency, we do not allow the \( G_0 \) states to hypothesize different surface and intended characters, so when we initially propose an unknown word, it must surface as itself.\(^3\)

4.2 Beam sampler

This machine has too many states to fully fill the chart before backward sampling, so we restrict the set of trajectories under consideration using beam sampling (Van Gael et al., 2008) and simulated annealing.

The beam sampler is closely related to the standard beam search technique, which uses a probability cutoff to discard parts of the FST which are unlikely to figure in the eventual solution. Unlike conventional beam search, the sampler explores using stochastic cutoffs, so that all trajectories are explored, but most of the bad ones are explored infrequently, leading to higher efficiency.

We design our beam sampler to restrict the set of potential intended characters at each timestep. In particular, given a stream of input characters \( S = s_1 \ldots s_n \), we introduce a set of auxiliary cutoff variables \( U = u_1 \ldots u_n \). The \( u_i \) variables represent limits on the probability of the emission of surface character \( s_i \); we exclude any hypothesized \( x_i \) whose probability of generating \( s_i, T(s_i|x_i) \), is less than \( u_i \). To create a beam sampling scheme, we must devise a distribution for \( U \) given a state sequence \( Q \) (as discussed above, the sequence of states encodes the intended character sequence and the segmentation of the surface string), \( P_u(U|Q) \) and then incorporate the probability of \( U \) into the forward messages.

If \( q_i \) is the state in \( Q \) at which \( s_i \) is generated, and \( x_i \) the corresponding intended character, we require that \( P_u < T(s_i|x_i) \); that is, the cutoffs must not exclude any states in the sequence \( Q \). We define \( P_u \)

\(^2\)Though not mentioned by Mochihashi et al. (2009) or Neubig et al. (2010), this construction is not exact, since transitions in a Bayesian HMM are exchangeable but not independent (Beal et al., 2001): if a word occurs twice in an utterance, its probability is slightly higher the second time. For single utterances, this bias is small and easy to correct for using a Metropolis-Hastings acceptance check (Börschinger and Johnson, 2012) using the path probability from the HMM as the proposal.

\(^3\)Again, this approximation is corrected for by the Metropolis-Hastings step.
as a \( \lambda \)-mixture of two distributions:

\[
P_u(u|s_i, x_i) = \lambda U(0, \min(0.05, T(s_i|x_i))) + (1 - \lambda)T(s_i|x_i)Beta(5, 1e^{-5})
\]

The former distribution is quite unrestrictive, while the latter prefers to prune away nearly all the states. Thus, for most characters in the string, we do not permit radical changes, while for a fraction, we do.

We follow Huggins and Wood (2013), who extended Van Gael et al. (2008) to the case of a non-uniform \( P_u \), to define our forward message \( \alpha \) as:

\[
\alpha(q_i, i) \propto P(q_i, s_{0..i}, U_{0..i})
= \sum_{q_{i-1}} P_u(u_i|s_i, x_i)T(s_i|x_i)\alpha(q_{i-1}, i-1)
\]

This is the standard HMM forward message, augmented with the probability of \( u \). Since \( P_u(\cdot|s_i, x_i) \) is required to be less than \( T(s_i|x_i) \), it will be \( 0 \) whenever \( T(s_i|x_i) < u \); this is how the \( u \) variables function as cutoffs. In practice, we use the \( u \) variables to filter the lexical items that begin at each position \( i \) in advance, using a simple 0/1 edit distance Markov model which runs faster than our full model. (For example, we can quickly check if the current \( U \) allows want as the intended form for walk at \( i \); if not, we can avoid constructing the prefix \( ST: [x_{i-1}] [ua] [w] \) since the continuation will fail.)

The algorithm’s speed depends on the size and uncertainty of the inferred LM: large numbers of plausible words mean more states to explore. When inference starts, and the system is highly uncertain about word boundaries, it is therefore reasonable to limit the exploration of the character sequence. We do so by annealing in two ways: as in Goldwater et al. (2009), we raise \( P(X) \) (Eq. 3) to a power \( t \) which increases linearly from .3. To sample from the posterior, we would want to end with \( t = 1 \), but as in previous noisy-channel models (Elsner et al., 2012; Bahl et al., 1980) we get better results when we emphasize the LM at the expense of the channel and so end at \( t = 2 \). Meanwhile, as \( t \) rises and we explore fewer implausible lexical sequences, we can explore the character sequence more. We begin by setting the \( \lambda \) interpolation parameter of \( P_u \) to 0 to minimize exploration and increase it linearly to .3 (allowing the system to change about a third of the characters on each sweep). This is similar to the scheme for altering \( P_u \) in Huggins and Wood (2013).

### 4.3 Dataset and metrics

We use the corpus released by Elsner et al. (2012), which contains 9790 child-directed English utterances originally from the Bernstein-Ratner corpus (Bernstein-Ratner, 1987) and later transcribed phonetically (Brent, 1999). This standard word segmentation dataset was modified by Elsner et al. (2012) to include phonetic variation by assigning each token a pronunciation independently selected from the empirical distribution of pronunciations of that word type in the closely-transcribed Buckeye Speech Corpus (Pitt et al., 2007). Following previous work, we hold out the last 1790 utterances as unseen test data during development. In the results presented here, we run the model on all 9790 utterances but score only these 1790. We average results over 5 runs of the model with different random seeds.

We use standard metrics for segmentation and lexicon recovery. For segmentation, we report precision, recall and F-score for word boundaries (bds), and for the positions of word tokens in the surface string (srf; both boundaries must be correct).

For normalization of the pronunciation variation, we follow Elsner et al. (2012) in measuring how well the system clusters together variant pronunciations of the same lexical item, without insisting that the intended form the system proposes for them match the one in our corpus. For example, if the system correctly clusters [ju] and [jit] together but assigns them the incorrect intended form /jI/, we can still give credit to this cluster if it is the one that overlaps best with the gold-standard /jI/ cluster. To compute these scores, we find the optimal one-to-one mapping between our clusters of pronunciations and the true lexical entries, then report scores for mapped tokens (mtk; boundaries and mapping to gold standard cluster must be correct) and mapped types\(^4\) (mlx).

\(^4\)Elsner et al. (2012) calls the mlx metric lexicon \( F \), which is possibly confusing. We map the clusters to a gold-standard lexicon (plus potentially some words that don’t correspond to anything in the gold standard) and compute a type-level F-score on this lexicon.
We begin by evaluating our model as a word segmentation system. (Table 1 gives segmentation and normalization scores for various models and baselines on the 1790 test utterances.) We first confirm that our inference method is reasonable. The bigram model without variability (“segment only”) should have the same segmentation performance as the standard \texttt{dpseg} implementation of GGJ. This is the case: \texttt{dpseg} has boundary $F$ of 80.3 and token $F$ of 62.4; we get 81.6 and 65.4. Thus, our sampler is finding good solutions, at least for the no-variability model.

We compare segmentation scores between the “segment only” system and the two bigram models with transducers (“oracle” and “EM”). While these systems all achieve similar segmentation scores, they do so in different ways. “Segment only” finds a solution with boundary precision 73.9% and boundary recall 91.0% for a total $F$ of 81.6%. The low precision and high recall here indicate a tendency to oversegment; when the analysis of a given subsequence is unclear, the system prefers to chop it into small chunks. The bigram models which incorporate transducers score $P$: 76.1, $R$: 83.8 (oracle) and $P$: 80.1, $R$: 83.0 (EM), indicating that they prefer to find longer sequences (undersegment) more.

In previous experiments on datasets without variation, GGJ also has a strong tendency to undersegment the data (boundary $P$: 90.1, $R$: 80.3), which Goldwater et al. argue is rational behavior for an ideal learner seeking a parsimonious explanation for the data. Undersegmentation occurs especially when ignoring lexical context (a unigram model), but to some extent even in bigram models. Human learners also tend to learn collocations as single words (Peters, 1983; Tomasello, 2000), and the GGJ model has been shown to capture several other effects seen in laboratory segmentation tasks (Frank et al., 2010). Together, these findings support the idea that human learners may behave in important respects like the Bayesian ideal learners that Goldwater et al. presented.

However, experiments on data with variation have called these conclusions into question. In particular, GGJ has previously been shown to oversegment rather than undersegment as the input grows noisier (Fleck, 2008), and our results replicate this finding (oversegmentation for the “segment only” model). In addition, the GGJ bigram model, which achieves much higher segmentation accuracy than the unigram model on clean data, actually performs worse on very noisy data (Jansen et al., 2013). Infants are known to track statistical dependencies across words (Gómez and Maye, 2005), so it is worrisome that these dependencies hurt GGJ’s segmentation accuracy when learning from noisy data.

Our results show that modeling phonetic variability reverses the problematic trends described above. Although the models with phonetic variability show similar overall segmentation accuracy on noisy data to the original GGJ model, the pattern of errors changes, with less oversegmentation and more un-

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
& Prec & Rec & F-score \\
\hline
Pipeline (segment, then cluster): (Elsner et al., 2012) & & & \\
\hline
Bds & 70.4 & \textbf{93.5} & 80.3 \\
Srf & 56.5 & 69.7 & 62.4 \\
Mtk & 44.2 & \textbf{54.5} & 48.8 \\
Mlx & 48.6 & 43.1 & 45.7 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
& Prec & Rec & F-score \\
\hline
Unigram model, oracle transducer & & & \\
\hline
Bds & 79.3 (-0.6:0.7) & 91.0 (-0.6:0.4) & \textbf{81.6} (-0.5:0.6) \\
Srf & 60.8 (-0.7:1.1) & \textbf{70.8} (-0.8:0.9) & 65.4 (-0.6:1.0) \\
Mtk & 41.6 (-0.6:1.2) & 48.4 (-0.5:1.2) & 44.8 (-0.6:1.2) \\
Mlx & 36.6 (-0.7:0.8) & \textbf{49.8} (-1.0:0.8) & 42.2 (-0.9:0.8) \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
& Prec & Rec & F-score \\
\hline
Bigram model, segment only & & & \\
\hline
Bds & \textbf{81.4} (-0.8:0.4) & 72.1 (-0.9:0.8) & 76.4 (-0.5:0.7) \\
Srf & 63.6 (-1.0:1.1) & 58.5 (-1.2:1.2) & 60.9 (-0.9:1.2) \\
Mtk & 46.8 (-1.0:1.1) & 43.0 (-1.1:1.2) & 44.8 (-1.0:1.2) \\
Mlx & \textbf{56.7} (-1.1:1.0) & 47.6 (-1.4:0.8) & \textbf{51.7} (-1.2:0.8) \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
& Prec & Rec & F-score \\
\hline
Bigram model, EM transducer & & & \\
\hline
Bds & 76.1 (-0.6:0.6) & 83.8 (-0.9:1.0) & 79.8 (-0.8:0.4) \\
Srf & 62.2 (-0.9:1.0) & 66.7 (-1.2:1.1) & 64.4 (-1.1:0.8) \\
Mtk & 47.2 (-0.7:0.9) & 50.6 (-1.0:0.8) & 48.8 (-0.8:0.7) \\
Mlx & 40.1 (-1.0:1.2) & 43.7 (-0.6:0.7) & 41.8 (-0.8:0.6) \\
\hline
\end{tabular}
\end{table}

Table 1: Mean segmentation (bds, srf) and normalization (mtk, mlx) scores on the test set over 5 runs. Parentheses show min and max scores as differences from the mean.

5 Results and discussion

In the following sections, we analyze how our model with variability compares to GGJ on noisy data. We give quantitative scores and also show that qualitative patterns of errors are often similar to those of human learners and listeners.

5.1 Clean versus variable input

We begin by evaluating our model as a word segmentation system. (Table 1 gives segmentation and normalization scores for various models and baselines on the 1790 test utterances.) We first confirm that our inference method is reasonable. The bigram model without variability (“segment only”) should have the same segmentation performance as the standard \texttt{dpseg} implementation of GGJ. This is the case: \texttt{dpseg} has boundary $F$ of 80.3 and token $F$ of 62.4; we get 81.6 and 65.4. Thus, our sampler is finding good solutions, at least for the no-variability model.

We compare segmentation scores between the
dersegmentation. Thus, their qualitative performance on variable data resembles GGJ’s on clean data, and therefore the behavior of human learners.

5.2 Phonetic variability

We next analyze the model’s ability to normalize variations in the pronunciation of tokens, by inspecting the mtk score. The “segment only” baseline is predictably poor, $F$: 44.8. The pipeline model scores 48.8, and our oracle transducer model matches this exactly. The EM transducer scores better, $F$: 49.6. Although the confidence intervals overlap slightly, the EM system also outperforms the pipeline on the other $F$-measures; altogether, these results suggest at least a weak learning synergy (Johnson, 2008) between segmentation and phonetic learning.

It is interesting that EM can perform better than the oracle. However, EM is more conservative about which sound changes it will allow, and thus tends to avoid mistakes caused by the simplicity of the transducer model. Since the transducer works segment-by-segment, it can apply rare contextual variations out of context. EM benefits from not learning these variations to begin with.

We can also compare the bigram and unigram versions of the model. The unigram model is a reasonable segmenter, though not quite as good as the bigram model, with boundary $F$ of 76.4 and token $F$ of 60.9 (compared to 79.8 and 64.4 using the bigram model). However, it is not good at normalizing variation; its mtk score is comparable to the baseline at 44.8%.\(^5\) Although bigram context is only moderately effective for telling where words are, the model seems heavily reliant on lexical context to decide what words it is hearing.

5.3 Error analysis

To gain more insight into the differing behavior of our model versus a pipelined system, we inspect the intended word strings $X$ proposed by each one in detail. Below, we categorize the kinds of intended word strings that the model might propose to span a given gold-standard word token:

**Correct** Correctly segmented, mapped to the correct lexical item (e.g., gold intended /ju/, surface /ju/).

**Wrong form** Correctly segmented, mapped to the wrong lexical item (/ju/, surf. /ju/, int. /jEs/)

**Colloc** Missegmented as part of a sequence whose boundaries correspond to real word boundaries (/ju•want/, surf. /juwant/, int. /juwant/)

**Corr. colloc** As above, but proposed lexical item maps to this word (/ar•ju/, surf. /juw/, int. /ju/)

**Split** Missegmented with a word-internal boundary (/d@giz/, surf. /d@giz#, int. /d@giz#)

**Corr. split** As above, but one proposed word maps correctly (/d@gi/, surf. /d@gi#, int. /d@gi#)

**One boundary** One boundary correct, the other wrong (/ju•wa…/, surf. /juw/, int. /juw/)

**Other** Not a collocation, both boundaries are wrong (/du•ju•wa…/, surf. /ujuw/, int. /ujuw/)

Table 2 shows the distribution over intended word strings proposed by the “segment only” baseline and the EM-learned transducer. Both systems propose a large number of correct forms, and the most common error category is “wrong form” (lexical error without segmentation error), an error which could potentially be repaired in a pipeline system. However, the remaining errors represent segmentation mistakes which a pipeline could not repair. Here the two systems behave quite differently. The EM-learned transducer analyses 14% of real tokens as parts of multiword collocations like “doyou”; in another 1.35%, the underlying content word is even correctly detected. The non-variable system, on the other hand, analyses 15% of real tokens by splitting them into pieces. Since infant learners tend to learn collocations, this supports our analysis that the model with variation better models human behavior.

\(^5\)Elsner et al. (2012) show a similar result for a unigram version of their pipelined system.
Infants are also conservative in generalizing across phonetic variability, showing a delayed ability to segment words beginning with vowels. We hypothesize that this could lead to difficulty in segmenting and recognizing vowel-initial words. The model's behavior is consistent with this hypothesis (Table 4). Both the “segment only” and EM transducer models find approximately the same proportion of vowel-initial tokens, and both systems do somewhat better on consonant-initial words than vowel-initial words. The advantage is stronger for the transducer model, which gets only 41.5% of vowel-initial tokens correct as opposed to 52.1% of consonant-initial words. It proposes more collocations for vowel-initial words (19.2%) than for consonants (12.5%). In cases where they do not propose a collocation, both systems are somewhat more likely to find the right boundary of a vowel-initial token than the left boundary (although again this difference is larger for the EM system); this suggests that the problem is indeed caused by the initial segment.

### 5.4 Phonetic Learning

We next compare phonetic variations learned by the model to characteristics of infant speech perception. Infants show an asymmetry between consonants and vowels, losing sensitivity to non-native vowel contrasts by eight months (Kuhl et al., 1992; Bosch and Sebastián-Gallés, 2003) but to non-native consonant contrasts only by 10-12 months (Werker and Tees, 1984). The observed ordering is somewhat puzzling when one considers the availability for distributional information (Maye et al., 2002), which is much stronger for stop consonants than for vowels (Lisker and Abramson, 1964; Peterson and Barney, 1952). Infants are also conservative in generalizing across phonetic variability, showing a delayed abil-

---

6Not all the variants are merged, however. /ju/ and /jæ/ etc. are still occasionally analyzed as separate lexical items.
ity to generalize across talkers, affects, and dialects. They have difficulty recognizing word tokens that are spoken by a different talker or in a different tone of voice until 11 months (Houston and Jusczyk, 2000; Singh et al., 2004), and the ability to adapt to unfamiliar dialects appears to develop even later, between 15 and 19 months (Best et al., 2009; Heugten and Johnson, in press; White and Aslin, 2011).

Similar to infants, our model shows both a vowel-consonant asymmetry and a reluctance to accept the full range of adult phonetic variability. Table 5 shows some segment-to-segment alternations learned in various transducers. The oracle learns a large amount of variation (u surfaces as itself only 68% of the time) involving many different segments, whereas EM is similar to infant learners in learning a more conservative solution with fewer alternations overall. Moreover, EM appears to identify patterns of variability in vowels before consonants. It learns a similar range of alternations for u as in the oracle, although it treats the sound as less variable than it actually is. It learns much less variability for consonants; it picks up the alternation of δ with s and z, but predicts that δ will surface as itself 91% of the time when the true figure is only 69%. And it fails to learn any meaningful alternations involving k. These results suggest that patterns of variability in vowels are more evident than patterns of variability in consonants when infants are beginning to solve the word segmentation problem.

To investigate the effect of data size on this conservativism, we ran the system on 1000 utterances instead of 9790. This leads to an even more conservative solution, with variations for u but none of the others (although i and δ still vary more than k).

### 5.5 Segmentation and recognition errors

A particularly interesting set of errors are those that involve both a missegmentation and a simultaneous misrecognition, since the joint model is prone to such errors while the pipelined model is not. Relatively little is known about infants’ misrecognitions of words in fluent speech, although it is clear that they find words in medial position harder (Plunkett, 2005; Seidl and Johnson, 2006). However, adults make missegmentation/misrecognition errors fairly often, especially when listening to noisy audio (Butterfield and Cutler, 1988). Such errors are more common when the misrecognized word belongs to a prosodically rare class and when the incorrectly hypothesized string contains frequent words (Cutler, 1990); phonetically ambiguous words are also more commonly recognized as the more frequent of two options (Connine et al., 1993). For the indefinite article “a” (often reduced to [ə]), lexical context is the main factor in deciding between ambiguous interpretations (Kim et al., 2012). In rapid speech, listeners have few phonetic cues to indicate whether it is present at all (Dilley and Pitt, 2010). Below, we analyze various misrecognitions made by our system (using the EM transducer), and find some similar effects.

The easiest cases to analyze are those with no missegmentation: the proposed boundaries are correct, and the proposed lexical entry corresponds to a real word\(^7\), but not the correct one. Most of them correspond to homophones (Table 6).

Common cases with a missegmentation include *it* and *is*, *a* and *is*, *it’s* and *is*, *who*, *who’s* and *whose*, *that’s* and *what’s*, and *there* and *there’s*. In general, these errors involve words which sometimes appear

\(^7\)The one-to-one mapping can be misleading, as it may map a large cluster to a real word on the basis of one or two tokens if all other tokens correspond to a different word already used for another cluster. We manually filter out a few cases like this.

---

### Table 5: Learned phonetic alternations: top 4 outputs

<table>
<thead>
<tr>
<th>System</th>
<th>x</th>
<th>top 4 outputs s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle</td>
<td>u i δ k</td>
<td>u .68 .85 .69 .93</td>
</tr>
<tr>
<td></td>
<td>i i δ k</td>
<td>i .05 .03 .91 .93</td>
</tr>
<tr>
<td></td>
<td>δ δ s z</td>
<td>δ .04 .02 .91 .02</td>
</tr>
<tr>
<td></td>
<td>k k .93 .9 d .02 .02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[φ] r .21 .21 .32 .32</td>
<td></td>
</tr>
<tr>
<td>EM (full)</td>
<td>u i δ k</td>
<td>u .75 .90 .91 .98</td>
</tr>
<tr>
<td></td>
<td>i i δ k</td>
<td>i .08 .04 .91 .95</td>
</tr>
<tr>
<td></td>
<td>δ δ s z</td>
<td>δ .04 .02 .91 .95</td>
</tr>
<tr>
<td></td>
<td>k k .93 .93 .99</td>
<td></td>
</tr>
<tr>
<td>EM (only</td>
<td>u i δ k</td>
<td>u .82 .90 .75 .21</td>
</tr>
<tr>
<td>1000 utts)</td>
<td>i i δ k</td>
<td>i .04 .04 .97</td>
</tr>
<tr>
<td></td>
<td>δ δ .95</td>
<td>δ .03 .02 .95</td>
</tr>
<tr>
<td></td>
<td>k k .99</td>
<td>k .02 .04 .99</td>
</tr>
<tr>
<td></td>
<td>[φ] o .32</td>
<td>o .21 .18 .32</td>
</tr>
</tbody>
</table>

---

50
Table 6: Top ten errors involving confusion between real, correctly segmented words: the most common pronunciation of the actual token and its orthographic form, the same for the proposed token, and the frequency.

<table>
<thead>
<tr>
<th>Actual</th>
<th>proposed</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>/tu/ “two”</td>
<td>/ta/ “to”</td>
<td>95</td>
</tr>
<tr>
<td>/kin/ “can”</td>
<td>/kænt/ “can’t”</td>
<td>67</td>
</tr>
<tr>
<td>/en/ “and”</td>
<td>/æn/ “an”</td>
<td>61</td>
</tr>
<tr>
<td>/hɪz/ “his”</td>
<td>/ɪz/ “is”</td>
<td>57</td>
</tr>
<tr>
<td>/æ/ “the”</td>
<td>/ə/ “ah”</td>
<td>51</td>
</tr>
<tr>
<td>/wʌts/ “what’s”</td>
<td>/wʌnts/ “wants”</td>
<td>40</td>
</tr>
<tr>
<td>/wʌnt/ “want”</td>
<td>/wʌnt/ “won’t”</td>
<td>39</td>
</tr>
<tr>
<td>/yu/ “you”</td>
<td>/jʊ/ “yeah”</td>
<td>39</td>
</tr>
<tr>
<td>/fɔr/ “for”</td>
<td>/fɔr/ “four”</td>
<td>30</td>
</tr>
<tr>
<td>/hɪ/ “here”</td>
<td>/hɪ/ “he’ll”</td>
<td>28</td>
</tr>
</tbody>
</table>

with a morpheme or clitic (which can easily be mis-segmented as part of something else), words which differ by one segment, and frequent function words which often appear in similar contexts. These tendencies match those shown by adult human listeners.

A particularly distinctive set of joint recognition and segmentation errors are those where an entire real token is treated as phonetic “noise”—that is, it is segmented along with an adjacent word, and the system clusters the whole sequence as a token of that word. The most common examples are “that’s a” identified as “that’s”, “have a” identified as “have”, “sees a” identified as “sees” and other examples involving “a”, a word which also frequently confuses humans (Kim et al., 2012; Dilley and Pitt, 2010). However, there are also instances of “who’s in” as “who’s”, “does it” as “does”, and “can you” as “can”.

6 Conclusion

We have presented a model that jointly infers word segmentation, lexical items, and a model of phonetic variability; we believe this is the first model to do so on a broad-coverage naturalistic corpus. Our results show a small improvement in both segmentation and normalization over a pipeline model, providing evidence for a synergistic interaction between these learning tasks and supporting claims of interactive learning from the developmental literature on infants. We also reproduced several experimental findings; our results suggest that two vowel-consonant asymmetries, one from the word segmentation literature and another from the phonetic learning literature, are linked to the large variability in vowels found in natural corpora. The model’s correspondence with human behavioral results is by no means exact, but we believe these kinds of predictions might help guide future research on infant phonetic and word learning.

Acknowledgements

Thanks to Mary Beckman for comments. This work was supported by EPSRC grant EP/H050442/1 to the second author.

References


Luc Boruta, Sharon Peperkamp, Benoît Crabbé, and Emmanuel Dupoux. 2011. Testing the robustness of online word segmentation: Effects of linguistic diversity and

---

*Software is available from the ACL archive; updated versions may be posted at [https://bitbucket.org/melsner/beamseg](https://bitbucket.org/melsner/beamseg).*


Naomi H. Feldman, Thomas L. Griffiths, Sharon Goldwater, and James L. Morgan. in press. A role for the developing lexicon in phonetic category acquisition. Psychological Review.


