Unsupervised extraction of recurring words from infant-directed speech

McInnes, F. R., & Goldwater, S. (2011). Unsupervised extraction of recurring words from infant-directed speech. In Proceedings of the 33rd Annual Conference of the Cognitive Science Society.

Unsupervised Extraction of Recurring Words from Infant-Directed Speech

Fergus R. McInnes (Fergus.McInnes@ed.ac.uk) and **Sharon J. Goldwater (sgwater@inf.ed.ac.uk)**
School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 1AB, UK

Abstract

To date, most computational models of infant word segmentation have worked from phonemic or phonetic input, or have used toy datasets. In this paper, we present an algorithm for word extraction that works directly from naturalistic acoustic input: infant-directed speech from the CHILDES corpus. The algorithm identifies recurring acoustic patterns that are candidates for identification as words or phrases, and then clusters together the most similar patterns. The recurring patterns are found in a single pass through the corpus using an incremental method, where only a small number of utterances are considered at once. Despite this limitation, we show that the algorithm is able to extract a number of recurring words, including some that infants learn earliest, such as *Mommy* and the child's name. We also introduce a novel information-theoretic evaluation measure.

Keywords: language acquisition; word segmentation; speech recognition; computational modelling.

Introduction

One of the first problems children face in learning language is how to segment individual words from the continuous stream of acoustic input they hear. Experimental evidence suggests that infants are sensitive to the statistical patterns created by strings of words in a nonsense language (Saffran, Newport, & Aslin, 1996), and can use these statistical cues to distinguish words from non-words at an earlier age than other cues such as stress (Thiessen & Saffran, 2003). Since this evidence began to surface, a number of computational models have been proposed to explain how infants might exploit the statistical information in the speech input in order to identify words or word boundaries (e.g., Brent, 1999; Christiansen, Allen, & Seidenberg, 1998; Goldwater, Griffiths, & Johnson, 2009; Rytting, 2007). Nearly all such models have assumed a string of phonemic or phonetic symbols as input, though recently some researchers have begun to explore the problem of word identification in a less idealized scenario, using acoustic feature vectors as input (Aimetti, 2009; Driesen, ten Bosch, & Van hamme, 2009; Räsänen, 2010). The work described here falls into the latter category, and is motivated by evidence that infants begin to segment some words as early as six months old, while their native language phonology is still incomplete (Bortfeld, Morgan, Golinkoff, & Rathbun, 2005). This suggests the possibility that some early words may be learned as a whole from the acoustic signal, without any sub-word level of representation.

The algorithm we present is similar to that of Aimetti (2009), in that both are based on recent work by Park and Glass (2008) (henceforth, P&G). However, our work differs from these and other recent papers in two important ways. First, we use a corpus of naturalistic infant-directed speech as input. This contrasts with P&G, who use adult-directed speech, and with Aimetti (2009), Driesen et al. (2009), and Räsänen (2010), who all use a carefully constructed corpus of utterances with a specified form and a small vocabulary. Second, P&G's algorithm (the only other one to be tested on unscripted speech) processes the entire input corpus at once, searching for acoustically similar fragments. Here, we develop an incremental version of the algorithm that processes only a few utterances at a time, simulating the limited memory of an infant learner. Although the incremental algorithm does not perform as well as the batch algorithm, it is still able to extract a number of words and phrases from the input. This success is due in part to the structure of infant-directed speech: since words are often repeated close together, a simple matching algorithm can find and extract such repetitions even when only a few utterances are considered at a time. Although we do not claim that the particular matching procedure used here is necessarily similar to one used by infants, our results do suggest that an incremental acoustic matching procedure could be a successful way to extract words despite variability in the speech signal.

In addition to these simulation results, we develop a novel entropy-based evaluation measure. Previous researchers working with P&G's algorithm have mostly used qualitative evaluation; quantitative evaluation was slow, limited in scope, and often subjective because it required examining the output by hand. An exception is presented by Jansen, Church, and Hermansky (2010), but their method depends on an information-retrieval task. Our evaluation method can be applied automatically (assuming a phonemic forced alignment is available), which makes quantitative comparisons between different conditions or learning algorithms much easier.

In the remainder of the paper, we first present the pattern extraction algorithm and the evaluation measure. We then describe our experiments, presenting both qualitative and quantitative results. We conclude with a discussion of our findings and directions for further work.

Pattern Extraction Algorithm

Our algorithm is a modification of the segmental dynamic time warping (DTW) algorithm of P&G. The input is a set of utterances, each represented by a sequence of *acoustic frames* (MFCC feature vectors, as standardly used for automatic speech recognition). The output is a set of clusters, with each cluster consisting of a set of acoustically similar speech *fragments* (subsequences of frames from the input). The intention is that each cluster should contain instances of a single word or phrase that occurs repeatedly in the input.

The algorithm has two stages (described in more detail in the following subsections): DTW matching, in which pairs of utterances are compared to identify any similar fragments; and clustering, in which the matched fragments are grouped together across the whole set of utterances using a graph-based clustering method. In P&G's original algorithm, every utter-

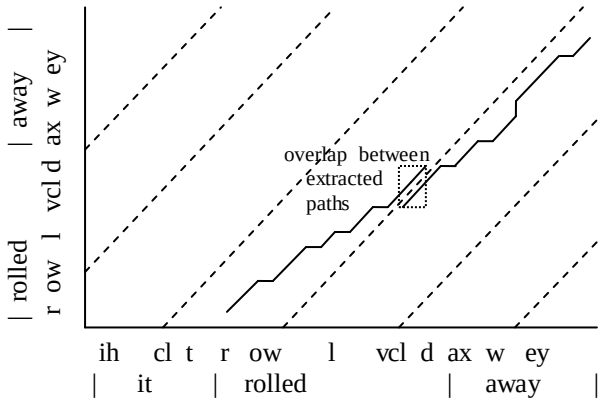

Figure 1: DTW matching of an utterance pair (with orthographic and phonemic transcriptions). See explanation in text.

ance is compared against every other utterance in the DTW matching stage. As a possible model of whole-word segmentation by human infants, this method is implausible because it requires all utterances to be stored in memory at once, and because the amount of computation grows as the square of the number of utterances heard. Here, we present an incremental version of the matching algorithm where utterances are only compared to other utterances within a fixed recency window. Any utterance in the more distant past (outside the window) is represented in memory only by the fragments already found in it. Our complete algorithm is not yet fully incremental, because individual fragments are stored separately in memory until the entire corpus has been processed, at which point all fragments are clustered. However, this already represents a large decrease in memory over storing the entire corpus, and we anticipate that further modifications to implement incremental clustering would be possible.

DTW Matching

Dynamic time warping is a standard algorithm in the speech recognition literature, used to align sequences of acoustic frames in such a way as to minimize the spectral distortion (the sum of the distances between aligned frames). It is a dynamic programming algorithm, conceptually similar to weighted minimum-edit-distance string matching. The segmental version introduced by P&G is intended to identify and align only those subsequences of frames that are relatively similar, and therefore might be different instances of the same word.

Figure 1 illustrates, using the utterances *rolled away* and *it rolled away*. The actual acoustic frames are not shown, but would range along the *x*-axis for one utterance (earlier frames to the left), and along the *y*-axis for the other (earlier frames at the bottom). Transcriptions are shown for illustration only, and are not used as input to the algorithm (although the phonemic transcripts are used during evaluation). The solid lines represent alignments between the two utterances, with diagonal parts indicating matches where consecutive frames in one utterance are matched to consecutive frames in the other, and horizontal or vertical parts indicating matches where consecutive frames in one utterance are matched to the same frame in the other (allowing for duration differences). If the two utterances were identical, then the best alignment would be a single diagonal line from the origin to the upper right corner of the graph. P&G's algorithm works by first identifying the optimal alignment path within each diagonal band (dashed lines; these bands constrain the algorithm so that only a certain amount of time offset is allowed in the alignment – otherwise it would be possible to align all frames of one utterance to the first frame of the other, or similar trivial solutions). It then extracts fragments of the alignments that have low spectral distortion – good matches.

We made a number of small modifications to the P&G matching procedure to improve its performance before implementing the incremental version. First, we constrain alignment paths so that any horizontal or vertical step must be preceded by a diagonal step (as in the "Type I" constraints of Myers, Rabiner, & Rosenberg, 1980), in order to minimize distortion in the time alignment. Second, to extract fragments with low spectral distortion, we take those paths of at least length *min_path_len* in which all the frame-to-frame distortions are below a threshold *core_thr*. This replaces the length-constrained minimum average algorithm of P&G, and has the advantage that multiple matching regions can be found within one diagonal band. The matching regions are then extended to include any nearby low-distortion matches, where up to *max_hi* consecutive points with higher distortion are allowed between low-distortion regions. Finally, we merge any matched regions in the adjacent diagonal bands that overlap in both the *x* and *y* directions (as shown in Figure 1). This allows the final path fragment to cross diagonal band boundaries, which may be important for longer words and phrases whose duration can vary substantially.

The above algorithm produces a set of matched fragment pairs, each with a known mean distortion. However, common words tend to create multiple overlapping fragments in the same utterance, each linked to a different instance of the word elsewhere in the data. For example, the words *it rolled away* might contain two fragments with slightly different start and end points, one linked to an utterance with another instance of *it rolled away*, and one linked to a different utterance with the words *rolled away*. Even if all three utterances contain the same words, the start and end points for each pair of matching fragments might not be exactly the same.

This proliferation of largely overlapping fragments not only seems cognitively implausible, but also complicates the clustering step of the algorithm, discussed below. We therefore refined the algorithm to include a fragment adjustment process in which temporally similar fragments are conflated so that each extracted word or phrase is associated with a single start and end point in the data. With this change, an incremental version of the algorithm can be developed which stores in memory only a single fragment for each extracted word or phrase, along with a small number of complete utterances inside the current processing window W. The final algorithm

processes the current utterance u as follows:

1. Match u against all previously extracted fragments. [1]
2. Match u against all previous utterances in W and against itself (disallowing overlapping fragments in the latter case). Extract only those fragment pairs where the fragment in the previous utterance does not clash with a fragment existing before step 1. Two fragments are deemed to clash if neither has at least a specified proportion (*min_frac_distinct*) of its duration outside the other, i.e., they have a high degree of overlap. (The rationale is that the relevant part of the previous utterance has already been matched against the current utterance in step 1.)
3. Sort the fragments from u in ascending order of mean distortion, and test for clashes between fragments; in case of a clash, adjust the start and end points of the higher-distortion fragment y to match those of the lower-distortion fragment x, and perform a new DTW match between x and the fragment or region to which y was originally matched. If y's matched fragment was from an utterance in W, then this new match may change the start and end points of that fragment, otherwise it just recomputes the mean distortion.
4. Repeat step 3 for the remaining utterances in W.

This algorithm is designed to operate with a limited window size, but can also be applied to the case with unlimited memory by setting the window size to include all previous utterances in the data set. In either case, the output is a set of matched fragment pairs with no clashes between fragments.

Clustering

Once the DTW matching has been completed for all the input utterances, the mean distortions are converted to similarity scores between pairs of fragments, and then the fragments are clustered using a graph-based clustering algorithm. The graph contains a node for each fragment, and an edge between each pair of DTW-matched fragments.[2] Each edge is assigned a weight according to the similarity between the pair of fragments it connects. Similarity is computed as

$$S(P) = (\theta - D(P))^2/\theta^2 \quad (1)$$

where $D(P)$ is the mean distortion for the DTW alignment path P between the two fragments (for $0 \leq D(P) \leq \theta$), and θ is a distortion threshold, so that paths with $D(P) > \theta$ are assigned similarity 0 or (equivalently) ignored in the clustering.

Given the weighted graph, we follow P&G in using the agglomerative clustering algorithm of Newman (2004). The algorithm works in a greedy fashion, at each step maximizing the increase in the modularity defined as

$$Q = \frac{\sum_i e_{ii}}{\sum_i a_i^2} \quad (2)$$

where e_{ii} is the fraction of all edges (weighted by strength) that connect nodes within the ith cluster, and a_i is the (weighted) fraction of all ends of edges that are attached to nodes within cluster i. The motivation for this measure is that $\sum_i e_{ii}$ is the proportion of all edges that are within clusters and $\sum_i a_i^2$ is the expected proportion of edges that would fall within clusters if the ends were connected at random. Thus Q is a ratio measuring how much better than random the fit between the current clustering and the edge strengths is.[3]

Information-Theoretic Evaluation Measure

Having defined our learning algorithm, we are left with the question of how to evaluate its performance. The most obvious measures include the number, sizes, and purities of the output clusters, and the proportion of frequently occurring content words that are found. While these measures are intuitive, for this task they involve some subjective decisions (e.g., how to define cluster purity given that some fragments in the clusters correspond to partial words, while others may be complete words). Moreover, they require examining the algorithm output and transcripts by hand, which slows down comparisons between versions of the algorithm during development.

To address these problems, we developed a new evaluation measure based on the idea that fragments within a cluster should be more phonemically similar to each other than to the average speech in the corpus. Put another way, we should be able to predict the phonemic content of a fragment better by using knowledge gleaned from other fragments in the same cluster. Specifically, we compute the phonemic entropy of each fragment both with and without using cluster-based information; the difference between the two gives a measure of the information provided by the clustering. We assume that a time-aligned phonemic transcription of the data is available for evaluation purposes (we use an automatic forced alignment in our experiments), and we compute the entropies of the phonemic transcripts of each extracted fragment. The entropy of a fragment *without* cluster information is computed using a phone bigram model trained on the full corpus. The entropy *with* cluster information is computed by estimating the probability of the fragment's phone sequence based on the phone sequences of the other fragments in the same cluster. The computations are described in more detail below.

Entropy without Cluster Information

Given a set of speech fragments with phonemic transcripts, we use a phone bigram model to compute the total entropy (negative log probability) of the transcripts without using cluster

[1] The utterance-to-fragment matching uses a form of DTW which finds, for each possible end frame in the utterance, the start frame achieving the best match between the interval [start,end] and the predefined fragment; a match is recorded at any local minimum of the resulting mean distortion that is below a specified value.

[2] Because we eliminated overlapping fragments in the matching phase, this method of graph construction is much simpler than the one presented by P&G.

[3] P&G used a slightly different Q measure, taking the difference between $\sum e_{ii}$ and $\sum a_i^2$ rather than the ratio. We found that using the ratio reduced the incidence of large clusters of low purity.

information. Specifically, we compute the bigram probability of the *i*th fragment transcript t_i, consisting of phones $x_1 \ldots x_m$, as $P_{bg}(t_i) = \prod_{j=1}^{m} P(x_j|x_{j-1})$. The probabilities $P(x_j|x_{j-1})$ are estimated from the corpus of phonemic transcripts, using smoothing to avoid setting any probability to 0. The entropy of t_i is then

$$H_{bg}(t_i) = -\log_2 P_{bg}(t_i) \quad (3)$$

and the total entropy of all fragment transcripts is $\sum_i H_{bg}(t_i)$.

Entropy using Cluster Information

To compute the fragment transcript probabilities (and thus entropies) using cluster information, we assume an ordering on the fragments in each cluster. We compute the probability of the first fragment using the bigram model above, and then compute the probability of each subsequent fragment using the transcript of the previous fragment, as described below. The resulting negative log probabilities are averaged across all possible orderings of the fragments within the cluster.

The model for predicting a phone sequence (the current fragment transcript) given another phone sequence (the transcript of the predecessor fragment) incorporates probabilities for all possible phone insertions, deletions, and substitutions (including substituting a phone for itself, which usually has a high probability). A recursion on possible alignments of the predictor and predicted phone sequences is performed to obtain the sum of the probabilities corresponding to all sequences of substitutions, insertions and deletions which transform the predictor sequence into the predicted one. The substitution, insertion and deletion probabilities are estimated from a corpus of transcript pairs representing within-cluster pairs of fragments, by an iterative process in which alignment of the transcripts (by dynamic programming) alternates with reestimation of the probabilities until the estimates converge.

When a cluster consists of fragments with identical or near-identical transcripts, the cluster-based prediction gives a higher probability than the bigram-based prediction; but when a cluster contains phonemically mismatched fragments the cluster-based probability can be substantially lower than the bigram-based one. To obtain a more robust prediction of the transcripts, we interpolate the cluster-based and bigram-based prediction probabilities, yielding the following expression for the entropy of the *i*th fragment transcript:

$$H_{cl}(t_i) = \frac{1}{n}(H_{bg}(t_i) + H_{tot}(t_i)) \quad (4)$$

with

$$H_{tot}(t_i) = \sum_j \left(-\log_2(\alpha P_{al}(t_i|t_j) + (1-\alpha)P_{bg}(t_i)\right) \quad (5)$$

where n is the size of the cluster containing t_i, α is an interpolation weight, $P_{al}(t_i|t_j)$ is the probability of t_i obtained by deriving it from t_j using the align+edit method described above, and j ranges over the $n-1$ other fragments in the cluster.

Entropy Reduction

For fragment transcript t_i, the reduction in entropy obtained by using the clustering information to augment the baseline bigram model is given by

$$H_{bg}(t_i) - H_{cl}(t_i) = \frac{n-1}{n}\left(H_{bg}(t_i) - \frac{H_{tot}(t_i)}{n-1}\right), \quad (6)$$

with the total entropy reduction over a set of utterances being the sum of the above quantity over all the extracted and clustered fragments. Expressing the reduction in this form makes clear its dependence both on the consistency of the transcripts within the cluster (which will tend to increase the probabilities $P_{al}(t_i|t_j)$ from Eq. 5, in turn increasing the parenthesized factor in Eq. 6) and on the cluster size n (which, as it increases, will push the $(n-1)/n$ factor closer to 1). Internally consistent clusters are better than inconsistent ones; and, for a given level of within-cluster consistency, large clusters are better than small ones, so that it is better to generate a single cluster containing (n_1+n_2) instances of the same word or phrase than separate clusters of n_1 and n_2 instances.

Experiments

Data

The data for our experiments comes from the Brent corpus (Brent & Siskind, 2001) in the CHILDES database (MacWhinney & Snow, 1985), which consists of recordings of mothers speaking to their infants (aged nine to 15 months) in a naturalistic setting. We used the "Brent33" subset of the corpus defined by Rytting (2007), which contains 7811 utterances from 15 recording sessions (three or four from each of four mother-infant dyads). This subset also contains a forced time-alignment of the audio to a phonemic transcript, which was produced by Rytting (2007) and which we use for evaluation purposes. The forced alignment uses a standard American English phone set, except that voiced and unvoiced closure are treated as phones in their own right, yielding transcripts such as /`l eh cl s vcl g ow`/ (*let's go*).

The acoustic feature representation used in segmental DTW consisted of a vector of 12 mel cepstral coefficients (computed in a 20ms window) and 12 delta coefficients every 5ms. The delta coefficients accompanying the cepstral vector c_n at frame n were derived as $0.3 \times (c_{n+1} - c_{n-1})$.

Procedure

We ran our algorithm separately on the data from each session, using the following parameters: *core_thr* = 1.2, *min_path_len* = 90 (i.e., 0.45 seconds), *max_hi* = 2, *min_frac_distinct* = 0.5, θ = 0.6. This yielded a set of clustered fragments for each session, which were transcribed using the forced alignments. A phone spanning the beginning or end of a fragment was included in its transcript if at least half of the phone's duration was within the fragment or if (as occurred for some very long phones) the phone contained the whole fragment.

For the entropy reduction measure, probabilities for the bigram-based and cluster-based prediction models were estimated per dyad. The bigram estimation data consisted of the

Table 1: Per-phone entropy reduction results for different window sizes w on the original and randomly permuted corpus.

Condition	Per dyad				Overall
	c1	f1	f2	q1	
$w = \infty$	.0486	.0077	.0187	.0000	.0231
$w = 20$	.0401	.0067	.0135	.0000	.0185
$w = 10$	.0373	.0075	.0131	.0000	.0177
$w = 5$	.0341	.0037	.0117	.0000	.0153
$w = 2$	.0269	.0043	.0091	.0000	.0124
$w = 20$, perm.	.0126	.0000	.0001	.0000	.0040
$w = 10$, perm.	.0078	.0002	.0001	.0000	.0025
$w = 5$, perm.	.0079	.0002	.0001	.0000	.0025
$w = 2$, perm.	.0075	.0000	.0000	.0000	.0023

phonemic transcripts of all the utterances. The cluster-based estimation data consisted of the transcripts of the extracted and clustered fragments, combined into all possible within-cluster pairs, with weight $1/(n-1)$ on each pair in a cluster of size n. The interpolation weight α was set for each dyad so as to minimize the total entropy $\sum_i H_{cl}(t_i)$.

Results and Discussion

Table 1 shows results for a fixed window size w ranging from 2 to 20, as well as for $w = \infty$ (batch processing: each utterance is matched against all other utterances in the session). For comparison, we also show results for $w = 2$ to 20 when the order of utterances is randomly permuted within each session. For each condition, entropy reductions are shown for each of the four dyads (c1, f1, f2, and q1) and overall, normalized by the total numbers of phones in the utterances.

Several trends are worth noting. First, although the results with fixed window sizes are not as good as the batch processing algorithm, the difference is not great, especially for the larger window sizes. Results on the permuted corpus are uniformly bad, with results for $w = 20$ generally worse than even the smallest window size ($w = 2$) on the corresponding correctly ordered corpus. These results are consistent with the hypothesis that frequent nearby repetitions in infant-directed speech are useful for extracting words, especially for an incremental learner. Unfortunately, due to the small number of dyads, the statistical significance of these results is weak. The recordings for dyad q1 are somewhat noisier than the others, which may explain the null results on this dyad.

Although we cannot draw statistical conclusions from the entropy reduction numbers, we can gain further insight into the algorithm by examining its output in more detail. Table 2 shows the clusters of size ≥ 3 obtained in the $w = 10$ condition from a typical c1 session (containing 547 utterances; entropy reduction = .0273 per phone). The word or phrase shown in the first column is the most frequent word or phrase in the cluster; the purity is the percentage of fragments matching this word or phrase. As noted above, purity scores necessarily involve subjective judgements since fragment start and end points may not correspond exactly to word boundaries or to other fragments in the cluster. Scores in brackets indicate that a substantial part of the word or phrase is missing from some or all of the fragment transcriptions.

Table 2: Clusters of size ≥ 3 from dyad c1, session 4.

Word(s)	Purity	Segment transcriptions
look at the	67	look at the look at the look at the -s look at the you're just taking book is tha-
sweetie *or* Mommy	33	sweetie sweetie -e page -ng swee- -t's Mommy go- -t's Mommy
yeah	100	yeah is th- yeah that's yeah d- yeah yeah is th-
[that's the yellow]	[60]	-t's the ye- -'s the ye- -t's the ye- -t's play a g- -t's a pret-
-	-	oh you s- uh-oh what do we turn the -th Mommy
[fun]	[40]	do- fu- fu- down -s and lo-
book	100	book book book books
sweetie	100	-ng sweetie -ng sweetie a sweetie -ng sweetie
Morgan	100	-t Morgan Morgan Morgan Morgan
read	50	Morg- -n't w- gonna read th- -n you read th-
on	50	-elf h- -ay -ot on on
[points]	[100]	-oints -oints -oints -oints
[book]	[100]	a boo- -n's boo- -n's boo- -ther boo-
[lots]	[50]	-own off the sh- -ots a- and lots of -ong swee-
yeah	100	yeah yeah yeah i-
kitty-cat	100	kitty-ca- kitty-ca- kitty-ca-
ball	100	ball ball -ow ball
points	100	points points points
two points	100	two poi- two poi- two poi-
yeah	100	yeah yeah yeah
books	67	books d- what fu- books d-
[doggie]	[100]	-gie -gie -gie
[what's in here]	[67]	yeah -'s in he- -'s in he-
[yay]	[67]	hea- -ay -ay

Looking at Table 2, we can see that although the algorithm does not extract a large number of words (either types or tokens), most of the clusters it finds are lexically consistent, i.e., have a high purity. This is true also of the 32 clusters of size 2 found in this session (not shown for reasons of space), of which 23 (72%) had a consistent lexical or phrasal identity. The algorithm is not entirely successful at pinpointing the exact start and end points of the words, but does surprisingly well given the unconstrained nature of the task. It is also worth noting that of the 14 content words that occurred more than 10

times in the data for this session, the algorithm detected at least some occurrences of 10 of them. As far as the particular words that are found, we see that some words (e.g., *book* and *ball*) reflect specific activities during the session, whereas others (e.g., *Mommy* and the child's name *Morgan*) are common to many sessions' results. Out of the 15 total sessions, the child's name was detected in 11 of them, and the words *Mommy* or *Mama* in eight. This is particularly notable since these words constitute some of infants' earliest vocabulary, and have been shown to help them segment other words as early as six months (Bortfeld et al., 2005).

General Discussion

The algorithm presented here represents one possible way to begin the process of extracting and learning words from continuous speech. As an incremental algorithm, it is more cognitively plausible than the original P&G version, and we have shown that it is sufficient to extract at least a small number of high-frequency words. Like the models of Aimetti (2009), Driesen et al. (2009), and Räsänen (2010), it operates directly at the acoustic level, without using intermediate-level units such as phonemes or syllables (in contrast to the algorithm of Neubig, Mimura, Mori, and Kawahara (2010), for example); this accords with the observation that familiarity effects in infant listening behavior appear to operate at a whole-word rather than subword level (Jusczyk, Houston, & Newsome, 1999). Unlike these previous models, however, ours has been shown to work on real child-directed speech, not just hand-built test corpora. It is selective rather than exhaustive, in the sense that it extracts selected intervals as instances of recurring patterns rather than attempting a complete segmentation of the input utterances; this seems plausible as a model of word segmentation in the early stages of learning, when only a few words are known. The patterns discovered are a mixture of words, parts of words, and sequences of words or part-words; further processing, perhaps using relative frequencies and partial similarities between patterns, would be required to distinguish the words from the other patterns.

As developed here, our algorithm is exemplar-based: all the extracted fragments are individually stored and compared against each new utterance. Thus, although only a fixed number of utterances are held in memory, memory and processing requirements still grow fairly rapidly with the amount of data. One way to reduce these requirements would be to develop a more compact representation of the extracted fragments, similar to a prototype-based model. This could be achieved either by creating a statistical model for each pattern as soon as it is discovered (combining the information from the initial two instances of the pattern) and then incorporating further instances into the model; or by starting with exemplars (as now) but then deriving a single model for each pattern when sufficient evidence has accumulated.

Acknowledgments

We thank Anton Rytting and John Pate for providing the forced alignments, and Oliver Watts for help with signal processing.

References

Aimetti, G. (2009). Modelling early language acquisition skills: Towards a general statistical learning mechanism. In *Proceedings of the student research workshop at EACL.*

Bortfeld, H., Morgan, J., Golinkoff, R., & Rathbun, K. (2005). Mommy and me: Familiar names help launch babies into speech stream segmentation. *Psychological Science*, *16*(4), 298–304.

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

Brent, M., & Siskind, J. (2001). The role of exposure to isolated words in early vocabulary development. *Cognition*, *81*(2), B33–44.

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

Driesen, J., ten Bosch, L., & Van hamme, H. (2009). Adaptive non-negative matrix factorization in a computational model of language acquisition. In *Proceedings of interspeech.*

Goldwater, S., Griffiths, T., & Johnson, M. (2009). A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, *112*(1), 21–54.

Jansen, A., Church, K., & Hermansky, H. (2010). Towards spoken term discovery at scale with zero resources. In *Proceedings of Interspeech* (pp. 1676–1679).

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

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

Myers, C. S., Rabiner, L. R., & Rosenberg, A. E. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*, 623–635.

Neubig, G., Mimura, M., Mori, S., & Kawahara, T. (2010). Learning a language model from continuous speech. In *Proceedings of Interspeech* (pp. 1053–1056).

Newman, M. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E*, *69*(066133).

Park, A. S., & Glass, J. R. (2008). Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech and Language Processing*, *16*, 186–197.

Räsänen, O. (2010). Fully unsupervised word learning from continuous speech using transitional probabilities of atomic acoustic events. In *Proceedings of interspeech.*

Rytting, A. (2007). *Preserving subsegmental variation in modeling word segmentation (or, the raising of Baby Mondegreen)*. Unpublished doctoral dissertation.

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

Thiessen, E., & Saffran, J. (2003). When cues collide: Use of stress and statistical cues to word boundaries by 7- to 9-month-old infants. *Developmental Psychology*, *39*(4), 706–716.