Audio Speech Segmentation Without Language-Specific Knowledge

Kevin Gold (kevin.gold@yale.edu) and **Brian Scassellati (scaz@cs.yale.edu)**
Department of Computer Science, 51 Prospect Street
New Haven, CT 06520 USA

Abstract

Speech segmentation is the problem of finding word boundaries in spoken language when the underlying vocabulary is still unknown. Here we show that a system with no phonemic knowledge can find word boundaries. The system first subdivides an utterance by recursively clustering similar parts of the signal together until the cepstral coefficient variance is low within each new segment. These segments are then used as inputs to a perceptron-like algorithm that finds repeated segments across utterances. With only a few sample utterances, and no previous linguistic knowledge, the system can find the words that were repeated across utterances and identify new utterances that contain those words. The findings show that the assumption of a phoneme classification module is not necessary for a "minimum description length" (Brent & Cartwright, 1996; de Marcken, 1996) explanation of word segmentation.

Introduction

The problems of infant speech segmentation and word discovery have inspired many different modeling approaches, but some of the most promising have operated under the principle of "minimum description length" (MDL). Minimum description length approaches combine aspects of exemplar and prototype theory to create a hierarchical language model that extends from phonemes all the way up to the phrase level (de Marcken, 1996; Brent & Cartwright, 1996). Essentially, a minimum description length approach attempts to compress all the utterances it has encountered so far by adding entries to a lexicon for any item that is repeated so often that it warrants a kind of "mental shorthand." The new entry is only added if the cost of storage is less than the cost of representing each instance of the item individually. Shorthand can be used within the lexicon as well, making for hierarchical compression: a phrase is represented as a sequence of words, which are in turn represented as a sequence of morphemes or syllables, which are in turn sequences of phonemes. In MDL models, word discovery is only a special case of recognizing common subsequences of input.

One gap in the minimum description length approaches proposed so far is that they have all treated the segmentation problem as if it were being solved for sequences of symbols (Brent, 1999). Usually, these symbols represent transcriptions of either syllables (Brent & Cartwright, 1996; Brent, 1999) or the phonemes from which the syllables are constructed (de Marcken, 1996). Using transcriptions of speech is convenient for the modeler, but they come at the price of assuming a "perfect" input processing module that can cluster and classify the raw audio signal, turning a noisy signal into a flawless symbolic representation.

This initial classification of phonemes or syllables can be difficult. Extensively trained neural networks can still perform quite poorly in classifying phonemes (Roy & Pentland, 2002). A phoneme can be so influenced by the surrounding phonemes that the signal resembles a different phoneme entirely; for example, the waveform of the /ɪ/ in "king" can more closely resemble that of the /u/ in "moves" than the /ɪ/ in "bishop," on account of the nasality of the /ŋ/ sound that follows (Jelinek, 1997). Professional speech recognition software solves the difficult phoneme-recognition problem by using contextual information to distinguish between words – for instance, by noting that the probability of *I need* is much higher than that of *I neat* (Jurafsky & Martin, 2000). Such contextual information is certainly not available to an infant if phoneme learning precedes word learning.

Transcriptions can also omit important information that is available from the audio signal. For instance, while it is a commonly accepted maxim in word segmentation models that the spaces we perceive between words are simply an illusion, stops such as /t/ and /k/ literally stop the flow of air briefly, making them natural delimiters. The knowledge that such consonants tend to delimit words and syllables may be captured by adding "phonotactic constraints" to a segmentation algorithm, but this can obscure the difference between rules that must be learned and rules that follow naturally from the structure of the signal. Another cue for segmentation lost in phonetic annotation is syllable stress, which may play an important role in segmenting English (Jusczyk, 1999; Cutler & Norris, 1988). The rule that each word in English can have at most one primary stress can do much of the work in segmenting child-directed speech, allowing easier segmentation of utterances with many strongly stressed syllables (Yang, 2004).

Though the minimum description length approaches proposed so far have all been demonstrated on phonetic transcriptions, there is no reason why they cannot be extended all the way down to the sub-phoneme level. The system presented here uses a minimum description length approach to cluster the audio signal into self-similar parts. These parts are then shown to be usable to find words in utterances, assuming the existence of external cues that indicate which utterances contain a target word. Since this approach works with the audio signal itself, several sources of information such as volume, phonotactic information, and coarticulation are naturally factored into the system's segmentation decisions. The system shows how a simple compression rule can take the place of multiple cue-specific rules, and suggests a synthesis between MDL approaches and auditory cue-based explanations of infant word segmentation.

Methods

Acoustic Data

For our experiments, the auditory data is not processed in the time domain, but in the "cepstral" domain (Schroeder, 1999). Since the audio signal can be seen as the result of a source signal coming from the vocal tract convolved with the filter action of the mouth and tongue, the goal of the cepstral transformation is to extract the part of the signal corresponding to the mouth and tongue while throwing out the variability of the individual vocal tract. This is accomplished by taking the log of the Mel-transformed frequency domain representation, then taking another Fourier transform. Under these transformations, convolution in the time-domain becomes multiplication in the mel frequency domain, which then becomes addition by the logarithm, and the final Fourier transform preserves this additivity while separating the two parts of the signal. The parts of the cepstrum relevant to phonetic information can then be characterized by a feature set called the "Mel-Frequency Cepstral Coefficients" (MFCCs).

Note that these coefficients do not represent any language-specific features. They are a more or less arbitrary decomposition of the signal into coefficients that could be used to reconstruct the spectral envelope that characterizes the shape of the mouth (though the first coefficient does correspond to overall power or volume). However, this representation does throw out the pitch and timbre information. Though we can justify this neurologically with the finding that pitch appears to use a different neurological pathway from other aspects of language (Baum & Pell, 1999), in truth we use the cepstral domain here mostly because automatic speech recognition methods have had better success with it than not.

For this study, 13 cepstral coefficients were used, a common convention that corresponds roughly to the degrees of freedom of the human mouth (Schroeder, 1999). The coefficients were estimated for 10ms intervals, using freely available MATLAB code (Ellis, 2005).

As a final, additional feature set, approximations to the derivative and second derivative of each cepstral coefficient were added, bringing the total number of features to 39. The approximations used for these "delta" and "double-delta" coefficients were taken from the Sphinx speech recognition system (Walker et al., 2004): the "delta" feature for cepstral coefficient C_j at time i is given by the difference $C_j(i+2) - C_j(i-2)$, and the double-delta coefficient is given by the formula $(C_j(i+3) - C_j(i-1)) - (C_j(i+1) - C_j(i-3))$. (Neurologically, auditory "edge" detectors have been proposed for spectral information (Fishbach, Nelken, & Yeshurun, 2001), but the authors know of no studies that have attempted to find neurons responsive to the cepstrum or its derivatives.)

Clustering

After preprocessing, an utterance is recursively clustered as follows. First, the loudest 10ms sample of the utterance is chosen as a seed for a new cluster. The selected vector of cepstral coefficients and their derivatives serves as the first data point in the new cluster's multivariate normal distribution. All the remaining samples of the utterance are treated as samples of a single multivariate background distribution. To ensure that the probability density function of this distribution is always well-defined, only the diagonal of the covariance matrix is used.

A cluster is grown by tentatively adding the time slice immediately preceding it and the time slice immediately following it to both the cluster's distribution and the background distribution. (This ensures that the distributions are not too heavily biased toward their existing states, and also results in each cluster having at least three points for calculating variance.) The probability density function (pdf) of each distribution is evaluated at both of these new points. If a time slice is determined to be more likely to have been generated by the tentative cluster than the background distribution, the slice permanently remains in the new cluster and is removed from the background distribution. Otherwise, it is removed from the cluster and remains in the background distribution. This process continues until the cluster ceases to grow.

This process produces the maximum likelihood clustering of the data under two assumptions. The first is that the data was generated from two multivariate normal distributions, one of which contains the loudest time slice. This normality is only an approximation, but it is one that has proven successful in the past (Wilpon, Rabiner, Lee, & Goldman, 1990). The second is that all examples of the louder distribution are contiguous.

Once this initial clustering has been performed, the clustering can be performed recursively on each of the three new segments of speech. Should this recursion occur indefinitely, each individual time slice would eventually receive its own cluster. Obviously, this would not achieve any compression of the data, and the Minimal Description Length principle (Rissanen, 1972; Grünwald, 2005) dictates that good learning implies good compression and vice versa. Instead, a stopping criterion is required, so that the algorithm does not make useless distinctions.

By the Minimal Description Length principle, an ideal stopping criterion would determine the cost, in bits, of specifying a new multivariate distribution and all of the data points belonging to it versus the cost of specifying all of those time slices in an already defined distribution. (Bits are here being used in their information-theoretic capacity; it does not limit the argument to machine learning.) Since good compression implies good future learning (Grünwald, 2005), the strategy that takes fewer bits to specify would be better for future language learning.

Determining the true cost in bits of a new multivariate normal distribution is difficult. While it is possible to choose an arbitrary representation for the distributions and count the bits they use, this is not necessarily the most efficient encoding, which should take into account the probabilities of each distribution's parameters. Specifically, the desirable mathematical properties of minimal description length representations are only guaranteed to hold when the cost of a given distribution D, in bits, is equal to $1/P(D)$, where $P(D)$ is the probability of that distribution. The total cost of the encoding is then the sum of $1/P(D_i)$ for all distributions D_i plus the sum of $1/P_{D(j)}(S_j)$ for each sample S_j, where $P_{D(j)}(S_j)$ is the probability a sample S_j according to its assigned distribution $D(j)$. (Even this "two-part encoding" is not necessarily optimal; see (Grünwald, 2005) for details.) Though it is easy to calculate this second part of the equation, the cost of encoding a sample given a distribution, it is hard to determine the probabilities of the distribution parameters themselves.

Instead, the algorithm takes a shortcut here and uses a variance threshold. Specifically, when the variance in the first cepstral coefficient (volume) for a given segment is below a manually defined threshold, recursion stops. The reason that this approximates an MDL criterion is that with a low variance, each individual segment can be encoded using fewer bits. By varying the threshold at which a new distribution is generated, the algorithm can mimic a higher or lower cost in bits of generating a new distribution, assuming that this cost is roughly uniform for most real-life distributions. Limiting the model to a single parameter, volume variance, rather than a function of all 39 variances, is a tradeoff that reduces the number of model parameters that the modeler must adjust in exchange for giving up some small amount of expressiveness. Volume variance is the most useful of the 39 features from a practical point of view because volume changes from phoneme to phoneme and from syllable to syllable. Vowels are louder than the consonants that delimit them, and syllables can be stressed or unstressed. The overall approach is thus similar in spirit to previous minimal description length approaches to the word segmentation problem (de Marcken, 1996; Brent & Cartwright, 1996; Brent, 1999), but it attempts to find a minimal representation of the audio signal, instead of a string of symbols. In the long term, existing entries in the lexicon for distribution parameters would reduce the encoding length of most new incoming audio data, with only unusual sounds requiring full encoding. The means of these distributions in the lexicon could serve as prototypes for phonemes, and also serve as symbols in an MDL-based lexicon for words.

See Appendix A for transcriptions of the output produced by the self-similarity clustering step.

Learning across utterances

The clustering method outlined above segments an individual utterance into a tree structure based on self-similarity. Later utterances are divided into their own trees, but these need to be matched to earlier utterances to find repeated words. Matching currently occurs at the leaves of the recursion tree, corresponding roughly to the phoneme level.

To find segments that are similar within two different utterances, pairwise comparisons are made between segments in the two utterances to find matches for the new clusters among the old ones. This is done by taking the mean of a new cluster to be a sample point, and computing the probability density function of each cluster in the reference utterance for that point. The distribution with the highest probability density function is the most likely match. This match is then compared to the null hypothesis that the new cluster matches no previously established point, which is represented by a multivariate distribution of all data from both utterances. If the probability of the old cluster exceeds that of this null hypothesis, the new cluster is classified as an instance of the old one. (A lexicon independent of all utterances would have been more elegant than pairwise comparisons between utterances, but this would have brought up encoding representation issues we do not wish to deal with here.)

In the case of learning a specific word for an object, associative learning can take place between the stimulus of the object and certain combinations of speech clusters. To achieve this associative learning, the present implementation used Winnow (Littlestone, 1988), an online learning algorithm that performs well when the number of input variables is large. Winnow is an online perceptron-like algorithm that attaches linear weights to nonnegative input stimuli and "fires" if the sum of the weighted inputs passes a threshold. If a false positive occurs, the weights of all positive inputs are halved; if a false negative occurs, the weights of all positive inputs are doubled. Since Winnow is an online algorithm, it does not require multiple passes through the data. It is only necessary to process each utterance once.

The Winnow weights can be interpreted as the strength of association between the segments found by the segmentation algorithm and some external cue that the system determines to be a referent for the sentence. For example, we assume that when the system hears "This is a dog" that somebody is directing the system's attention toward an actual dog, or at least is clearly talking about a dog, so that the segments found in the sentence can be associated with that external stimulus. When Winnow fails to predict the referent of an utterance (e.g., somebody is clearly talking about a dog but the utterance was not identified as containing a word about a dog), weights between the target concept and all segments contained in the utterance are boosted. Likewise, if Winnow expects a certain referent given the segments in the sentence, but there is no external confirmation (e.g., the concept of "dog" was excited but the speaker was clearly talking about something else), the weights to the excited segments are halved. For each target concept, there is an instance of Winnow attempting to learn the word for it. (We set aside for more knowledgeable scholars the question of if and how an infant decides a concept needs a word.)

In the results that follow, we abstract away the external referents, and assume that the system has access to reliable external cues that indicate which sentences contain a word related to a target referent. This means utterances are labeled only as positive or negative examples for concepts and the words that correspond to them.

Note that though this approach uses "utterances" as a basis for learning, it does not require that the system be able to segment sentences from one another. The assumed paradigm here is that of examples of speech separated by relatively long gaps. If the speech separated by these gaps includes more than one sentence, this does not matter much, except that very long utterances will end up conveying very little useful information. At any rate, the emphasis here on "utterances" is more an artifact of programming convenience than any theoretical justification. The method should be extendable to a real-time system that acts on a stream of input.

The system currently has no principled method of representing cluster order; during production, it simply uses the order in which the reference clusters were first encountered. In principle, lexicon entries would contain such information, but in this experiment this functionality simply wasn't implemented.

Experimental Results

Thirty training utterances were recorded from a single speaker at 22050 Hz. Ten target words for segmentation were contained in three utterances each. For each target word, the utterances containing the word were treated as positive examples for Winnow, while the other utterances were negative examples. The list of utterances is given in Appendix A.

Winnow initialized weights to all segments to 1, and used a threshold of 1 instead of the standard $n/2$ to increase the usefulness of negative examples. A volume variance threshold of 50 was used for segmentation recursion.

The three positive examples were each presented to the system twice, while the twenty-seven negative examples were each presented once. The goal was to determine whether the system could learn to segment the target words with a relatively small number of utterances, as a child might.

To probe the system's representations, "best guesses" were generated for each target word by concatenating cepstral sequences corresponding to the clusters with the highest weight until their combined weight passed the Winnow threshold. The cepstral sequences were not generated randomly from the cluster distributions, but preserved from the first utterance in which the cluster appeared. These cepstral sequences were then transformed into spectral envelopes imposed over a white noise source, using software from (Ellis, 2005) to regenerate the speech signal. Table 1 shows the transcription of the system's output.

Target	Guess	Target	Guess
bɑl	bɑl	kiz	i
bʊk	bʊk	pɛn	maɪ
kar	kar	fon	maɪn
tʃɛir	ðɪs...eɪr	ʃu	ʃu
dɔɡi	gi	spun	spu

Table 1: Transcriptions of the "best guesses" generated for each target word by Winnow.

	Recall	Precision
Recursive clustering	40%	50%
Non-recursive clustering	20%	33%
Hidden Markov Model	60%	32%

Table 2: Recall and precision for each of the methods implemented.

To test the system's ability to detect words in new utterances, the Winnow-trained word detectors were then employed on utterances that either contained the target word but were new to the system, or contained none of the target words. Each target word was contained in one test sentence, and five sentences that did not contain any target words were tested for each target word.

Table 2 shows recognition results for three variants of the algorithm. The recursive variant is the one described above. In the non-recursive variant, the clustering step is performed only once, so that the sentence is divided into only three segments. The non-recursive method was implemented to check whether fine distinctions in sound were actually necessary for recognition, or if the overall vowel sound of the target word would be sufficient for classification; as the results show, the recursive decomposition does aid recognition. Finally, the hidden Markov model (HMM) implementation used the clusters generated from the self-similarity step to generate hidden Markov models (Jurafsky & Martin, 2000) for each statement, then used the model that best described all three positive examples as a model for the target word. The added complexity of performing expectation maximization and the Viterbi algorithm did not appear to afford much benefit over the simpler maximum likelihood matching described above.

Discussion

The present work combines several previous approaches to automated word segmentation. First, the clustering method presented here to find segments is somewhat similar to a method used by Tim Oates' system PERUSE to find recurring segments of audio (Oates, 2002), though PERUSE was not an online algorithm. The minimum description length literature, on the other hand, has always used phonemes as its atomic units, and built syllables, words and phrases from these (de Marcken, 1996; Brent & Cartwright, 1996). Finally, the Winnow algorithm is here used to simulate association between words and external concepts. Though no previously proposed algorithm has used Winnow specifically, others have shown that mutual information between phonemes and visual cues can aid segmentation (Roy & Pentland, 2002).

The Winnow algorithm here acts as a somewhat temporary measure that takes the place of a more MDL-based word learning approach. In theory, external meaning should be representable in an MDL fashion, as should the higher levels of organization beyond phonemes (de Marcken, 1996). However, no MDL word segmentation algorithm to date has been an “online algorithm,” as the major MDL approaches to date have required multiple passes through the data (de Marcken, 1996; Brent & Cartwright, 1996). The Winnow perceptron algorithm is interesting because it is an online algorithm that works very quickly to perform the necessary associations, and it provably functions well when irrelevant features abound (Littlestone, 1988). Its lack of a hidden layer also makes it more neurally plausible than most neural networks, since the backpropagation algorithm is not required.

The minimum description length approach to word segmentation is attractive because it can take into account many disparate psychological findings and theories (Brent, 1999). Though infants have been interpreted as remembering phoneme transition probabilities (Saffran, Aslin, & Newport, 1996), the same results may be explained by positing the formation of lexicon entries for the more common phoneme pairs. The existence of “episodic” memories for words (Goldinger, 1998) can be explained as the brain’s attempt to retain all possible information for compression; while a prototype effect would result from encoding the mean of many clustered words or sounds, individual instances should still be retrievable if it only requires a few more bits to specify an instance – though if memory is scarce, the brain may maintain a low error rate by discarding these few bits. Unusual words or phrases may be more cheaply encoded in their entirety; hence the episodic memory for the particular prosodic affect that accompanies *Rosebud* or *Stella* (Goldinger, 1998).

One of the original hopes of this project was that the recursion tree might also provide a natural structure for higher level MDL learning, with syllables clustered naturally into words by sound similarity and coarticulation effects. For instance, the self-similarity algorithm clusters “night rate” as [naɪt]ret, separating “night” from “rate” and then subdividing “rate” into its constituent sounds, while “nitrate” is clustered as naɪ[tre]t. The transcriptions in Appendix A reveal that obtaining higher structure from the auditory signal is often not so simple, as words can be spread across several subtrees. Still, in 25 of the 35 utterances, all parts of the noun were clustered at the same level of recursion, which would mean Winnow learning on the higher nodes of the tree may have proven useful. Some interior nodes contained two related subtrees and an unrelated one, suggesting that a binary tree, rather than the current ternary structure, may be more appropriate for such learning.

Self-similarity segmentation tends to split stop phonemes down the middle, where the airflow is stopped. A human listener can clearly make out the phoneme both before and after the break because of coarticulation effects, but only one side of the segmentation can be technically correct. The fact that the self-similarity algorithm was successful at learning words with consonants that were divided in this way suggests that some consonants may be better thought of as auditory “edges,” a point suggested by quantal theory (Stevens, 1989).

Comparison with prior word segmentation methods is difficult because no previous method solved quite the same problem. PERUSE (Oates, 2002) reported successful segmentation of 65% of the most frequent words it encountered, but it used an expectation maximization procedure that required iterating repeatedly over the entire data set, making it unlikely to scale. CELL (Roy & Pentland, 2002), an online system, reported 54% segmentation accuracy on its highest-ranked word candidates, compared to our algorithm’s 40% segmentation accuracy for our ten target words, but CELL used an extensively trained and hand-modified phoneme classifier as input to its segmentation module. To our knowledge, no previous work has both avoided using a phoneme classifier and used a single-pass algorithm for segmentation.

The present data set is small, and is not entirely representative of a broad range of words and circumstances. The utterances were only produced by one speaker, tended to have the target word in the final position, and only included one multisyllabic word and no vowel-initial words in the production and recognition tests. Future work will address these concerns; meanwhile, we hope that Appendix A will convince the reader that even the samples used contained a fair amount of diversity.

Another question that would be worthwhile to address is whether this system will reproduce some of the common mistakes or biases of children learning new words. For example, if unstressed syllables appear first in a sentence’s final word, they may be incorrectly grouped with the linking words and left unparsed until late in the recursion. If the target word begins with a strong syllable, however, the remaining weak syllable is left relatively high in the parse tree. A learning algorithm that takes tree distance into account should therefore display the bias of children for learning words that begin with strong syllables more easily than those that begin with weak syllables, as has been observed in children learning English at 7 $\frac{1}{2}$ months (Jusczyk, 1999).

Though attempting to explain infants’ word segmentation abilities with one grand theory of everything may be an impossible task, a minimal description length approach at all levels, from sentence to sub-phoneme, is appealing in its elegance. Implementing it and determining what effects it does and does not explain may be the best way to create a theory of word segmentation that is itself both concise and general.

Appendix A: Transcriptions of Segmentations

The following are phonetic transcriptions of the initial segmentations that the system performs on the 35 utterances in the experiment before Winnow is applied, using only intra-utterance self-similarity measures. Straight brackets indicate the loudest part of the utterance, or top level of recursion; parentheses indicate deeper levels of recursion. In places where a phoneme sounds is split, the corresponding symbol is repeated on either side of the split. All phonetic judgments were made post hoc for transcription only; the system segmented audio without a phoneme classifier.

ðɪs(s (ɪz ə) b)[bɑl].
yə wɑnʌ p[pleɪ wɪ (ðə b)bɑl]?
tra(j) ɾə (kɑ)tʃ ðə b[bɑl]!
(test) ɪ(z) [ðæɾ ə](bɑl)?
ðɪs iz ə b[bʊ]k.
s(i)? ɪts ə b[bʊ]k!
yə wɑn[nə r(id) ðə (bʊ)k?
(test) ɪz [ðæt ə] b(bʊ)k?
ðæts ə (k)[kar]r.
s(i) ðə k[kar] ro(l)?
[yæ], ɪt(s ə) k(ar)r!
(test) z (ð æɾ ə) k[kar]r?
ðɪs(s (ɪ)z ə (tʃ))[eɪr]r.
pi(p)pl s(sɪɾ) (ɪn) tʃ[eɪr]z.
w[eɪr]z (yor (tʃ))eɪr?
(test) z [ð æɾ ə] (tʃ)eɪr?
ɪts ə d[dɔgi]!
ðæt(s (r))(aɪt), ə d[dɔg]i!
yə wanə (pɛt (ðə d))[dɔ]gi?
(test) s (ð æɾ ə) d[dɔ]gi?
ð[owz ər maɪ] k(ki)z.
(yə wanə) p[pleɪ wɪ] ðow(z k(i)z?
aɪ (ʃ)[r h æ]v ə lɔt ə (k(i)z).
(test) (ar) ð[owz maɪ k](i)z?
ðɑts (maɪ) p[ɛ]n.
((si) ðə) p[ɛn]?
tr(aɪ) dr[raɪŋ] s(əm)θ(ɪŋ) (wɪ)θ (ðɑʔ) (pɛ)n.
(test) ɪz [ð æɾ ə] pɛn?
ð æ(tʃ) (maɪ) f[o]n.
yə wɑnə (c (ɑ))[ɑl] sʌm(wən ɔn) m(aɪ) f(o)n?
(yæ, yu) k (æn)(pleɪ wɪ)θ (maɪ) f[o]n.
(test) ɪz (ðæ) m[maɪ] f(o)n?
[ðæ](ts ʌ ʃ)u.
s(i) [ðə ʃu]u?
[yæ], ɪt(s (ə) ʃ)(u)u!
(test) ɪz ðæt ə ʃ[u]?
[ðɪs ɪs]z ə (s)p(pun).
s(i)? (yu) it [θɪŋ]z (wɪθ) ə (s)p(pun).
ðæt(s(r))[aɪ]... s(p)(pu)n.
(test) ɪz [ðæɾ ʌ] sp(pu)n?
(test) ɪz (ðæɾ ɛn) ɛ[r]p(le(in))?
(test) ɪz ð(ðæɾ ə) b[ɑɾl]?
(test) z [ðæɾ ə] kʊ(k(i))?
(test) ɪz [ð æɾ ə] h(or)s?
(test) [ɪz ð æɾ ə] (k(ɪɾi))?

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful and extensive feedback. Support for this work was provided by a National Science Foundation CAREER award (#0238334). Some parts of the architecture used in this work was constructed under NSF grants #0205542 (ITR: A Framework for Rapid Development of Reliable Robotics Software) and #0209122 (ITR: Dance, a Programming Language for the Control of Humanoid Robots) and from the DARPA CALO/SRI project.

References

Baum, S. R., & Pell, M. D. (1999). The neural bases of prosody: Insights from lesion studies and neuroimaging. *Aphasiology*, *13*(8), 581–608.

Brent, M. R. (1999). Speech segmentation and word discovery: a computational perspective. *Trends in Cognitive Sciences*, *3*(8), 294–301.

Brent, M. R., & Cartwright, T. A. (1996). Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, *61*, 93–125.

Cutler, A., & Norris, D. (1988). The role of strong syllables in segmentation for lexical access. *Journal of Experimental Psychology*, *14*(1), 113–121.

de Marcken, C. G. (1996). *Unsupervised language acquisition.* Unpublished doctoral dissertation, MIT.

Ellis, D. (2005). *PLP and RASTA (and MFCC, and inversion) in Matlab using melfcc.m and invmelfcc.m.* http://www.ee.columbia.edu/ dpwe/resources/matlab/rastamat/.

Fishbach, A., Nelken, I., & Yeshurun, Y. (2001). Auditory edge detection: a neural model for physiological and psychoacoustical responses to amplitude transients. *Journal of Neurophysiology*, *85*, 2303–2323.

Goldinger, S. D. (1998). Echoes of echoes? An episodic theory of lexical access. *Psychological Review*, *105*(2), 251–279.

Grünwald, P. (2005). A tutorial introduction to the Minimum Description Length principle. In P. Grünwald, I. J. Myung, & M. Pitt (Eds.), *Advances in minimal description length: Theory and applications.* MIT Press.

Jelinek, F. (1997). *Statistical methods for speech recognition.* MIT Press.

Jurafsky, D., & Martin, J. H. (2000). *Speech and natural language processing.* Upper Saddle River, NJ: Prentice Hall.

Jusczyk, P. W. (1999). How infants begin to extract words from speech. *Trends in Cognitive Sciences*, *3*(9), 323–328.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine learning*, *2*(4), 285–318.

Oates, T. (2002). PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *Proceedings of the international conference on data mining* (pp. 330–337). IEEE.

Rissanen, J. (1972). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Roy, D. K., & Pentland, A. P. (2002). Learning words from sights and sounds: a computational model. *Cognitive Science*, *26*, 113–146.

Saffran, J. R., Aslin, R. N., & Newport, E. L. (1996). Statistical learning by 8-month-old infants. *Science*, *274*, 1926–1929.

Schroeder, M. R. (1999). *Computer speech.* Springer.

Stevens, K. N. (1989). On the quantal nature of speech. *Journal of Phonetics*, *17*, 3–45.

Walker, W., Lamere, P., Kwok, P., Raj, B., Singh, R., Gouvea, E., et al. (2004, Nov). *Sphinx-4: A flexible open source framework for speech recognition* (Tech. Rep. No. TR-2004-139). Sun Microsystems.

Wilpon, J. G., Rabiner, L. R., Lee, C.-H., & Goldman, E. R. (1990). Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *38*(11), 1870–1878.

Yang, C. (2004). Universal Grammar, statistics or both? *Trends in Cognitive Sciences*, *8*(10), 451–456.