RIPTIDE: Segmenting Data Using Multiple Resolutions

Tom Armstrong and Tim Oates
University of Maryland Baltimore County
1000 Hilltop Circle
Baltimore, MD 21250

Abstract— **Segmenting real-valued data, be it speech wave-forms into words and phrases or temperature readings into environmental epochs, is a challenging, open problem. We introduce an unsupervised, domain-independent algorithm, RIPTIDE, that discovers segments in real-valued time series data while constructing a hierarchy of segments. Our top-down approach begins with a coarse approximation of the input data, finds segment boundaries, and recursively considers discovered segments with a finer resolution. We demonstrate the drawbacks of an existing segmentation algorithm and the multiresolution capabilities of a discretization method for time series.**

Index Terms— **Segmentation, Unsupervised Learning, Perceptual Organization, Word Discovery**

I. INTRODUCTION

Segmenting data is a hard problem. In the presence of an ocean of noisy data, humans and animals are facile at processing inputs from sensory receptors to discover meaningful change points [1], [2], [3], [4], [5], yet, at the beginning of life, these abilities are rudimentary [6], [7]. Learning over time and diversity of experience are necessary to identify correct segment boundaries. For example, children require over one year of stimuli to gain adult-level performance in parsing speech across phonological phrase boundaries [8].

How to define a *segment* and what constitutes a *good* segmentation (collectively, the points between segments) remain open problems. Perceptual organization theories vary in a spectrum from structuralism to the Gestaltists. In vision research, the structuralist state of the art uses graph cuts as segment boundaries and maximizes the homogeneity of each segment globally for a good segmentation [9]. This structuralist view fails on other domains when homogeneity is harder to define than simple uniformity of color and intensity. On the other end of the spectrum, Gestaltists minimize the description length of the data and use the description to define a good segmentation, but there are no working unsupervised, domain-independent Gestalt methods.

We present a novel, domain-independent, unsupervised algorithm, RIPTIDE, for segment discovery in time series. We investigate the utility of a compression algorithm for low-level segment discovery and propose a multiresolution extension to an approximation method for coarse to fine refinement of discovered segments.

The remainder of the paper is organized as follows. First, we introduce related work on discovery of segments and segmentations in categorical and real-valued data. Second, we motivate our work through the demonstration of the strengths and weaknesses of two approaches: the SEQUITUR [10] algorithm for compressing streaming categorical data and the SAX [11] representation, a Piecewise Aggregate Approximation (PAA), for time series. Next, we detail our unsupervised algorithm, RIPTIDE, before reporting on experimental results and future directions in the research.

II. BACKGROUND

Bellman introduced the first dynamic programming solution for finding the optimal k-segmentation of a real-valued time series [12]. The solution, and more efficient approximations [13], fits a linear model to each segment where the number of segments is known. In real-world domains like speech data, knowing the number of words or sentences is an unreasonable assumption. However, even with that knowledge, the quantity of segments makes this approaches inefficient.

Moreover, methods for segmenting speech data are successful if they are afforded too much language information and gloss over how they arrive at their representation of basic elements like phonemes [14]. Recent work by Park and Glass use a segmental version of dynamic time warping (DTW) to discover words, found as subsequences of utterance, in speech data only and requires instances of the same word in multiple utterances [15]. These approaches are restricted in their application to natural language.

Categorical data offer more leverage in real-world domains with small alphabets (e.g., bioinformatics, orthographic representations). Wolff advocates compression as a way to discover meaningful segments by reifying frequently occurring bigrams with a unique symbol [16]. More recently, Nevill-Manning uses more judicious merges to compress a single string [10]. The resulting hierarchical representations consist of macro segments, but do not generalize.

A. Segmentation Approaches

Child directed speech is frequently used to both train and test algorithms to maintain a developmentally plausible approach to learning. The most frequently used corpus of these data is the Child Language Data Exchange System (CHILDES) [17]. Using these data, Hammerton applies self-organising maps (SOM), otherwise known as Kohonen Maps, to segment phonetically transcribed speech (speech

defined over a discrete, finite alphabet) [14]. Brent presents a language independent, unsupervised, incremental algorithm named Model-Based Dynamic Programming (MBDP-1), and compares it to approaches that use mutual information and transition probabilities between n-grams [18].

Batchelder's BOOTLEX algorithm incrementally builds a lexicon and segments novel utterances in an unsupervised fashion [19]. The input data must be categorical and the lexicon is initialized with the alphabet of the language. The subsequent inputs are segmented using the lexicon and the parse that maximizes the likelihood of the data is considered. From the parse, lexicon entry frequencies are incremented and novel words are added to the lexicon. Unlike similar approach in the past [20], BOOTLEX parses input strings with the added constraint of the optimal length of segments. However, the solution does not construct a hierarchy, only a lexicon. That is, an additional parameter is given as input limiting the size of lexical entries. The results of this simple algorithm are comparable, on the same data, to MBDP-1.

Other approaches consider higher level linguistic information. Magerman et al. and Brill et al. consider the problem of segmentation while building a hierarchy using part-of-speech (POS) tagged corpora or only the POS tags [21], [22]. Both approaches look at natural language POS tagged corpora and segment using the generalized mutual information (GMI) criterion and divergence, respectively. The former finds *distituents* (i.e. constituent boundaries) at data n-gram positions by computing GMI at each candidate location. Multiple iterations of the algorithm result in an *n*-ary branching hierarchy. A constituent is denoted by two boundary distituents and no *a priori* distituents between the two. The latter operationalizes the concept of free variation by counting POS tags and recording the words in the surrounding contexts. If a tag and a pair of tags can occur in the same context, then they construct a rule for these tags. From the counts of words and contexts, they generate rules and compute a divergence value for each rule. After searching over rule space, the algorithm returns the set of rules that minimize overall divergence. Both approaches produce hierarchies, but only work on a small number of categories, with known boundaries, and do not generalize.

B. Human and Animal Cognition

The ease by which children learn to discover boundaries in their environments and sensor data belies the complexity and computational challenges of the task. To understand the importance of learning to segment and building hierarchies, we present two examples of human capabilities. Early on, children acquire linguistic proficiency in segmenting and parsing. For example, children learn that a prosodic hierarchy governs the structure of speech utterances. The highest level in the hierarchy covers the entire utterance and, in English, starts with high intonation and decreases over the course

of the utterance. Inside the utterance, the phonological level governs the structure of a phrase. To assess the development of phonological knowledge in children, Christophe devised a set of sentences that included the two exemplars below [8]:

(1) "[The college] [with the biggest **paper** forms] [is best]"
(2) "[The butler] [with the highest **pay**] [**per**forms the most]"

Children at the age of 13 months were trained to identify a particular word (in the above case, *paper*). Christophe examined which case the children preferred: the word presented as phonological phrase internal (sentence 1) or the word presented across a phonological phrase boundary (sentence 2). Children at 13 months old preferred the phonological phrase internal presentation of the word. In other words, children as early as 13 months old have acquired sufficient linguistic knowledge about phonological phrase boundaries that they are able to segment fluent speech, and thus disallow the incorrect segmentation in sentence 2.

Recently, Patel et al. [23] demonstrated the differences exhibited by native speakers of English and Japanese and gave intuition for the exhibited differences. While native speakers of the two test languages do not segment audio based upon amplitude (e.g., high, low, high, low, . . .), the segmentation results do differ if the duration of the audio segments is changing (e.g., short, long, short, long, . . .). Their investigation found a statistically significant difference in short to long and long to short syllables in Japanese and English given short and long syllable bigram counts.

III. ALGORITHM

RIPTIDE is an algorithm for segmenting real-valued time series. It performs the segmentation iteratively beginning with a coarse view of the data, finding segments using a compression technique, and then recursively considers each discovered segment as a new time series until a minimum segment length is encountered.

A. Compression

Our approach adopts the beneficial components of a compression algorithm for discovering segment boundaries, but avoids the pitfalls of the approach. Consider one popular algorithm for compressing a categorical data string: SEQUITUR [10]. It compresses a streaming input string by replacing pairs of equivalent digrams with a single element (digram uniqueness), a rule, and requiring that each replacement rule be used more than once (rule utility). Compressing the data using these two constraints finds frequently occurring subsequences and hierarchical structures of subsequences. SEQUITUR, however, does not generalize rules and therefore can only produce the input string from the induced rules. The approach's primary objective is compression, therefore a segment with one occurrence or few occurrences is of little interest as it cannot reduce the number of bits in the

representation through a new rule or replacement with an old rule.

```
((Most Labour)
 (sentiment
     ((would still)
          ((favour the)
               abolition))))
 ((of
          (the House))
     (of Lords)))
```

Fig. 1.   "Most Labour sentiment would still favour the abolition of the House of Lords" tree structure from [10].

In spite of the simplicity of the algorithm and the requirements of the data, SEQUITUR can perform quite well at discovering meaningful segments in an input string. For example, on a paragraph of text consisting of sentences like "the cat hates the dog," SEQUITUR generates mid-tree level rules that derive the words in the sentences. The performance degrades with the removal of spacing and punctuation, but the algorithm still generates reasonable rules for lexical items. Unfortunately, the success at finding higher-level concepts stops at the word level. Nevill-Manning uses the example sentence "Most Labour sentiment would still favour the abolition of the House of Lords" and the resulting tree structure is shown in figure 1.

Compared with a more accepted segmentation of the example sentence (see figure 2), SEQUITUR fails to discover higher level phrase structure in the same input. SEQUITUR posits groupings of '*favour the*' and '*would still*' on the lowest level and the problem compounds itself on the subsequent higher levels (e.g., prepositional phrase attachment).

```
(S (NP Most Labour sentiment)
   (VP would
       (VP (ADVP still)
           favour
           (NP (NP (NP the abolition)
                   (PP of
                       (NP the House)))
               (PP of
                   (NP Lords)))))))
```

Fig. 2.   Example sentence from [10] "Most Labour sentiment would still favour the abolition of the House of Lords" parsed using the CMU Link parser.

We have seen that SEQUITUR performs well at the task of finding low-level segments (e.g., words in text), but cannot be relied upon for higher-level segments (e.g., phases and sentences). The degradation in performance may have to do with the changes in alphabet size that can occur when words are reified. Initially, SEQUITUR operates with a fixed alphabet (e.g., 26 letters plus punctuation and spaces), but when words are used as tokens, the size of the alphabet

increases dramatically and the distribution shifts to one more Zipfian.

B. Multiple Resolutions

There are a myriad of representation choices for time series data. Our interest is in a representation that can begin with a coarse view of the data and as subsequences of the data are identified as segments, the representation becomes finer in detail. Coarse approximations of real-valued data can smooth local variation by using a global view of homogeneity. Finer approximations are useful at more local levels to exploit short term variation. Coarse-to-fine approaches have been used in image analysis like face classification and identification [24]. Lin's symbolic aggregate approximation (SAX) takes a real-valued time series and represents it with a fixed, finite-sized alphabet that is generally small (e.g., less than 20). Lin et al. empirically found that while globally time series may have any distribution and are varied, over short windows of the time series, the distributions are Gaussian [11]. After normalizing the data, SAX splits the range into n sections of uniform probability and assigns an alphabet element to each section (see figure 3).
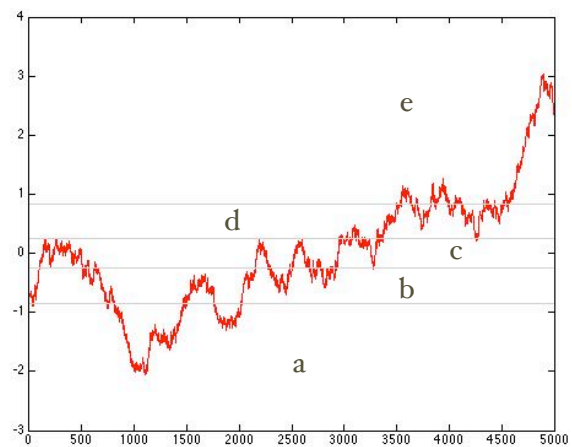

Fig. 3.   A time series and the SAX sections with alphabet elements where the size of the alphabet is 5.

SAX with a small alphabet is analogous to viewing the data through a coarse lens. Slight variations in the data are smoothed in the beginning to find the most globally homogenous segments. As the alphabet size increases, SAX provides a finer approximation to the real-valued data.

C. RIPTIDE

RIPTIDE takes as its input a real-valued time series, the minimum length of a segment for the domain, and an initial alphabet size. The psuedocode for RIPTIDE (see figure 5) details the precise steps of the algorithm. Here we outline the higher-level operations and the intuition for them.
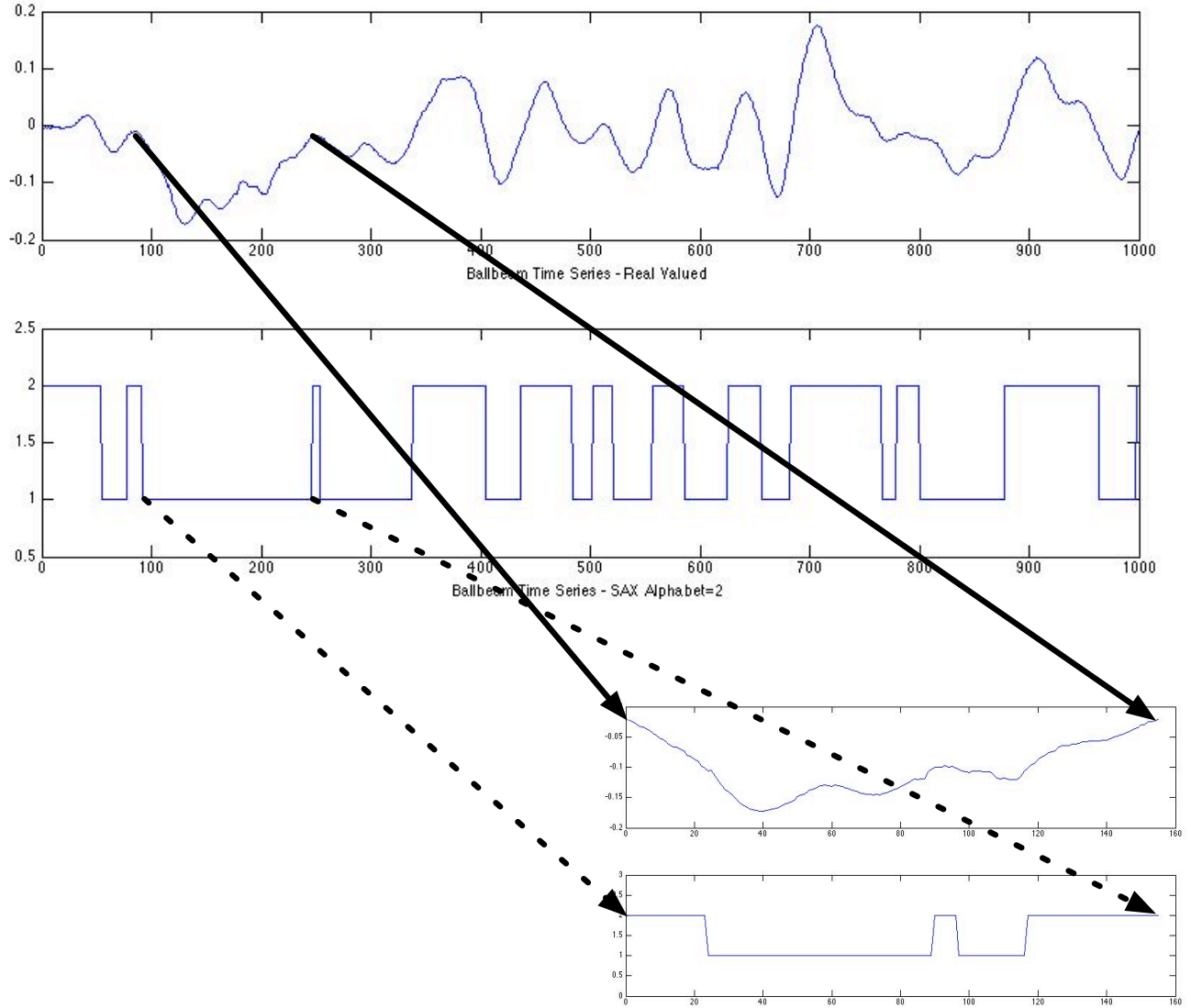
Fig. 4. The data used is the Ballbeam time series [25]. The top larger displays contain the first 1000 values from the original data set - first from the real-valued time series and second in the SAX representation with an alphabet size of 2. The bottom smaller displays contain one segment discovered in the first iteration of RIPTIDE first as a real-valued time series (solid arrow) and then as a further refined SAX representation with an alphabet size of 2 (dotted arrow).

RIPTIDE begins by converting the time series into the SAX representation with a small alphabet, Σ. As we described above, there are benefits to looking at image or audio data first in a coarse way. Second, the reduced alphabet SAX representation is used as input to the standard SEQUITUR algorithm where there are benefits from large inputs that have small alphabets. The output from SEQUITUR is a compressed version of the time series in the form of rules. In other words, SEQUITUR returns a hierarchy for the time series in the form of the SAX alphabet. Instead of using the output from SEQUITUR, the time series could be segmented by

grouping together consecutive subsequences of homogenous values, but empirical results suggest that this approach fails to capture higher-level hierarchical relationships correctly much like SEQUITUR. A minimum depth threshold sets the derivation depth from the start rule and only the lower-level rules are used in creating segments effectively (expandRule in the pseudocode). For example, if SEQUITUR returned a set of rules which derived the string "the cat sleeps," RIPTIDE only considers the rules that derive lower level items like *cat* and *sleeps*, but not *cat sleeps* (as is the standard error case). For each segment that SEQUITUR discovers on a low

```
RIPTIDE(timeSeries, min, Σ)
 1   if |timeSeries| > min
 2      then
 3              timeSeries ← SAX(timeSeries, Σ)
 4              segments ← FINDSEGMENTS(timeSeries, min)
 5              for segment ∈ segments
 6                  do
 7                      RIPTIDE(segment, min, Σ)
 8      else
 9          do
10   RETURN(timeSeries)


FINDSEGMENTS(timeSeries, ruleDepthLimit)
 1   segments ← {}
 2   rules ← SEQUITUR(timeSeries)
 3   for rule ∈ rules
 4       do
 5           if RULEDEPTH(rule) > ruleDepthLimit
 6               then
 7                   segments ← EXPANDRULE(rule)
 8   RETURN(segments)
```

Fig. 5.   RIPTIDE Pseudocode

Fig. 6.   Segmentation of ballbeam time series continued for the segment highlighted in figure 4

level, RIPTIDE recurses on it as input. At each level in the recursion, RIPTIDE uses a new SAX representation as a finer resolution on a constrained portion of the original time series. Figure 4 shows the first two iterations of RIPTIDE on a UCR Time Series Data Mining Archive data set called "ballbeam" [25]. When the segment reaches a minimum length, the recursion ceases.

IV. DISCUSSION AND CONCLUSION

Evaluating the output of RIPTIDE is a challenging task given the nature of the input time series. Methods for categorical data that operate on natural-language text often can begin with the gold standard (or some approximation like the CMU Link parser output).

Figure 6 contains a plot of a subsequence found in the first iteration of RIPTIDE. The superimposed hierarchy shows, in the bold solid line, segments less than 25 units long discovered in the second iteration. The dotted hierarchy shows the next level and smaller segments discovered before RIPTIDE finished processing this subsequence. These data contain valleys and troughs that RIPTIDE is able to identify – first as descents and ascents then as series of valleys.

In this paper, we presented an unsupervised, domain-independent algorithm that segments real-valued time series into a hierarchy of homogenous segments. The concept of homogeneity here is defined by a modification to a respected compression algorithm. We saw how the state-of-the-art
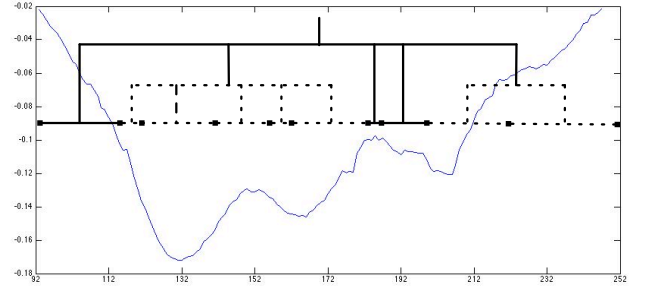
approaches attempt to segment categorical data and fail to address real-valued inputs. Children and animals perform these tasks with ease, but computational approaches are limited. Future work will proceed in a number of directions. First, available speech data provide untapped resources for applications of grammatical inference and hierarchies imposed on strings provide input to many state-of-the-art algorithms. Second, we will use robot sensor data as another rich source that can be evaluated without an expert (e.g., mapping walls, hallways, etc.). Finally, we will explore using other metric-based approaches (e.g., Kullback-Leibler divergence, Gish likelihood ratio) to evaluate the quality of segmenting at points of high entropy change.

REFERENCES

[1] P. Jouventin, T. Aubin, and T. Lengagne, "Finding a parent in a king penguin colony: the acoustic system of individual recognition," *Animal Behaviour*, vol. 57, pp. 1175–1183, 1999.

[2] M. D. Beecher, P. K. Stoddard, and P. Loesche, "Recognition of parents' voices by young cliff swallows," *The Auk*, vol. 102, pp. 600–605, 1985.

[3] J. Fischer, "Emergence of individual recognition in young macaques," *Animal Behavior*, vol. 67, pp. 655–661, 2004.

[4] F. Ramus, M. D. Hauser, C. Miller, D. Morris, and J. Mehler, "Language discrimination by human newborns and by cotton-top tamarin monkeys," *Science*, vol. 288, 2000.

[5] J. M. Toro, J. B. Trobalon, and N. Sebastian-Galles, "Effects of backward speech and speaker variability in language discrimination by rats," *Journal of Experimental Psychology: Animal Behavior Processes*, vol. 31, no. 1, pp. 95–100, 2005.

[6] P. W. Jusczyk, "Investigations of the word segmentation abilities of infants," in *Proceedings of the Fourth International Conference on Spoken Language Processing ICSLP*, vol. 3, Philadelphia, PA, 1996, pp. 1561–1564.

[7] M. J. Spence, P. R. Rollins, and S. Jerger, "Children's recognition of cartoon voices," *Journal of Speech, Language, and Hearing Research*, vol. 45, pp. 214–222, 2002.

[8] A. Christophe, A. Gout, S. Peperkamp, and J. Morgan, "Discovering words in the continuous speech stream: the role of prosody," *Journal of Phonetics*, vol. 31, pp. 585–598, 2003.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000.

[10] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences: A linear-time algorithm," *Journal of Artificial*

Intelligence Research, vol. 7, p. 67, 1997. [Online]. Available: http://www.citebase.org/cgi-bin/citations?id=oai:arXiv.org:cs/9709102

[11] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*.   New York, NY, USA: ACM Press, 2003, pp. 2–11.

[12] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, no. 6, p. 284, 1961.

[13] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. Toivonen, "Time series segmentation for context recognition in mobile devices," in *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*.   Washington, DC, USA: IEEE Computer Society, 2001, pp. 203–210.

[14] J. Hammerton, "Learning to segment speech with self-organising maps," *Language and Computers*, vol. Computational Linguistics in the Netherlands, no. 14, pp. 51–64, 2002.

[15] A. Park and J. R. Glass, "Unsupervised word acquisition from speech using pattern discovery," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006, pp. 409–412.

[16] J. G. Wolff, "An algorithm for the segmentation of an artificial language analogue," *British Journal of Psychology*, vol. 66, pp. 79–90, 1975.

[17] B. MacWhinney, *The CHILDES project: Tools for analyzing talk*, 3rd ed.   Mahwah, NJ: Lawrence Erlbaum Associates, 2000.

[18] M. Brent, "An efficient, probabilistically sound algorithm for segmentation and word discovery," *Machine Learning*, vol. 34, pp. 71–105, 1999.

[19] E. O. Batchelder, "Bootstrapping the lexicon: A computational model of infant speech segmentation," *Cognition*, vol. 83, no. 2, pp. 167–202, 2002.

[20] D. C. Olivier, "Stochastic grammars and language acquisition mechanisms," PhD Dissertation, Harvard University, 1968.

[21] D. M. Magerman and M. P. Marcus, "Parsing a natural language using mutual information statistics." in *AAAI*, 1990, pp. 984–989.

[22] E. Brill and M. Marcus, "Automatically acquiring phrase structure using distributional analysis," in *HLT '91: Proceedings of the workshop on Speech and Natural Language*.   Morristown, NJ, USA: Association for Computational Linguistics, 1992, pp. 155–159.

[23] A. D. Patel, J. R. Iversen, and K. Ohgushi, "Nonlinguistic rhythm perception depends on culture and reflects the rhythms of speech: Evidence from english and japanese," *Journal of the Acoustical Society of America*, vol. 120, p. 3167, 2006.

[24] H. Sahbi and N. Boujemaa, "Coarse-to-fine support vector classifiers for face detection," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*.   Washington, DC, USA: IEEE Computer Society, 2002, p. 30359.

[25] E. Keogh, "The UCR time series data mining archive," http://www.cs.ucr.edu/ eamonn/TSDMA/index.html, Riverside CA. University of California - Computer Science & Engineering Department, 2006.