Artificial General Segmentation

Daniel Hewlett and **Paul Cohen**
Department of Computer Science
University of Arizona
Tucson, AZ 85721, USA

Abstract

We argue that the ability to find meaningful chunks in sequential input is a core cognitive ability for artificial general intelligence, and that the Voting Experts algorithm, which searches for an information theoretic signature of chunks, provides a general implementation of this ability. In support of this claim, we demonstrate that VE successfully finds chunks in a wide variety of domains, solving such diverse tasks as word segmentation and morphology in multiple languages, visually recognizing letters in text, finding episodes in sequences of robot actions, and finding boundaries in the instruction of an AI student. We also discuss further desirable attributes of a general chunking algorithm, and show that VE possesses them.

Introduction

To succeed, artificial general intelligence requires domain-independent models and algorithms that describe and implement the fundamental components of cognition. Chunking is one of the most general and least understood phenomena in human cognition. George Miller described chunking as "a process of organizing or grouping the input into familiar units or chunks." Other than being "what short term memory can hold 7 +/- 2 of," chunks appear to be incommensurate in most other respects. Miller himself was perplexed because the information content of chunks is so different. A telephone number, which may be two or three chunks long, is very different from a chessboard, which may also contain just a few chunks but is vastly more complex. Chunks contain other chunks, further obscuring their information content. The psychological literature describes chunking in many experimental situations (mostly having to do with long-term memory) but it says nothing about the intrinsic, mathematical properties of chunks. The cognitive science literature discusses algorithms for forming chunks, each of which provides a kind of explanation of why some chunks rather than others are formed, but there are no explanations of what these algorithms, and thus the chunks they find, have in common.

The Signature of Chunks

Miller was close to the mark when he compared bits with chunks. Chunks may be identified by an information theoretic signature. Although chunks may contain vastly different amounts of Shannon information, they have one thing in common: Entropy within a chunk is relatively low, entropy at chunk boundaries is relatively high. Two kinds of evidence argue that this signature of chunks is general for the task of chunking sequences and series (see (KB01) for a similar idea applied to two-dimensional images). First, the Voting Experts (VE) chunking algorithm and its several variants, all of which detect this signature of chunks, perform very well in many domains. Second, when sequences are chunked all possible ways and ranked by a "chunkiness score" that combines within- and between-chunk entropy, the highest-ranked chunks are almost always real chunks according to a gold standard. Here, we focus primarily on the former kind of evidence, but also provide some early evidence of the latter kind.

Voting Experts

What properties should a general-purpose chunking algorithm have? It must not simply exploit prior knowledge of a particular domain, but rather must be able to learn to chunk novel input. It must operate without supervision in novel domains, and automatically set any parameters it has to appropriate values. For both humans and artificial agents, working memory is finite, and decisions must be made online, so the algorithm must be efficient and rely on local information rather than global optimization. Finally, learning should be rapid, meaning that the algorithm should have relatively modest data requirements.

VE has these properties. Its name refers to the "experts" that vote on possible boundary locations. The original version of VE had two experts: One votes to place boundaries after sequences that have low internal entropy, given by $H_I(seq) = -log(p(seq))$, the other places votes after sequences that have high boundary entropy, given by $H_B(seq) = -\sum_{c \in S} p(c|seq)log(p(c|seq))$, where S is the set of successors to seq. All sequences are evaluated locally, within a sliding window, so the algorithm is very efficient.

The statistics required to calculate H_I and H_B are stored efficiently using an n-gram trie, which is constructed in a single pass over the corpus. The trie depth is 1 greater than the size of the sliding window. Importantly, all statistics in the trie are normalized so as to be expressed in standard deviation units. This allows statistics from sequences of different lengths to be compared to one another.

The sliding window is passed over the corpus and each expert votes once per window for the boundary location that best matches its criteria. VE creates an array of vote counts, each element of which represents a location and the number of times an expert voted to segment at that location. The result of voting on the string `thisisacat` could be represented as `t0h0i1s3i1s4a4c1a0t`, where the numbers between letters are the total votes cast to split at the corresponding locations.

With vote totals in place, VE segments at locations that meet two requirements: First, the number of votes must be locally maximal (this is called the *zero crossing rule*). Second, the number of votes must exceed a *threshold*. Thus, VE has three parameters: the window size, the vote threshold, and whether to enforce the zero crossing rule. For further details of the VE algorithm see Cohen et al. (CAH07), and also Miller and Stoytchev (MS08). A fully-unsupervised version to the algorithm, which sets its own parameters, is described briefly later in the paper.

Extensions to Voting Experts

Some of the best unsupervised sequence-segmentation results in the literature come from the family of algorithms derived from VE. At an abstract level, each member of the family introduces an additional expert that refines or generalizes the boundary information produced by the two original VE experts to improve segmentation quality. Extensions to VE include Markov Experts (CM05), Hierarchical Voting Experts - 3 Experts (HVE-3E) (MS08), and Bootstrap Voting Experts (BVE) (HC09).

The first extension to VE introduced a "Markov Expert," which treats the segmentation produced by the original experts as a data corpus and analyzes suffix/prefix distributions within it. Boundary insertion is then modeled as a Markov process based on these gathered statistics. HVE-3E is simpler: The third expert votes whenever it recognizes an entire chunk found by VE on the first iteration.

The new expert in BVE is called the *knowledge expert*. The knowledge expert has access to a trie (called the *knowledge trie*) that contains boundaries previously found by the algorithm, and votes to place boundaries at points in the sequence that are likely to be boundaries given this information. In an unsupervised setting, BVE generates its own supervision by applying the highest possible confidence threshold to the output of VE, thus choosing a small, high-precision set of boundaries. After this first segmentation, BVE repeatedly re-segments the corpus, each time constructing the knowledge trie from the output of the previous iteration, and relaxing the confidence threshold. In this way, BVE starts from a small, high-precision set of boundaries and grows it into a larger set with higher recall.

Related Algorithms

While Cohen and Adams (CA01) were the first to formulate the information-theoretic signature of chunks that drives VE, similar ideas abound. In particular, simpler versions of the chunk signature have existed within the morphology domain for some time.

Tanaka-Ishii and Jin (TIJ06) developed an algorithm called Phoneme to Morpheme (PtM) to implement ideas originally developed by Harris (Har55) in 1955. Harris noticed that if one proceeds incrementally through a sequence of phonemes and asks speakers of the language to list all the letters that could appear next in the sequence (today called the *successor count*), the points where the number *increases* often correspond to morpheme boundaries. Tanaka-Ishii and Jin correctly recognized that this idea was an early version of boundary entropy, one of the experts in VE. They designed their PtM algorithm based on boundary entropy in both directions (not merely the forward direction, as in VE), and PtM was able to achieve scores similar to those of VE on word segmentation in phonetically-encoded English and Chinese. PtM can be viewed as detecting an information-theoretic signature similar to that of VE, but relying only on boundary entropy and detecting change-points in the absolute boundary entropy, rather than local maxima in the standardized entropy.

Also within the morphology domain, Johnson and Martin's HubMorph algorithm (JM03) constructs a trie from a set of words, and then converts it into a DFA by the process of minimization. Within this DFA, HubMorph searches for *stretched hubs*, which are sequences of states in the DFA that have a low branching factor internally, and high branching factor at the edges (shown in Figure 1). This is a nearly identical chunk signature to that of VE, only with successor/predecessor count approximating boundary entropy. The generality of this idea was not lost on Johnson and Martin, either: Speaking with respect to the morphology problem, Johnson and Martin close by saying "We believe that hub-automata will be the basis of a general solution for Indo-European languages as well as for Inuktitut."

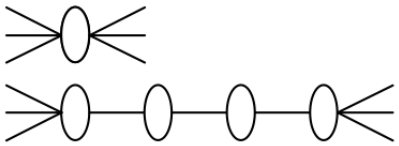

Figure 1: The DFA signature of a *hub* (top) and *stretched hub* in the HubMorph algorithm. Figure from Johnson and Martin.

VE Domains

To demonstrate the domain-independent chunking ability of VE, we now survey a variety of domains to which VE has been successfully. Some of these results appear in the literature, others are new and help to explain previous results. Unless otherwise noted, segmentation quality is measured by the boundary F-measure: $\mathrm{F} = (2 \times \mathrm{Precision} \times \mathrm{Recall})/(\mathrm{Precision}+\mathrm{Recall})$, where precision is the percentage of the induced boundaries that are correct, and recall is the percentage of the correct boundaries that were induced.

Language

VE and its variants have been tested most extensively in linguistic domains. Language arguably contains many levels of chunks, with the most natural being the word. The word segmentation task also benefits from being easily explained, well-studied, and having a large amount of gold-standard data available. Indeed, any text can be turned into a corpus for evaluating word segmentation algorithms simply by removing the word boundaries.

Word Segmentation Results for one corpus, in particular, have been reported in nearly every VE-related paper, and so is the most general comparison that can be drawn. This corpus is the first 50,000 characters of George Orwell's *1984*. Table 1 shows the aggregated results for VE and its derivatives, as well as PtM.

Algorithm	Precision	Recall	F-score
VE	0.817	0.731	0.772
BVE	0.840	0.828	0.834
HVE-3E	0.800	0.769	0.784
Markov Exp.	0.809	0.787	0.798
PtM	0.766	0.748	0.757
All Points	0.185	1.000	0.313

Table 1: Results for VE and VE variants for word segmentation on an English text, *1984*.

Similar results can be obtained for different underlying languages, as well as different writing systems. Hewlett and Cohen showed similar scores for VE in Latin (F=0.772) and German (F=0.794) texts, and also presented VE results for word segmentation in orthographic Chinese ("Chinese characters"). VE achieved an F-score of 0.865 on a 100,000 word section of the Chinese Gigaword Corpus.

The higher score for Chinese than for the other languages has a simple explanation: Chinese characters correspond roughly to syllable-sized units, while the letters in the Latin alphabet correspond to individual phonemes. By grouping letters/phonemes into small chunks, the number of correct boundary locations remains constant, but the number of potential boundary locations is reduced. The means that even a baseline like All Locations, which places a boundary at every possible location, will perform better when segmenting a sequence of syllables than a sequence of letters.

VE has also been tested on phonetically-encoded English, in two areas: First, transcripts of of child-directed speech from the CHILDES database (MS85). Second, on a phonemic encoding of *1984* produced with the CMU pronouncing dictionary. On the CHILDES data, VE was able to find word boundaries as well or better (F=0.860) than several other algorithms, even though the other algorithms require their inputs to be sequences of utterances from which information about utterance beginnings and endings can be gathered (HC09). VE achieved an F-score of 0.807 on the phonemically-encoded version of *1984* (MS08).

Morphology While the word segmentation ability of VE has been studied extensively, its ability to find morphs has not been examined previously. Morph segmentation is a harder task to evaluate than word segmentation, because intra-word morph boundaries are typically not indicated when writing or speaking. We constructed a gold standard corpus of Latin text segmented into morphs with the morphological analyzer *WORDS*.

Algorithm	Precision	Recall	F-score
PtM	0.630	0.733	0.678
VE	0.645	0.673	0.659
BidiVE	0.678	0.763	0.718
All Points	0.288	1.000	0.447

Table 2: Morph-finding results by algorithm. All Points is a baseline that places a boundary at every possible location.

From the table above (Table 2), it is clear that VE in its standard form has some difficulty finding the correct morphs. Still, its performance is comparable to PtM on this task, as expected due to the similarity in the two algorithms. PtM's advantage probably is due to its bidirectionality: VE only actually examines the boundary entropy at the right (forward) boundary. VE was modified with the addition of an expert that places its votes *before* sequences that have high boundary entropy in the *backward* direction. This bidirectional version of VE, referred to as BidiVE, is a more faithful implementation of the idea that chunks are sequences with low internal entropy and high boundary entropy. BidiVE performed better than VE at finding morphs in Latin, as shown in the table.

For reference, when the task is to find word boundaries, the F-score for VE is approximately 0.77 on this same corpus. The reason for this is somewhat subtle: Because VE only looks at entropy in the *forward* direction, it will only consider the entropy after a morph, not before it. Consider a word like `senat.us`: The entropy of the next character following `senat` is actually fairly low, despite the fact that it is a complete morph. This is because the set of unique endings that can appear with a given stem like `senat` is actually fairly small, usually less than ten. Furthermore, in any particular text a word will only appear in certain syntactic relationships, meaning the set of endings it actually takes will be smaller still. However, the entropy of the character *preceding* `us` is very high, because `us` appears with a large number of stems. This fact goes unnoticed by VE.

Child Language Learning VE has also provided evidence relevant to an important debate within the child language learning literature: How do children learn to segment the speech stream into words? Famously, Saffran et al. (SAN96) showed that 8-month-old infants were able to distinguish correctly and incorrectly segmented words, even when those words were nonsense words heard only as part of a continuous speech stream. This result challenges models of word segmentation, such as Brent's MBDP-1 (Bre99), which cannot operate without some boundary information. Saffran et al. proposed that children might segment continuous sequences at points of low transitional probability (TP), the simplest method which would successfully segment their

data.

However, TP alone performs very poorly on natural language, a fact which has not escaped opponents of the view that word segmentation is driven by distributional properties rather than innate knowledge about language. Linguistic nativists such as Gambell and Yang (GY05), argue that this failure of TP to scale up to natural language suggests that the statistical segmentation ability that children possess is limited and likely orthogonal to a more powerful segmentation ability driven by innate linguistic knowledge. Gambell and Yang demonstrate that an algorithm based on linguistic constraints (specifically, constraints on the pattern of syllable stress in a word) significantly outperforms TP when segmenting a corpus of phonetically-encoded child-directed speech. In fact, VE can further outperform Gambell and Yang's method (F=0.953 vs. F=0.946) even though VE has no prior knowledge of linguistic constraints, suggesting that adding innate knowledge may not be as useful as simply increasing the power of the chunking method.

Algorithms like VE and PtM provide a counter-argument to the nativist position, by fully explaining the results that Saffran et al. observed, and also performing very well at segmenting natural language. When represented symbolically as a sequence of phonemes, VE perfectly segments the simple artificial language generated by Saffran et al. (SAN96), while also performing well in the segmentation of child-directed speech. Miller et al. (MWS09) reinforce this case by replicating the experimental setup of Saffran et al., but feeding the speech input to VE instead of a child. The audio signal had to be discretized before VE could segment it, but VE was able to achieve an accuracy of 0.824.

Vision

Miller and Stoytchev (MS08) applied VE in a hierarchical fashion to perform a visual task similar to optical character recognition (OCR). The input was an image containing words written in a particular font. VE was to first segment this image into short sequences corresponding to letters, and then chunk the short sequences into longer sequences corresponding to words. The image was represented as a sequence of columns of pixels, where each pixel was either black or white. Each of these pixel columns can be represented by a symbol denoting the particular pattern of black and white pixels within it, thus creating a sequence of symbols to serve as input to VE. Depending on the font used, VE scored between F=0.751 and F=0.972 on segmenting this first sequence.

After finding letters, VE had to chunk these letters together into words, which is essentially the same as the well-studied word segmentation problem except with some noise added to the identification of each character. VE was still able to perform the task, with scores ranging from F=0.551 to F=0.754 for the three fonts. With perfect letter identification, VE scored F=0.776.

Robot Behaviors

Cohen et al. (CAH07) tested VE on data generated by a mobile robot, a Pioneer 2 equipped with sonar and a pan-tilt-zoom camera running a subsumption architecture. The robot wandered around a large playpen for 20-30 minutes looking for interesting objects, which it would orbit for a few minutes before moving on. At one level of abstraction, the robot engaged in four types of behaviors: wandering, avoiding, orbiting and approaching. Each behavior was implemented by sequences of actions initiated by controllers such as move-forward and center-camera-on-object. The challenge for Voting Experts was to find the boundaries of the four behaviors given only information about which controllers were on or off.

This experiment told us that the encoding of a sequence matters: When the coding produced shorter behaviors (average length of 7.95 time steps), VE's performance was comparable to that in earlier experiments (F=0.778), but when the coding produced longer behaviors, performance is very much worse (F=0.183). This is because very long episodes are unique, so most locations in very long episodes have zero boundary entropy and frequency equal to one. And when the window size is very much smaller than the episode length, then there will be a strong bias to cut the sequence inappropriately.

Instruction of an AI Student

The goal of the DARPA's Bootstrapped Learning (BL) project is to develop an "electronic student" that can be instructed by human teachers, in a natural manner, to perform complex tasks. Currently, interaction with the electronic student is not very different from high-level programming. Our goal is to replace many of the formal cues or "signposts" that enable the electronic student to follow the teacher, making the interaction between them more natural. VE can largely replace one of these cues: the need to inform the student whenever the teacher's instruction method changes.

In BL, teachers communicate with the student in a language called *Interlingua language* (IL). Some IL messages serve only to notify the student that a "Lesson Epoch" (LE) has ended.

Several curricula have been developed for BL. VE finds LE boundaries with high accuracy in all of them – and can be trained on one and tested on another to good effect. To illustrate, we will present results for the Unmanned Aerial Vehicle (UAV) domain. To study the detection of LE boundaries, a training corpus was generated from version 2.4.01 of the UAV curriculum by removing all of the messages that indicate boundaries between LEs. This training corpus contains a total of 742 LEs. A separate corpus consisting of 194 LEs served as a test corpus. As the teacher should never have to provide LE boundaries, the problem is treated as unsupervised and both the training and test corpora are stripped of all boundary information.

Each individual message in the corpus is a recursive structure of IL objects that together express a variety of relations about the concepts being taught and the state of teaching. LEs are defined more by the structure of the message sequence than the full content of each message. Thus, we represent each message as a single symbol, formed by concatenating the IL type of the two highest composite IL objects (generally equivalent to the message's type and subtype). The sequence of structured messages is thus translated into

	TRAINING			TEST		
Size	**P**	**R**	**F**	**P**	**R**	**F**
1.00	0.927	0.888	0.907	0.933	0.876	0.904
0.75	0.881	0.839	0.859	0.904	0.829	0.864
0.50	0.905	0.784	0.840	0.871	0.772	0.819
0.25	0.961	0.772	0.856	0.836	0.606	0.703

Table 3: BVE Results on UAV Domain trained on different subsets of the training corpus. "Size" is percentage of the training corpus given to BVE.

a sequence of symbols, and it is this symbol sequence that will be segmented into LEs.

BVE is allowed to process the training corpus repeatedly to gather statistics and segment it, but the segmentation of the test corpus must be done in one pass, to model more closely the constraints of a real teacher-student interaction. If allowed to operate on the full UAV corpus, BVE finds LE boundaries handily, achieving an F-score of 0.907. However, this domain is non-trivial: VE achieves an F-score of 0.753, only slightly lower than its score for word segmentation in English text. As a baseline comparison, segmenting the corpus at every location results in an F-score of 0.315, which indicates that LE boundaries are roughly as frequent as word boundaries in English, and thus that high performance is not guaranteed simply by the frequency of boundaries of the data.

Results from segmenting a test corpus (not drawn from the training corpus) consisting of 194 lesson epochs are shown in Table 3. "Training Size" refers to the percentage of the training corpus processed by BVE before segmenting the test corpus. From these results, it is evident that BVE can perform very well on a new corpus when the training corpus is sufficiently large. However, with a small training corpus BVE does not encounter certain boundary situations, and thus fails to recognize them during the test, resulting in lower recall.

Evidence for Generality

So far, we have discussed in detail one kind of evidence for the general applicability of VE, namely that VE successfully performs unsupervised segmentation in a wide variety of domains. In order for VE to be successful in a given domain, chunks must exist in that domain that adhere to the VE's signature of chunks, and VE must correctly identify these chunks. Thus, the success of VE in each of these domains is evidence for the presence of chunks that adhere to the signature in each domain. Also, VE's chunk signature is similar to (or a direct generalization of) several other independently-developed signatures, such as PtM, HubMorph, and the work of Kadir and Brady (KB01). The independent formulation of similar signatures by researchers working in different domains suggests that a common principle is at work across those domains.

Optimality of the VE Chunk Signature

Though the success of VE in a given domain provides indirect evidence that the chunk signature successfully identifies chunks in that domain, we can evaluate the validity of the chunk signature much more directly. To evaluate the ability of the chunk signature to select the true segmentation from among all possible segmentations of a given sequence, we developed a "chunkiness" score that can be assigned to each possible segmentation, thus ranking all possible segmentations by the quality of the chunks they contain. The chunkiness score rewards frequent sequences that have high entropy at both boundaries (Equation 1), just as in VE. The score for a complete segmentation is simply the average of the chunkiness of each segment. If the chunk signature is correct, the true segmentation should have a very high score, and so will appear close to the top of this ranking. Unfortunately, due to the exponential increase in the number of segmentations (a sequence of length n has 2^{n-1} segmentations), this methodology can only be reasonably applied to short sequences. However, it can be applied to many such short sequences to better gain a better estimate of the degree to which optimizing chunkiness optimizes segmentation quality.

$$Ch(s) = \frac{H_f(s) + H_b(s)}{2} - \log Pr(s) \qquad (1)$$

For each 5-word sequence (usually between 18 and 27 characters long) in the Bloom73 corpus from CHILDES, we generated all possible segmentations and ranked them all by chunkiness. On average, the true segmentation was in the 98.7th percentile. All probabilities needed for computing the chunkiness score were estimated from a training corpus, the Brown73 corpus (also from CHILDES). Preliminarily, it appears that syntax is the primary reason that the true segmentation is not higher in the ranking: When the word-order in the training corpus is scrambled, the true segmentation is in the 99.6th percentile. Still, based on these early results we can say that, in at least one domain, optimizing chunkiness very nearly optimizes segmentation quality.

Automatic Setting of Parameters

VE has tunable parameters, and Hewlett and Cohen (HC09) showed that these parameters can greatly affect performance. However, they also demonstrated how these parameters can be tuned without supervision. Minimum Description Length (MDL) provides an unsupervised way to set these parameters indirectly by selecting among the segmentations each combination of parameters generates. The *Description Length* for a given hypothesis and data set refers to the number of bits needed to represent both the hypothesis and the data given that hypothesis. The *Minimum* Description Length, then, simply refers to the principle of selecting the hypothesis that minimizes description length. In this context, the data is a corpus (sequence of symbols), and the hypotheses are proposed segmentations of that corpus, each corresponding to a different combination of parameter settings. Thus, we choose the vector of parameter settings that generates the hypothesized segmentation which has the minimum description length.

Extension to Non-Symbolic Data

Strictly speaking, VE can only operate over sequences of discrete symbols. However, as already demonstrated by Miller et al.'s applications of VE to the visual and auditory domains, many sequences of multivariate or continuous-valued data can be transformed into a symbolic representation for VE. Also, the SAX algorithm (LKWL07) provides a general way to convert a stream of continuous data into a sequence of symbols.

Allowing Supervision

While the ability of VE to operate in a fully unsupervised setting is certainly a strength, the fact that VE contains no natural mechanism for incorporating supervision may be seen as a limitation: If some likely examples of ground truth boundaries are available, the algorithm ought to be able to take advantage of this information. While VE itself cannot benefit from true boundary knowledge, one of its extensions, BVE, does so handily. BVE's knowledge trie can store previously discovered boundaries (whether provided to or inferred by the algorithm), and the knowledge expert votes for boundary locations that match this prior knowledge. The Markov Experts version is able to benefit from supervision in a similar way, and, if entire correct chunks are known, HVE-3E can as well.

An Emergent Lexicon

VE does not represent explicitly a "lexicon" of chunks that it has discovered. VE produces chunks when applied to a sequence, but its internal data structures do not represent the chunks it has discovered explicitly. By contrast, BVE stores boundary information in the knowledge trie and refines it over time. Simply by storing the beginnings and endings of segments, the knowledge trie comes to store sequences like `#cat#`, where # represents a word boundary. The set of such bounded sequences constitutes a simple, but accurate, emergent lexicon. After segmenting a corpus of child-directed speech, the ten most frequent words of this lexicon are *you, the, that, what, is, it, this, what's, to*, and *look*. Of the 100 most frequent words, 93 are correct. The 7 errors include splitting off morphemes such as *ing*, and merging frequently co-occurring word pairs such as *do you*.

Conclusion

Chunking is one of the domain-independent cognitive abilities that is required for general intelligence, and VE provides a powerful and general implementation of this ability. We have demonstrated that VE and related algorithms perform well at finding chunks in a wide variety of domains, and provided preliminary evidence that chunks found by maximizing chunkiness are almost always real chunks. This suggests that the information theoretic chunk signature that drives VE is not specific to any one domain or small set of domains. We have discussed how extensions to VE enable it to operate over nearly any sequential domain, incorporate supervision when present, and tune its own parameters to fit the domain.

References

[Bre99] Michael R Brent. An Efficient, Probabilistically Sound Algorithm for Segmentation and Word Discovery. *Machine Learning*, pages 71–105, 1999.

[CA01] P Cohen and N Adams. An algorithm for segmenting categorical time series into meaningful episodes. *Lecture notes in computer science*, 2001.

[CAH07] Paul Cohen, Niall Adams, and Brent Heeringa. Voting Experts: An Unsupervised Algorithm for Segmenting Sequences. *Intelligent Data Analysis*, 11:607–625, 2007.

[CM05] Jimming Cheng and Michael Mitzenmacher. The Markov Expert for Finding Episodes in Time Series. In *Proceedings of the Data Compression Conference (DCC 2005)*, pages 454–454. IEEE, 2005.

[GY05] Timothy Gambell and Charles Yang. Word Segmentation: Quick but not Dirty. 2005.

[Har55] Zellig S. Harris. From Phoneme to Morpheme. *Language*, 31:190, 1955.

[HC09] Daniel Hewlett and Paul Cohen. Bootstrap Voting Experts. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009.

[JM03] Howard Johnson and Joel Martin. Unsupervised learning of morphology for English and Inuktitut. *Proceedings of the 2003 North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL-HLT 03)*, pages 43–45, 2003.

[KB01] Timor Kadir and Michael Brady. Saliency, Scale and Image Description. *International Journal of Computer Vision*, 45:83–105, 2001.

[LKWL07] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15:107–144, April 2007.

[MS85] Brian McWhinney and Cynthia E. Snow. The child language data exchange system (CHILDES). *Journal of Child Language*, 1985.

[MS08] Matthew Miller and Alexander Stoytchev. Hierarchical Voting Experts: An Unsupervised Algorithm for Hierarchical Sequence Segmentation. In *Proceedings of the 7th IEEE International Conference on Development and Learning (ICDL 2008)*, pages 186–191, 2008.

[MWS09] Matthew Miller, Peter Wong, and Alexander Stoytchev. Unsupervised Segmentation of Audio Speech Using the Voting Experts Algorithm. *Proceedings of the 2nd Conference on Artificial General Intelligence (AGI 2009)*, 2009.

[SAN96] Jenny R Saffran, Richard N Aslin, and Elissa L Newport. Statistical Learning by 8-Month-Old Infants. *Science*, 274:926–928, 1996.

[TIJ06] Kumiko Tanaka-Ishii and Zhihui Jin. From Phoneme to Morpheme: Another Verification Using a Corpus. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages (ICCPOL 2006)*, volume 4285, pages 234–244, 2006.