Project no. 034362

ACORNS

Acquisition of COmmunication and RecogNition Skills

Instrument: STREP
Thematic Priority: IST/FET

WP2: PD-module with self-directed search, derived segmental quality measures, full integration of CMM to ACORNS

Due date of deliverable: 30 November 2009
Actual submission date:

Start date of project: 1 December 2006 Duration: 36 months

Project coordinator name: Prof. Lou Boves
Project coordinator organisation name: Radboud University,
Revision [draft, v03.2]

WP2: PD-module with self-directed search, derived segmental quality measures, full integration of CMM to ACORNS

Unto K. Laine, Okko Räsänen, Seppo Fagerlund, Toomas Altosaar, TKK
Guillaume Aimetti, University of Sheffield
Gustav Henter, KTH

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	X

Table of contents:

0. Introduction

WP2 – *Signal Patterning* of the ACORNS project consists of two subtasks. The main objectives of *Task 1* are:

- Develop methods and theory for Pattern Discovery (PD) to learn elementary patterns from signals.
- Develop bottom-up strategies to automatically learn and store more complex pattern structures.
- Study noise tolerant features and processing methods[1].

The main objective of *Task 2* is:

- Study the applicability of Computational Mechanics Models (CMM) and corresponding methods to discover, describe, and quantify structure and patterns[1].

This report consists of three main sections. Section 1 leads off with some very general, introductory notes on two important dimensions of information: *quantity* (value) and *quality* (order, structure). These topics are discussed in order to illuminate the studied methods, their derivations, and how they have been applied to pattern discovery and modeling.

Section 2 deals with practical experimentation in two different fields of pattern discovery activities within ACORNS (*Task 1*): the search for subword units and a *permutation transformation* based time-domain method to discover the burst section of stop consonants. The latter is directed by sections 1's discussion of quantity vs. quality.

Finally, section 3 presents a summary of the studies performed in the field of computational mechanics (CMM, *Task 2*).

[1] Technical Annex of the ACORNS project

1. General notes on information and pattern theory

In a recent paper, *Toward a source coding theory for sets*, the authors Varshney and Goyal from MIT develop coding theory based on *multisets* and their *permutations* (Varshney & Goyal, 2006). Multisets only give quantitative information about the elements in the sequence without their order. Since order doesn't affect the number of elements (quantities) it can be thought of as the qualitative, structural aspect of the information sequence. The authors use the terms *value* and *order* when talking about these two different aspects of information.

Sometimes the order of the elements in a sequence is not very important and the coding can be realized purely based on the probabilities of the occurrence of the elements. In some cases the situation may be just the opposite when order is significant. It is relatively easy to show that when the length of the sequence increases without increasing the size of the alphabet or changing their statistics (a stationary process), information is transmitted more and more through the ordering of the elements and less and less through their statistics, i.e., the size and type of the multiset in question.

The latter view dominates speech signals as well. We may quantize the amplitude values to a few bits and still recognize the message. However, if we start to permute the sequence the message soon becomes lost in noise. The importance of sample order in speech signals is demonstrated by Figure 1.1 where the time waveforms of the eight Finnish vowels (each sequence contains only 1024 samples, about five pitch periods) are first sorted (according to sample value from the smallest to the largest) and then averaged (the middle blue curve). The upper and lower curves depict the standard deviation from the mean value. One notes that the deviation is relatively small, indicating that the sorted signals are almost identical and difficult to classify. The important differences in the temporal structures of the vowels have fully disappeared by the sorting operation.

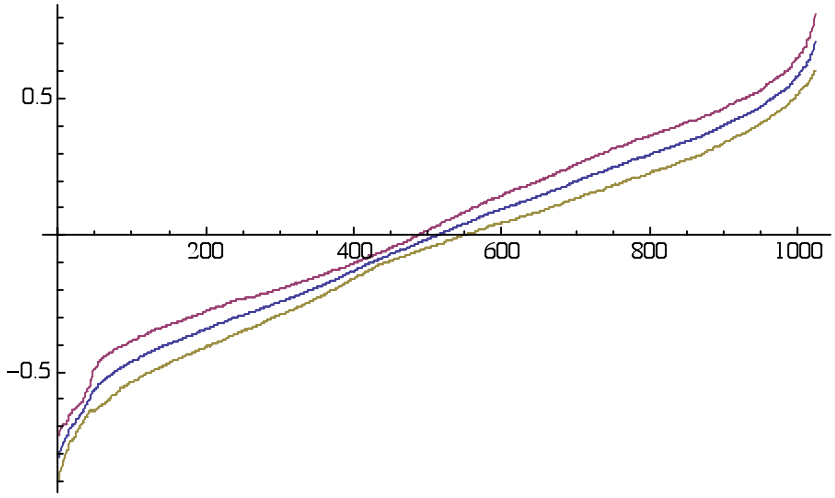

Figure 1.1 Mean amplitudes of the eight, sorted Finnish vowels (blue) with standard deviations (red and brown).

The two aspects of information, the quantity (value) and quality (order, structure) are important when trying to develop new coding and recognition methods. Especially the importance of the structural component has given motivation to study and apply *permutation theory* to pattern discovery. If the structural component is important, then the permutation method should provide an effective tool for pattern analysis and classification. This hypothesis was confirmed by the preliminary study described in section 2.2.

From the philosophical point of view distinguishing information into two aspects is not a fundamentally new idea. This way of thinking has a long history with its roots in *Metaphysics* by Aristotle (Aristoteles, 1990).

Whatever we do, we need methods and tools for logical thinking, including concepts and conceptual analysis, i.e., how to define the basic concepts used in the actual field and what the entities are, or items the concepts are referring to. Metaphysics[2] in philosophy is the forum dealing with these aspects. It has two traditional branches: *cosmology* and *ontology*, both trying to explain the fundamental nature of being and the world.

Especially *ontology* deals with *definition* and *classification* of *entities*, physical or mental, the nature of their properties, and the nature of change. It tries to clarify the notions by which people understand the world, including *existence*, *objecthood, property, space, time, causality,* and *possibility*[3].

This section consists of some introductory notes and discussions on the ontological aspects of information. This provides a nice, unifying view of the topic. The discussion extends the ideas presented in our earlier ACORNS WP2 Year 2 report (Laine et al., 2008; in the following we refer to this report as: *D2.2*) and hopefully illuminates the background of information and pattern theories, especially those related to the theory of permutations.

1.1 Some ontological aspects of information

The basic elements in the Aristotelian ontology are *matter* and *form*. Aristotle gives illuminating examples, some of which are concrete and some more abstract, in nature. Let us first consider a statue made of bronze. Bronze metal itself is not very interesting as it just provides the necessary material dimension and the *substrate* for the artistic work. More informative and intriguing is the actual shape of the statue, what it represents and how talented the artist has been in creating it.

Another, more abstract example is when we compare an alphabet with words made from the alphabet. In order to be able to write something we need a set of symbols called an alphabet. An alphabet is a necessity but not very interesting as such. We may study books written in some language by analyzing the frequency of different characters, however, we may soon note that the statistics are very similar for different books. After these analyses we still cannot say much about the writers or quality of the texts based

[2] The term metaphysics was coined by the students of *Aristotle.* After the death of their famous teacher they wanted to publish a collection of his studies mainly covered after his book *Physics*. So they came up with the new title *Metaphysics* that simply and literally means something coming *after the physics*.

[3] http://en.wikipedia.org/wiki/Metaphysics

only on this aspect. Thus an alphabet can be seen as just the *matter* necessary to write books. Studying their frequency does not reveal their true quality, i.e., the true *form* or *structure* of the text.

The *matter* aspect seems to provide answers to questions like: *Made of what? How much? How large?* It deals mainly with the quantitative aspect whereas the *form* seems to give answers to questions like: *What has been made? What has been created? What is its complexity?* These aspects are more qualitative and holistic in nature.

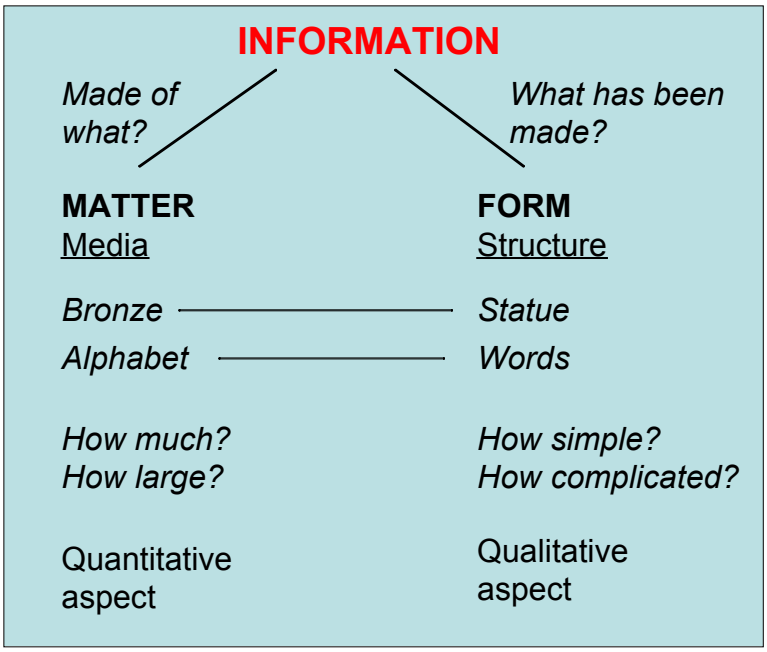

Figure 1.1: Two ontological aspects of information

We could now borrow this general view and try to apply it to analyze the ontology of the concept of information. When replacing the concepts *matter* and *form* with the concepts *media* and *structure,* a new view is created where the media is just the flow of bits (quantity) and the structure is how they are organized (quality)[4]. We need some concrete or abstract media to code and transform information. However, it is just a basic necessity (a kind of substrate), not directly dealing with the true contents or qualities transmitted through the media. For example, from the physical point of view, speech sounds are just temporally and locally limited variations in air density and pressure. They are, in this sense, just variations in *matter* (quantity), whereas their temporal structure allows us to define different *forms* (qualities), e.g., is a bird singing, a car passing, or a child laughing.

The classical Shannonian view on information is concerned mainly with quantitative aspects. The theory gives tools to measure information capacity and to use the media in an efficient way. However, the Shannon entropy (information entropy) is not

[4] The quantitative and qualitative aspects should not be understood as *absolute* categories in this discussion. *Bronze* has also qualitative properties, e.g., in relation to other metals, an alphabet may be expressed different fonts, etc. The qualities of bronze emerge from the lower level of matter hierarchy (molecules and atoms). Thus on every level of a hierarchy it is possible to discuss matter (quantity aspect) and emerged quality (or qualities) based on the form or structure of the matter.

sensitive to the variation in the structure of the message so far as the statistics (frequency) of the elements is not affected. For example, the entropy of the sequence {1,2,3,4,5,6} is equal to the entropy of {4,1,5,2,6,3}. In other words, permutations do not affect Shannon entropy.

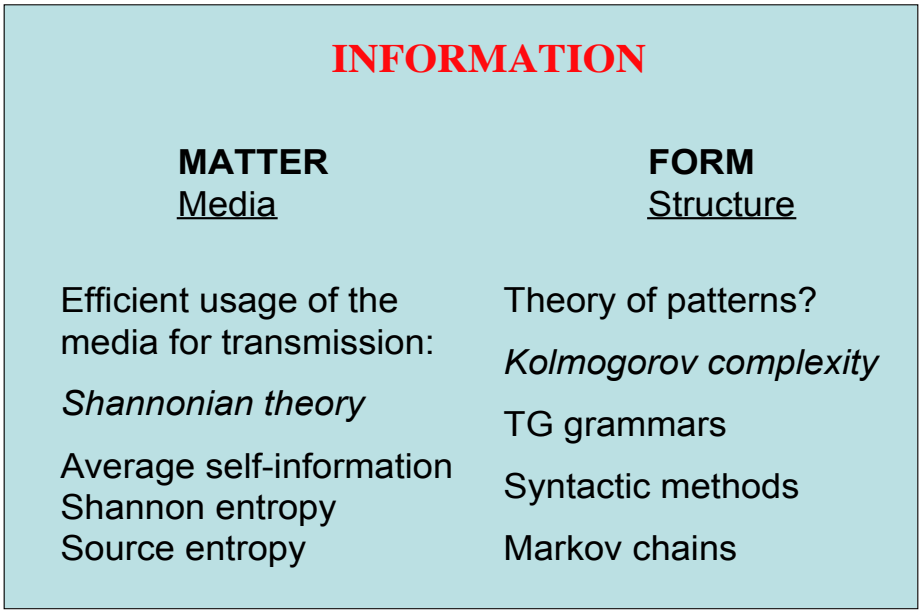

Figure 1.2: Some theories associated with *media* and *structure* of information.

The structural aspect of information approaches an area of research where attempts to formulate a universal theory of patterns has been made (see the related discussion in *D2.2*). *Kolmogorov complexity* deals with structural aspects, too, as seen in Figure 1.2. Examples of other methods dealing mainly with structural aspect of information are syntactic methods, transformation-generative grammars, Minimum Description Length (MDL), and Markov models. Also, the *Concept Matrix* (CM; Räsänen et al., 2009) method developed within ACORNS WP2 can be classified as belonging to this category.

This brief ontological analysis allows us to classify information processing methods into quantity and quality oriented clusters. Through closer examination of these clusters and publications dealing with them it is evident that most of the post-Shannonian theoretical and practical work has occurred in the field of information structures and patterns. This has been one of the main motivations to continue the search in this direction for new tools and methods within ACORNS.

In *D2.2* (Laine et al., 2008) we initiated a general discussion on *patterns* and *pattern theory*, and brought out the issue that no comprehensive theories of patterns and their modeling exist in current pattern recognition research. An interesting reference found is a paper arising from statistics by David J. Hand and Richard J. Bolton (2004). In the abstract of the paper they state the following:

> *For sound reasons, which are outlined in the paper, the data mining community has tended to focus on the algorithmic aspects of pattern discovery, and has not developed any general underlying theoretical base. However, such a base is important for any technology: it helps to steer the direction in which the technology develops, as well as serving to provide a basis from which algorithms can be compared, and to indicate which problems are the important ones waiting to be solved. This paper attempts to provide such a theoretical base, linking the ideas to statistical work in spatial epidemiology, scan statistics, outlier detection, and other areas. One of the striking characteristics of work on pattern discovery is that the ideas have been developed in several theoretical arenas, and also in several application domains, with little apparent awareness of the fundamentally common nature of the problem. Like model building, pattern discovery is fundamentally an inferential activity, and is an area in which statisticians can make very significant contributions.*

This publication happens to follow very closely the thoughts expressed in *D2.2*. The fact that two different teams with very different backgrounds have come up with similar views independently of each other can be seen as a sign for an objective need for a more comprehensive pattern theory and a firmer basis for the processing of patterned information.

2. Activities in pattern modeling and discovery

This section gives an overview of the research activities undertaken in WP2 during the third year of the project. The first sub-section will present findings from attempts to develop methodology for automatic acquisition of subword units using Concept Matrices (CM) and vector quantized speech. Then an interesting novel approach to signal and pattern modeling, namely permutation transformation, will be introduced. The study of the method was planned in the Technical Annex (TA) of ACORNS project, however, due to relatively high risk associated to the method, its closer study was limited and left to the final phase of the project. Finally, methodology for learning of sub-word units using DP-ngrams is presented in section 2.3. Computational Mechanics (CMM; Shalizi & Crutchfield, 2001), yet another pattern discovery method studied in the WP2 of the ACORNS project, has its own chapter and is discussed in the section 3.

2.1 Searching for subword units from VQ data

2.1.1 Introduction

The Concept Matrix -approach (CM) developed in WP2 (Räsänen et al., 2009) has been found to be effective in learning statistical models of words in a weakly supervised manner. However, the word models that CM learns are based on fuzzy and noisy observation statistics that do not explicitly state the temporal structure of the words in a form where different signal characteristics could be assigned to, e.g., beginning, middle, and word ending. Therefore it became of interest whether it would be possible to develop a method that would represent learned words by more explicitly defined spectrotemporal models and whether it would be possible to discover subword structures with the help of these word models. In this report, the subword structures refer to repeatedly occurring units such as sub-phones, phones, or syllables that make up the larger word-like units.

The starting point for building novel word specific models was the detection of temporal locations of keyword realizations in the test utterances. By training and using CM to recognize keywords from speech in the ACORNS corpus, it is possible to locate the temporal segments in each utterance where the keyword models become strongly activated. These activation data enable extraction of those parts of the input VQ-sequences that correspond to known keywords. Once a section of VQ-sequence is extracted, it is stored into a so-called word library that consists of lists of VQ-sequences corresponding to each keyword. In addition to the VQ-representations of the keywords, the spectral properties of the entries in the VQ-codebook are known, enabling reconstruction of word spectrograms from the extracted VQ-data. Figure 2.1 shows an example of Association Response Table (see Räsänen et al., 2009) where three keywords W_1, W_2 and W_3 are detected and identified. Corresponding VQ-sequences underlying these words are extracted and stored.

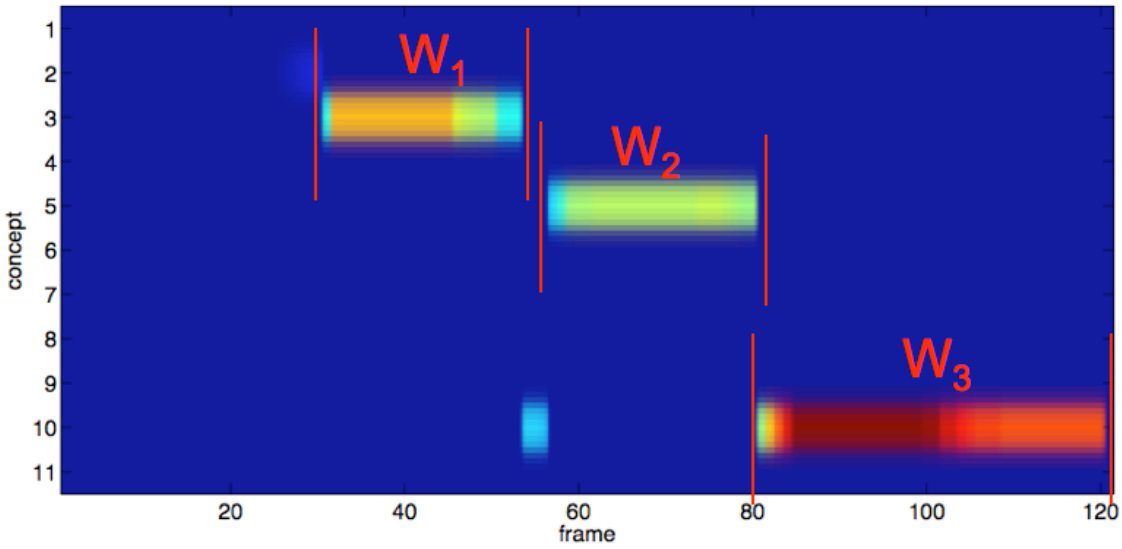

Figure 2.1: Extraction of word realizations based on word model activations in CM. Three words. W_1, W_2, and W_3, are detected in the utterance and the corresponding temporal segments are highlighted with red bars in the Association Response Table (ART).

2.1.2 Alignment of VQ-data

The extraction of the keyword related VQ-sequences is not a perfectly accurate process. Since activation curves of the CM are used as temporal pointers to the word in the VQ-sequence, they suffer from several sources of variation. In addition to normal temporal variation in pronunciation of words at different positions in utterances, one central problem is the fact that the learning system does not know all of the words in the utterances due to differentiation to tagged (learnable) keywords and untagged carrier sentences in the ACORNS corpora. This causes inaccuracies in the detection of the beginning and ending of words since neighboring words are not always modeled in the system. Another source of variation that was present in the experiments is the morphemic structure of the Finnish language, since all inflections of a word are included in the same CM model, and on the other hand, many inflectional suffixes are shared between several words. Finally, it is possible that CM simply fails to detect the keyword in its entire length, but only extracts parts of varying size from the words.

The above sources of variation impose a need of alignment of the VQ-sequences stored in the word library. Several different alignment methods were studied to solve this problem. The use of Dynamic Time Warping (DTW) was studied for alignment of the spectrograms obtained from the VQ-sequence library and the corresponding VQ codebook. The major challenge was how to align a large number of spectrograms into one coherent format. Several different warping procedures were studied, e.g., by warping pairs of sequences to the same length at first, and then warping results from each pair to results of another pair etc., or by warping all realizations to one "proto-template" that was chosen automatically using some specific criterion (e.g., the longest sequence or the sequence closest to the mean keyword length). However, the DTW alone was not found to be sufficient due to very large differences in spectrogram lengths.

A different method to align multiple N-dimensional time-series, or in this case, spectrograms, was also studied. This method does not adjust the length of the sequences,

but rather attempts to find the best possible alignment to the original sequences by adjusting their relative positions in time. The optimization criterion of the algorithm is the mean cross-correlation of frames between the current spectrogram and all other spectrograms, summed over all frames *t* in the spectrogram under analysis. Changes to the alignment of a spectrogram are performed by sliding the spectrogram against all other *K*-1 spectrograms that are fixed in time, and by computing the overall correlation across all other spectrograms for each position. This produces a correlation curve like in fig. 2.2. Then the spectrogram under analysis is moved towards the point of maximal correlation from its previous location by zero-padding it with a suitable number of empty frames. This process is then repeated for the next spectrogram, which is then again adjusted towards the point of maximal correlation. This is repeated iteratively until there are no changes in the overall positioning of the spectrograms or the maximum number of iterations is reached. Iteration by iteration, the correlation peak becomes sharper as the alignment improves. See Appendix A for algorithm description.

Figure 2.3 shows an example alignment obtained for the keyword "*kissa*" (English "*cat"*). It contains 30 realizations of the word, represented as MFCC spectrograms reconstructed from the VQ-data. As can be seen, the lengths of the spectrograms are very different from each other, and some of the realizations are missing large parts of the signal when compared to the longer ones. Even after the alignment, large amounts of variation exist, especially further away from the point of maximal contrast. This lead to the development of a more sophisticated version of the alignment algorithm, where alignment was performed piecewise in a temporally local window and by using information from automatic blind segmentation of speech. The detailed explanation of this algorithm is outside the scope of this report, but in principle the algorithm attempts to align phone-like segments of different realizations of a keyword using a local alignment process similar to the one described above. Once the segments have been matched across all realizations, each segment can be labeled automatically and (temporal) distributions of VQ-labels for each phone-like segment can be computed.

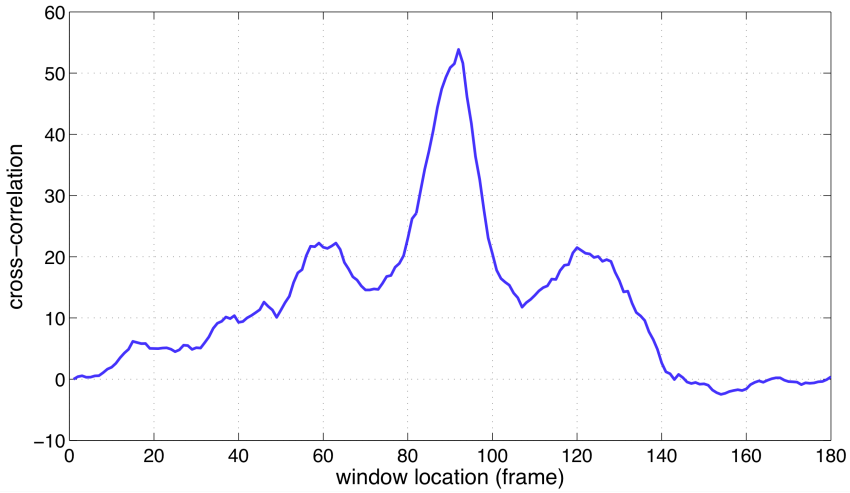

Figure 2.2: Non-normalized averaged cross-correlation function obtained by correlating a sliding spectrogram (current sample) with the other spectrograms in a fixed position (reference).

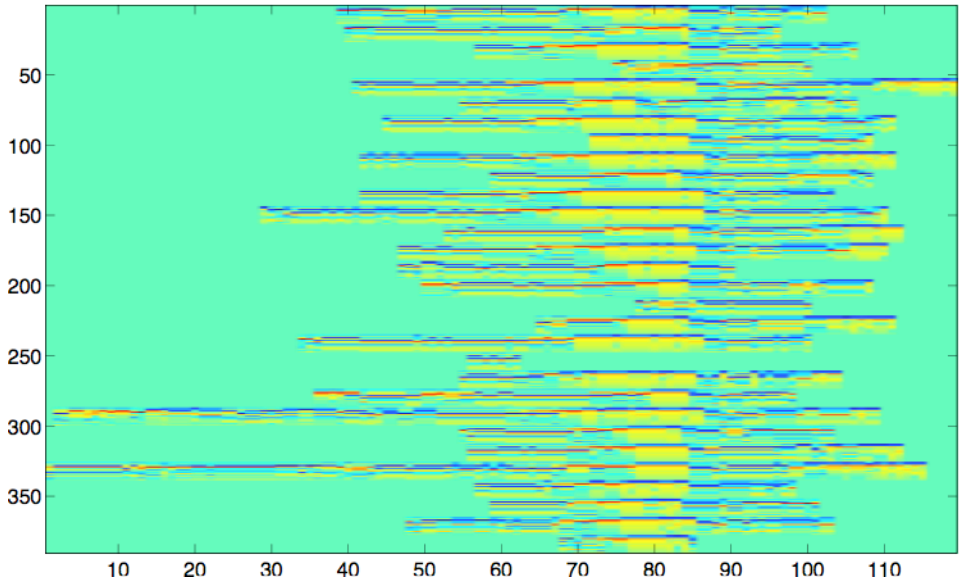

Figure 2.3: Global alignment obtained for 30 realizations of word "kissa" (cat). Columns correspond to signal frames, and rows correspond to elements of the spectrograms. Enormous variation in the lengths of extracted keyword realizations is evident.

2.1.3 Building of spectrotemporal word templates

Once the alignment of VQ-sequences has been performed, a straightforward way to define the spectrotemporal model (STM) of a keyword is to compute a distribution of occurrences of VQ-labels as a function of their location in the keyword. A matrix $\mathbf{M}_k$ of size C x T is created for each keyword k, where T is the number of temporal bins and C is the size of the VQ-codebook. This matrix indicates how many times a given VQ label has occurred in a given temporal position in a keyword. In order to map word related VQ-sequences in the word library to a finite number of temporal bins, each occurrence of a keyword undergoes a linear temporal normalization, where the first VQ-label of a realization is assigned to time bin $t(1)$ and the last label to time $t(T)$. All other labels in the realization are assigned to time bins in the range [1,T] by scaling the position linearly between 1 and T based on the label's respective location in the word.

Now each realization of a keyword is used to update the frequency matrix $\mathbf{M}_k$ by going through the VQ-sequence and adding one to $M_k[c(x),t(x)]$ for each label c(x) assigned to bin t(x). This results in a distribution of VQ-labels as a function of time for the given keyword (fig. 2.4). In addition, the lengths of VQ-sequences are computed and normal distribution parameters of the keyword lengths are stored in addition to the normalized-time spectrotemporal models. The spectrotemporal distributions $\mathbf{M}_k$ in each temporal bin can be normalized in different ways. One possibility is to simply normalize the sum of VQ-labels in each bin to one, yielding a proper probability $P(a_i,t)$ for each VQ-label a_i at different moments of time in a word. Logarithmic compression $\log(P(a_i,t))$, and, e.g., information theoretically motivated $P\log(P)$ compressions were also studied.

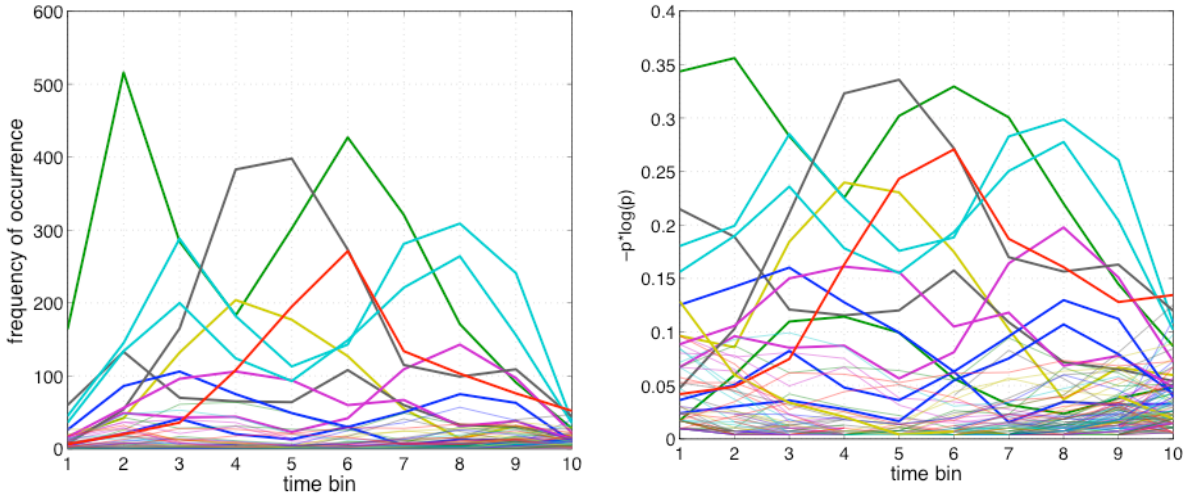

Figure 2.4: An example STMs of a keyword "*hymyilevä*" ("smiling"). X-axis denotes normalized time bins and Y-axis denotes frequency of occurrence (left panel) or $-P\log(P)$ (right panel) of each VQ-label at the given bin. Different VQ-indices are shown in different colors.

According to the Shannonian information theory the amount of information in bits is given by a weighted sum of logarithmic probabilities of the symbols (alphabets) used. Thus the probability $P(a_i)$ of a symbol a_i contributes the information measure by a term: $-P(a_i)\log(P(a_i))$. The n-tuple of these terms forms a local information vector. Thus each time bin of a word (in STM representation) is associated with an information vector. Based on Euclidean distance these vectors were then used to measure "information differences/similarities" between different words. The informal visual comparison of different representations for word similarity matrices supports the usage of the described information measure (see Figures 2.5 and 2.6).

In principle, the obtained STMs can be used for word recognition directly from novel vector quantized speech input. By resampling the models to their proper length using the learned keyword length parameters, it is easy to compute likelihood scores for each model starting from each moment of time t in the input sequence. The best word recognition results obtained with this approach using Y2 Finnish ACORNS data (50 keywords) lead to an approximately 70 % keyword recognition rate, which is far worse than the original CM recognition rate (~95 % correct). The use of several word templates of slightly different lengths did not yield a notable increase in performance, but increased the computational complexity notably. The reason why the recognition based on full word STMs did not work very well was found to be due to the fact that there was still much temporal inaccuracy in the models even after numerous attempts to improve alignment of the original VQ-data. This can be also verified by inspecting the STM distributions (e.g., fig. 2.4) where there are no clear-cut boundaries in the distributions between the syllables of the word.

2.1.4 Subword detection using the STMs

Instead of using the STMs for keyword recognition, it was studied whether the models of different keywords sharing similar linguistic units would contain localized similarities in their spectrotemporal distributions. The aim was to build a mechanism for automatic extraction of subword structures that make up the words. The first step in the analysis

was to compute Euclidean distances between STM distributions of different keywords. Then an algorithm was applied for localization of diagonal regions with small inter-STM distances (i.e., parts of the STMs that share similarities with the other STMs) and collection of respective parts of STMs to a sub-word STM library.

Figures 2.5 and 2.6 display two different distance matrices between STMs for different words. In figure 2.5, the acoustic and linguistic similarities in the beginnings and endings of words /k//o//i//r//a/ and /k//o//t//k//a/ ("*koira*" and "*kotka*") cause the mutual distances of STMs to be small. However, even after numerous different alignment attempts, the representations of the words are still blurry in the temporal domain and no clear diagonal structure can be seen. The same problem is seen between the words /k//i//s://a/ and /i//s//i/ ("kissa" and "isi") (fig. 2.6), where the /i/ and /s/ in both words correlate well, but the region of correlation has spread over a large temporal distance instead (each frame on the x and y axes correspond to 10 ms in time).

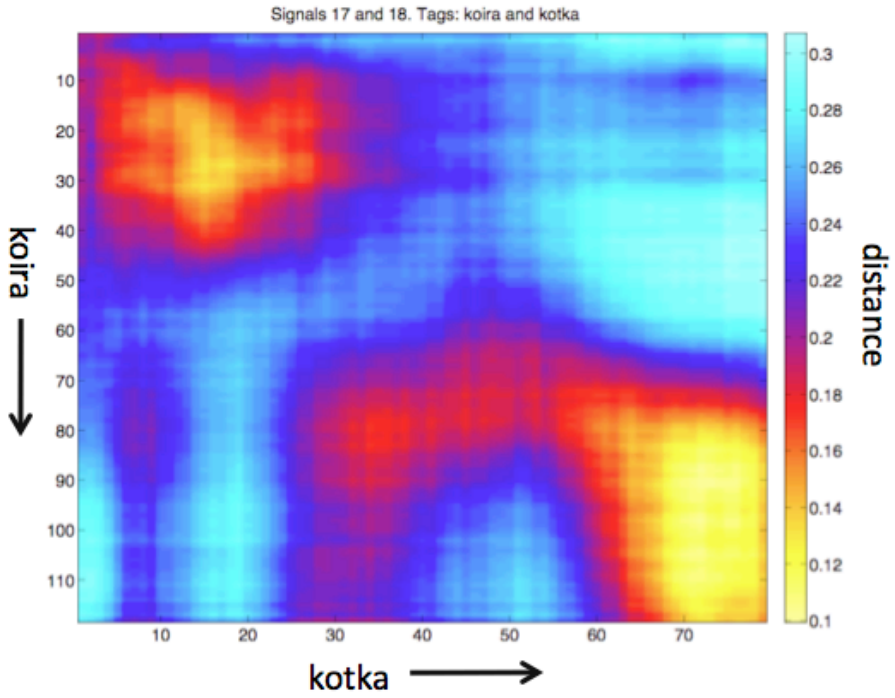

Figure 2.5: A distance matrix between spectrotemporal models (STMs) of keyword "koira" (dog) and "kotka" (eagle) using Euclidean distance as a distance metric of the information vectors $P \log(P)$ of STMs. Both axes are time as 10 ms frame numbers. VQ-alignment was performed using DTW. Initial /ko/ and final /a/ similarity is evident for the two words (bright yellow areas).

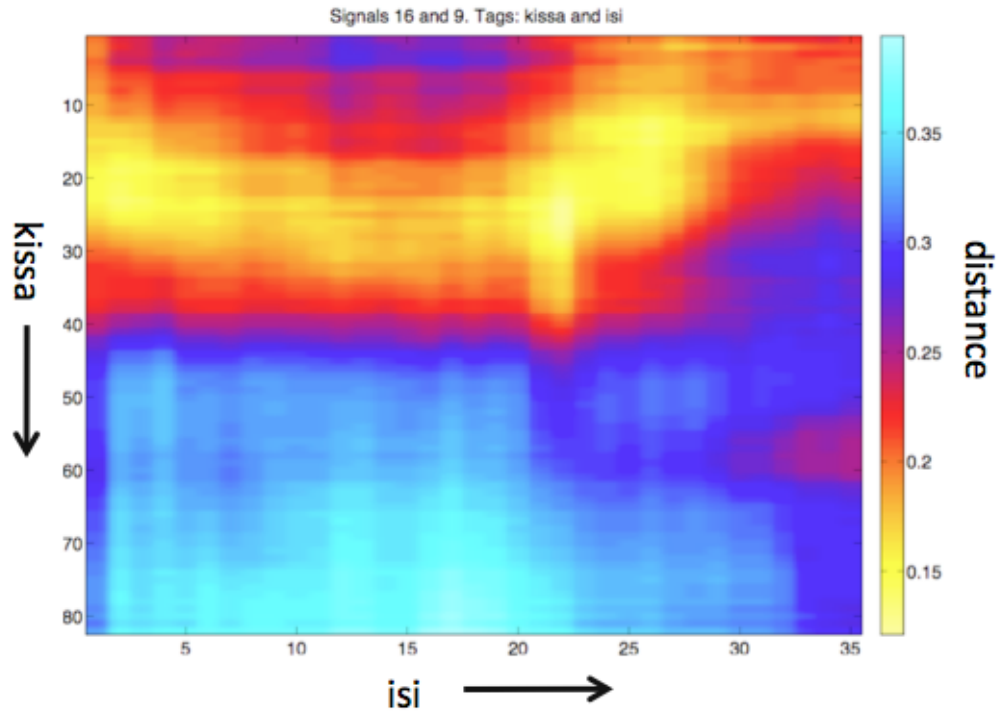

Figure 2.6: A distance matrix between STMs of keyword "kissa" (cat) and "isi" (daddy) using the same metric as in Figure 2.5. Both axes are time as 10 ms frame numbers. VQ-alignment was performed using DTW. A strong correlation exists between the /is/ regions of both words but is temporally blurred.

Despite the insufficiencies in the temporal accuracy of STMs, automatic extraction of well-correlating parts was performed and these sub-word STMs were assigned into a sub-word STM library. Then the subword STMs were used as recognition templates similarly to previously tested full word STMs in order to see how they react to continuous speech input. When automatic recognition was performed using these subword STMs, it become clear that the selectivity of most of the obtained units was not very good. Most STMs represented frication noise and silence. Only some specific vowels, e.g., /i/ (fig. 2.7), or syllables (/ka/) were represented by a small number of very selective STMs that reacted only in a given context. However, a majority of the models were reacting to several phone classes at the same time, which is not a good property if automatic detection and classification of phone-like units or syllables is desired.

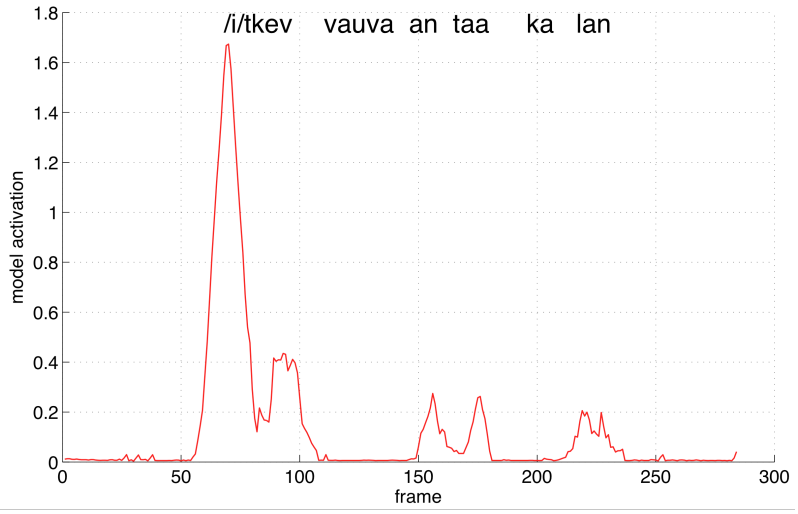

Figure 2.7: Response for a STM model of /i/ for utterance "*Itkevä vauva antaa kalan*". A selective response is obtained for the only /i/ in the utterance.

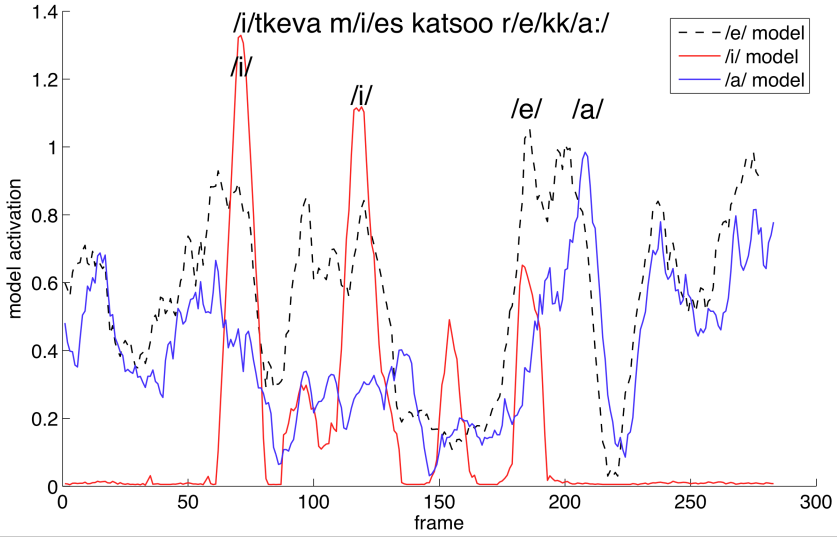

Figure 2.8: Responses for STM models of /i/, /e/ and /a/ for utterance "*Itkevä mies katsoo rekkaa*". The orthography is approximately aligned with the underlying signal. The /e/ and /a/ models react only to specific allophones in a specific context, but not to all corresponding phones in the utterance. Note that in Finnish orthography and pronunciation of words are very similar.

Naturally, the number of learned subword units was also very limited due to the small amount of linguistic overlap in different keywords. This made the attempts to reconstruct entire word models using the subword units impossible. Also, the temporal inaccuracy was exceedingly high for successful learning of CV structures to take place outside the points of maximal contrast[5], since average length of plosives and their related context cues were of the same scale as the variation in the temporal alignment of the VQ-data, not to mention the fact that the plosives might not have been coded very accurately in VQ

[5] The alignment of the VQ-data was most successful near the most contrasting spectral changes in the middle of words, e.g., between the /an/ and /ka/ in word "*ankka*" (eng. *duck*).

in the first place. Only some of the most prominent CV syllables were quite successfully discovered. As an example, figure 2.9 shows 25 automatically extracted signals that react most to a sub-word STM that resembles syllable /k//a/.

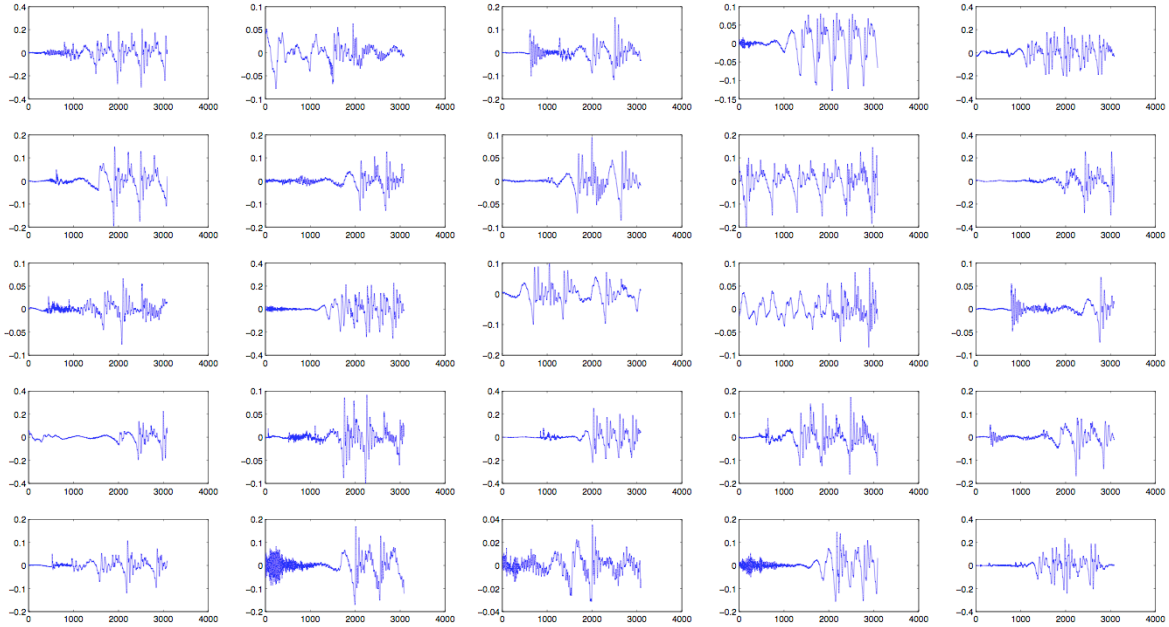

Figure 2.9: Signal waveforms extracted automatically from continuous speech using automatically discovered unit that has most perceptual resemblance with phonetic /k//a/. Some waveforms are very similar, starting from closure and ending to voiced [a]. However, this is not always the case.

2.1.5 Segmental histogram approach

Also studied was the idea to recode segmental units into new phone-like classes. The blind segmentation algorithm developed in WP2 during the first year of the project provided information regarding hypothetical segment boundaries between phone-like units. Since it was now possible to represent each word with a segmental structure by collecting the statistics of segments across all realizations of the word with the help of the alignment process mentioned earlier, it was also possible to compute a distribution of VQ-labels inside each segment. Each segment was further divided into onset and offset sections by splitting it at the middle, and distributions of VQ-labels were computed for both sections for each segment. For the 50 keywords in the ACORNS Y2 material, 185 segments were considered reliable (coherence of segment boundaries across realizations) during the alignment process and thereby modeled in this manner. Mean lengths of the segments were also stored.

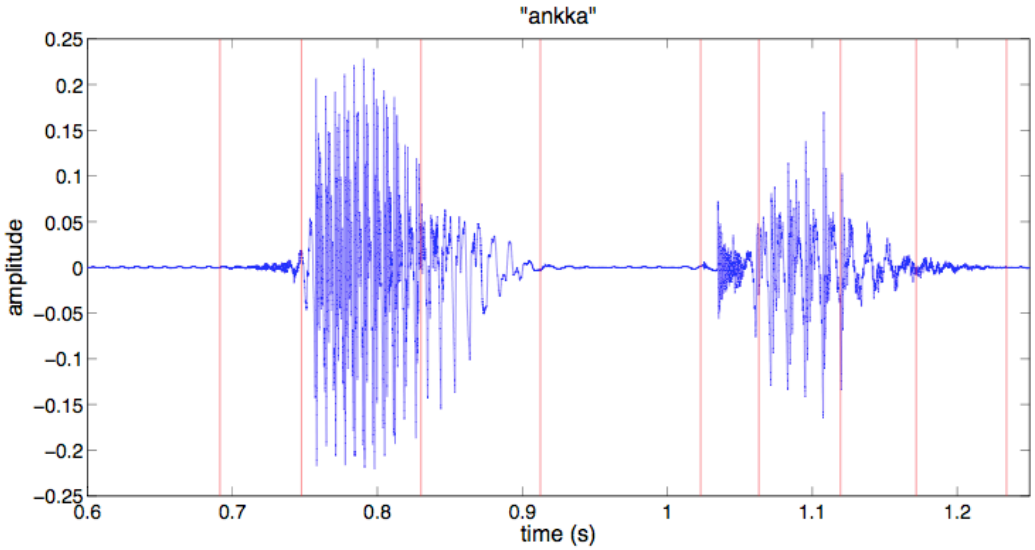

Figure 2.10: An example of automatic segmentation of the word "ankka". Red vertical lines indicate detected segment boundaries.

Activations of the obtained segmental units were studied with speech VQ-data by sliding a window over the VQ-sequence and computing the Euclidean distance between normalized VQ-label distributions in the window and each segment model. The window size can be fixed, or it can be varied depending on the mean segment length corresponding to each segmental model. By replacing the VQ-index at the beginning of each window with a non-terminal symbol pointing to the best matching segmental model, it is possible to recode original VQ-sequences with a segmental VQ code that hopefully generalizes across different pronunciations of the segment. It is also possible to perform clustering for the segmental units in order to reduce the size of the segmental codebook.

After recoding the signals, the CM algorithm can be taught again and recognition can be performed on previously unseen utterances. When a 60 ms fixed size window was used for recoding the signals with 185 segmental models, a keyword recognition rate of 81.80 % was obtained for the four main speakers of the Y2 Finnish ACORNS corpus. The adjustment of the window size did not enhance this notably. The use of k-means clustering to reduce the number of segments from 185 to a smaller number (e.g., 30, 50 or 130 segments; VQ-histograms as features) did always have an adverse effect on recognition. The baseline recognition with the given test set was 92 % correct word recognitions using the original MFCC features extracted every 10 ms and vector quantized with a k-means codebook.

2.1.6 Discussion

PRIMIR theory of infant language acquisition proposes that learning a number of words precedes learning of phonetic sub-word structures of a language (Werker & Curtin, 2005). This theory is in line with the findings of the ACORNS project, where possibilities to learn meaningful linguistic units have been attempted from several perspectives. Purely bottom-up learning of linguistically meaningful phonetic units based on statistical properties of the signals, as suggested by Native Language Magnet Theory expanded (NLM-e; Kuhl et al., 2008), turns out to be extremely difficult if not impossible without

some contextual constraints on the acoustic level due to significant overlap and variability in phonetic units. It has also been shown that with the help of constraints from the lexical layer, it is possible to learn a Bayesian classifier that classifies phones into proper phonemic categories (Feldman et al., in press). In absence of lexical constraints, the learning process leads to improper feature distributions for the categories. Although Feldman and her colleagues did not use real speech in their experiments, but only synthetic vowel formant frequency information from the study of Hillenbrand et al. (1995), their experiment shows that even preliminary knowledge of the possible structures at the lexical level helps to structure lower level information in a new manner.

In our attempts to find new representations for speech, the small-scale events at the acoustic level were constrained by assigning them to lexical items. By comparing different realizations of the same linguistic units, we hoped to find sub-word structures that would both describe the known words in a more precise manner, and also generalize more efficiently towards new words across different pronunciation variations.

The reason why we did not succeed in this task very well is not explicitly clear, but it certainly has to do with the complexity of real speech. First of all, the segmentation and classification of words from continuous speech is error prone, and from the point of view of signal processing, it requires much hard work to build an unsupervised system that can say that two given signals are different realizations of the same linguistic event. Even if the classification, alignment, and warping of the word forms were perfect, the manner in which the signals are represented has a large impact on the way how their assumed sub-word structures are dealt with. In the WP2 studies we limited ourselves to a discrete space of approximately 150 different acoustic classes, defined by vector quantization of spectral features (namely MFCCs) extracted from signal frames every 10 milliseconds. Although this representation has been shown to be efficient for pattern discovery from speech using the CM and Non-negative Matrix Factorization (NMF) frameworks (Van hamme, 2008), it might not be optimal for the modeling of highly detailed sub-word structures. Exact alignment of VQ-sequence fragments turned out to be especially difficult, whereas the dynamic time warping of continuous spectral domain speech spectrograms has been successful in many different applications. In retrospect, it might have been wiser to abandon the original VQ-domain and study the different realizations of the keywords using some other spectrotemporal feature representations, possibly with the support from automatic segmentation of speech.

If we take another perspective to the question why the studied sub-word structures do not enhance keyword recognition from the ACORNS speech, it might be so that methods like CM and NMF are already modeling the data very efficiently given the weakly supervised learning framework. Attempts to move away from holistic statistical representations towards explicitly defined spectrotemporal word models might throw away the strengths of these pattern discovery methods. What might be actually lost is the capability to account for great variability and noise in data by modeling dependencies at long temporal distances, which can be considered as modeling of several parallel, piecewise, and possibly intersecting spectrotemporal trajectories for each keyword. The challenges in modeling of VQ data and the pros of the given pattern discovery methods were already noted in ACORNS deliverable D2.2 (Laine et al., 2008) (p. 45). This also speaks on behalf for the creation of more intelligent signal representations after preliminary lexical learning instead of sticking to the original frame based VQ.

It is also important to note that from a self-learning agent's or infant's point of view, the discovery of underlying sub-word structure is not a self-evident fact that the learner should perform. There is no real reason for the learner to search for new ways to structure the speech input, as long as the current representations of speech and language are sufficiently functional. This is especially true with small vocabularies like the ones studied in ACORNS, where the number of meaningful words is small. It is much more straightforward and efficient to code each word as a whole instead of using sub-word units that might be more numerous than the number of known words themselves. Only if the vocabulary becomes sufficiently large, it becomes ecological to find and represent words by their smaller systematically recurring constituents. This is very similar to the situation in ASR, where each word is trained separately for a small vocabulary recognizer, whereas large vocabulary recognizers have to resort to bi- or triphone modeling. In material like the ACORNS Y2 corpus, the sub-word structures that occur in several keywords in a similar context are very rare, making modeling and testing of context-sensitive sub-word units impractical. The number of words that the artificial learner hears in these experimental settings is much less than the amount of speech human infants are exposed to during first years of their lives. So in the end, we return to the problem of sufficient training material for the bootstrapping of a system.

2.2 Temporal analysis of speech based on the permutation transformation – a study of stop consonants

2.2.1 Introduction

A central goal of WP2 research was to study different discrete representations of speech in order to provide rich representations for higher-level processes. The general concept selected to denote all these different representations was *discrete model elements* (DME). The project started by studying conventional spectrotemporal elements based on the (fast) Fourier transform (FFT) and MFCCs (Laine et al., 2008). The primary domain of these elements is frequency, however, when they are computed in a small sliding window the detailed temporal structure of the signal can be revealed.

However, the number of methods applicable directly in the time-domain is much more limited. Basically, we compute features like: energy envelope, autocorrelation and linear prediction (LP) coefficients (as is known, LP can also be interpreted as a method to create spectral models for signals, even though its primary design occurs purely in the time-domain).

Typically the time domain is considered as a "problematic domain" for signal modeling, because it is quite sensitive to many disturbances like variation in phase (e.g., in the transmission channel), additive noise, nonlinear distortion, and echoes. However, when working with permutations with small time windows spanning only a few samples, we have not noted any such problems. Rather, many of the obtained results are promising.

Coming back to the frequency domain methods, many of them, e.g., the FFT, wavelets, etc., can also be interpreted as descriptions of special *temporal structures* existing within the signal. For example, each Fourier component gives an estimate of the amount of sinusoidal structure of a certain frequency present in the temporal waveform.

Recent literature in mathematics, theoretical physics, and signal theory shows a gradually increasing number of attempts to describe the temporal structures of signals *directly in the time domain* by looking at signal segments composed of only a few samples. These form the basic elements and by coding or quantizing these segments, discrete representations can be formed from them (Bandt, 2005). One promising method is based on permutation theory[6] and the permutation transformation.

The importance of information conveyed by *temporal ordering* can be easily demonstrated by sorting longer segments of speech by amplitudes. After this sorting there is very little difference left between segments from different speech regions. Once sorted, they all just reflect the same general distributions of signal amplitudes while all individual details are lost.

In the following section we shortly describe the background of the method and an attempt to apply a permutation transformation to produce new kinds of time-domain DMEs for speech signals, and further, to produce simple statistical models for speech segments based on these elements. A preliminary test of the method was performed by constructing a classifier for stop consonants. The method is primarily based on the permutation analysis of the waveforms of the burst segments of stop consonants. The results are comparable to those obtained by more conventional frequency-domain methods existing in literature.

2.2.2 Permutation transformation

Permutation theory can be applied in many ways in signal processing (Keller et al., 2007). Our approach is to concentrate on the *ordering* of the individual samples present in a small analysis window of only 2-7 samples. Ordering is based on ranking of sample amplitude values. For example, considering a window (list) of three samples $\{a,b,c\}$ we can transform it to the list $\{1,2,3\}$ if $a \leq b \leq c$ and to the list $\{3,2,1\}$ when ordering is reversed. In this way every possible ordering of the samples in a list of n elements leads to an individual permutation of the list $\{1,2,3, \ldots, n\}$ among $n!$ different permutations.

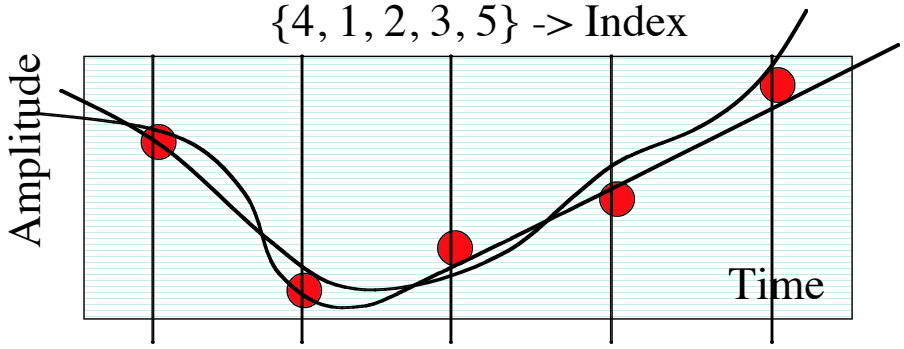

Figure 2.11: Permutation {4,1,2,3,5} obtained from two slightly different waveforms.

Figure 2.11 depicts an example of two similar waveforms with their common permutation transformation: {4, 1, 2, 3, 5} which can be represented by one of the 120 possible indices. Note, that as long as the amplitude ordering of the samples do not change, the transformation's output remains unchanged. Thus each permutation produced

[6] *Cauchy* (1789-1857) initiated the study of *permutation groups* indicating that ideas related to ordering already have a long history.

by the transformation represents an *equivalent class* of signal segments. Another important aspect of the permutation transformation is that it is *scale-free*. The permutation transformation is not affected by scaling of the signal (multiplication by a positive constant or adding a constant). The samples may even be mapped by any monotonic nonlinear function (e.g. *tanh*) without any effect in the permutation obtained. Also, a small amount of additive noise or other variation (e.g., fluctuations in the channel properties) does not affect the permutation code as long as the *ordering* of the samples is not affected.

The permutation transformation can be seen as a method to map the temporal structure of small signal segments in a scale free manner into a set of indices each pointing to one choice among the $n!$ different permutations (Groth, 2005). Also, we can consider the transformation as a method to quantize a small segment of a signal as a single integer where each integer represents a permutation index, a code to represent one of the n! possible permutations. In this manner quantization doesn't occur on a sample-by-sample basis but rather on a small list (set) of samples simultaneously.

Sometimes it is practical and efficient not to select adjacent signal samples to form a permutation index. Instead, we may select every kth sample. In this case we say that the applied *time-delay* is k (Keller et al., 2007). Note that even though this process in linear signal processing is called sub-sampling and typically requires filtering to avoid aliasing, here we are dealing with temporal structures of signals coded by permutations using a non-linear operation which does not necessarily require filtering.

2.2.3 Metric for permutations

In some cases we may need a tool to measure the similarity (and dissimilarity) between two permutations. One possible metric is based on Kendall's *tau* (Kendall, 1938).

$$\tau = 1 - \frac{2\,d_K(a,b)}{d_{K_{Max}}}, \quad -1 \le \tau \le 1 \qquad (2.2.1)$$

where d_K is *Kendall's distance or metric,* $0 \le d_K \le d_{Kmax}$ and $d_{Kmax} = n(n-1)/2$ is the maximum distance among the elements (e.g., $d(\{1,2,3\},\{3,2,1\}) = 3$). Kendall's distance is defined as the minimum number of local, elementary permutations needed to reorganize a permutation b to form an equal list with permutation a. The elementary permutation means an operation where the neighboring elements are interchanged, e.g., {1,2,3}-> {2,1,3} has $d_K = 1$. Thus Kendall's metric is not *geometric* in the nature, but *algorithmic*.

In the following simulation this metric is applied to smooth the statistical image obtained by the permutation transformation. We integrate the number of events inside a certain Kendall distance to form a smoothed (averaged) statistical image of occurrences of certain permutation pairs in the sequence of permutations created from the signal.

2.2.4 A study of stop consonant classification

In this work we have applied the permutation transformation method to stop consonant classification. This topic was chosen since stop consonant classification has been considered as a difficult task using traditional methods.

The permutation transformation method starts by collecting information from the signal's internal structure using very short time windows. We have used a permutation order of 5, and thus there exist 120 different permutations where each permutation window maps onto one of the 120 indices. Furthermore, we have tested *time delay* values of 1 and 2 (distances between selected samples) that correspond to window lengths of 0.63 ms and 1.3 ms, respectively. Before applying the permutation transformation, the signals were filtered with a low-pass or pre-emphasis filter. Used low-pass filters were all order 10 FIR filters.

2.2.5 Transition frequency matrix

Time domain signals were transformed into permutation code sequences using the permutation transformation with a sample step size of one. The permutation code sequence therefore has nearly an equal length to the original signal. The permutation code sequence is further transformed into a transition frequency matrix that describes the statistics of permutation transitions (permutation pairs) existing within the signal. The transition frequency matrix is created using different *time lags*, which describe the time delay between two permutations (*window hop size*). Transition frequency matrices are then used as simple statistical models in the classification tests. Since a permutation order of five was used, the transition frequency matrices have a size of 120x120 elements.

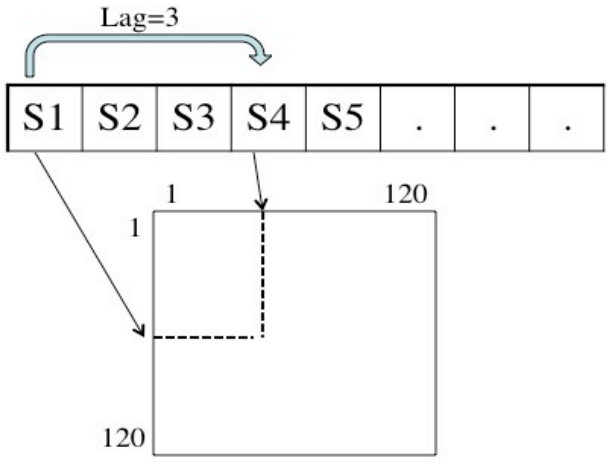

Figure 2.12: Composing a permutation transition frequency matrix from a permutation code sequence. Each element (*Sx*) in permutation code sequence refers to one permutation transform of the signal.

The transition frequency matrices become very sparse since most of the possible permutation transitions are not present in the matrices. This can cause problems during classification. To overcome this we decrease the matrix sparseness using a spatial filter. The spatial filter is not applied directly to the transition frequency matrices but through

Kendal distances (see Equation 2.2.1). Each matrix element is composed of two different permutations and every permutation has four closest neighbors (d_K=1), thus each element pair (corresponding to one point in the frequency matrix) is affected by eight other elements that are at a d_K=1 distance from the original elements of the pair. New values for the matrix elements are obtained by weighting the sum of nine values (neighboring elements have half of the weight related to the original value). In classification experiments we found optimal results when the smoothing was repeated three times. Recall that the closest neighbor of a permutation corresponds to interchanging two adjacent samples in a permutation window. Thus smoothing has the effect of attenuating the variance in the signal amplitudes.

2.2.6 Classification tests

Classification was tested for stop consonants in the context of vowels [AA] and [EH] (Arpabet) separately. Training and testing samples were taken from the TIMIT corpus, and only male speaker samples were used in classification tests. One model was created for each consonant as an average from all consonant training set. Models were created from the consonant release parts only.

In testing, the phase transition frequency matrix was first created with a 4 ms time window and incrementally updated using a 1 ms time shift. The test model was compared to the consonant models using the Euclidean distance measure. The sample being tested was assigned to the model that had the smallest distance.

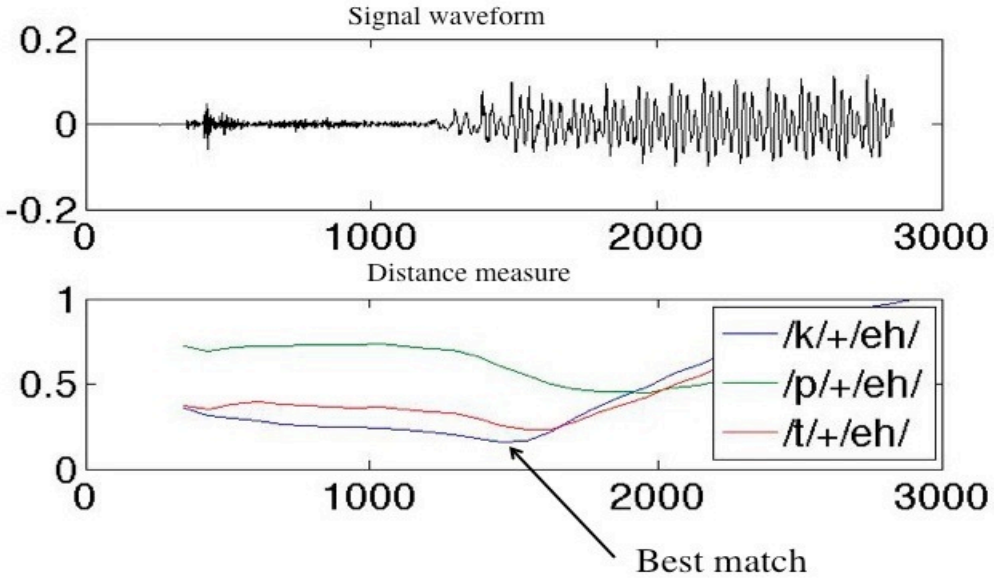

Figure 2.13: Recognition of the test sample. In this case the sample is assigned to the phone model [k].

Classification was also tested using the spectral model of consonant bursts in order to obtain a result that could be compared to the performance level of traditional methods. Again, one model for each consonant was created using an average of spectral models for all consonants in the training set. Classification was performed in a similar way as in the permutation-based method; the spectral model of the test sample was updated

incrementally and the sample was assigned to the class with the smallest distance measure.

Recognition accuracy for correct consonant in the context of vowel [AA] using different filtering conditions, different time lags, and time delays are presented in table 2.1.

Table 2.1: Recognition accuracies for consonants the context of the [AA]-vowel. Optimum results for each different configuration are shown in bold and underlined.

k=1	HP	No Filter	LP 7.2 kHz	LP 5.6 kHz	LP 4 kHz	LP 2.4 kHz
Lag 1	45	48	48	54	**57**	**57**
Lag 3	58	61	65	66	**70**	68
Lag 5	55	70	**71**	**71**	70	72
Lag 7	56	71	**75**	**75**	73	68
Lag 9	56	65	**72**	**72**	70	68
Lag 11	57	63	60	63	**65**	62
k=2	HP	No Filter	LP 7.2 kHz	LP 5.6 kHz	LP 4 kHz	LP 2.4 kHz
Lag 1	64	70	69	69	**70**	68
Lag 3	63	70	73	71	**77**	72
...						
Lag 13	69	80	**82**	**82**	74	72
Lag 15	69	**84**	**84**	82	76	72
Lag 17	72	80	**81**	78	76	67
FFT (n=1024)	79	**80**	**80**	77	74	75

Recognition results for correct consonant detection in context of vowel [EH] are in table 2.2.

Table 2.2: Recognition accuracies for consonants the context of [EH]-vowel. Optimum results for each different configuration are shown in bold and underlined.

k=1	HP	No Filter	LP 7.2 kHz	LP 5.6 kHz	LP 4 kHz	LP 2.4 kHz	LP 0.8 kHz
Lag 1	55	66	65	71	**76**	71	62
Lag 3	54	69	70	74	**81**	80	71
Lag 5	57	75	79	79	82	**83**	78
Lag 7	60	78	80	83	84	**85**	79
Lag 9	57	70	72	75	75	**76**	75

k=2	HP	No Filter	LP 7.2 kHz	LP 5.6 kHz	LP 4 kHz	LP 2.4 kHz
Lag 1	59	79	80	82	**84**	83
Lag 3	63	83	83	83	**87**	83
Lag 5	65	86	84	85	**87**	83
Lag 7	71	**87**	86	86	85	83
Lag 9	70	**88**	86	86	87	84
Lag 11	72	**86**	83	83	85	83
FFT(n=1024)	81	81	82	**83**	77	70

2.2.7 Discussion and conclusions

Recognition results show that the permutation transformation method efficiently codes relevant information at low frequencies. Recognition accuracy does not vary much if the signals are low-pass filtered while pre-emphasis reduces classification accuracy significantly.

Recognition results were better with a *time–delay* value of two rather than one while increasing the *time-lag* from 1 also had the effect of increasing recognition accuracy up to a certain point. These results suggest that the current permutation time window might be too short in length to collect all required and relevant information needed for consonant classification. However, increasing the window size would increase the number of possible permutations and the methods presented here would become computationally infeasible.

Recognition results using the permutation method can be compared with results obtained using spectral models. In this study they were compared to the classification rates found in literature (see, e.g., Niyogi & Sondhi, 2002; Ali et al., 2001). However, information from the following vowel is typically utilized in consonant-vowel recognition. Niyogi and Sondhi (2002) tested several different algorithms for detecting and recognizing stop consonants from continuous speech. Segmentation and consonant classification was divided into two separate problems and classification rates for consonant recognition ranged from 70 % to 90 %. Ali et al. (2001) used several different acoustics-phonetic features for phoneme representation and rule based classifiers for recognition and gained an 86 % overall recognition rate for stop consonant recognition.

In this study models for consonants were created from the consonant release (burst) part only and information from the upcoming transition to the vowel locus was not used. Recognition results were still found to be quite impressive since it is believed that the following vowel has a strong influence on overall stop consonant recognition. Having separate models for vowels would probably increase recognition accuracy. Also, it is possible to extend the method to biphones (a set of two adjacent phones) by modeling burst-vowel combinations. In any case permutations have shown to be a promising method to analyze and classify speech events directly in the time domain. The results also speak for the importance of the temporal fine structure and temporal order of the acoustic waveform.

2.3 Discovering fundamental acoustic units using DP-ngrams

2.3.1 Introduction

This report documents the work being carried out to automatically discover the fundamental units in speech. This work is inspired by Wolff (1982) who states that certain aspects of language acquisition occur as a result of 'Cognitive efficiency' and uses simple compression algorithms to illustrate this (Miller, 1956: notion of chunking). For an infant to learn language, he must successfully discover and store acoustic units that allow him to understand speech from his environment and in order for him to do this efficiently the infant must be able to dynamically optimize this set.

An efficient system tries to find the smallest number of units to explain the input by generalizing past experience. We hypothesize that phonemic contrasting units emerge as a property of an efficient system endowed with the ability to discriminate acoustic sounds, but there may also be smaller and larger units depending on the models environment. The Acoustic DP-ngram algorithm (Aimetti, 2009) is used to discover repeating patterns within the speech signal, while an active forgetting mechanism is used to filter out noisy templates from useful ones.

2.3.2 Pattern discovery

Pattern discovery is carried out using the Acoustic DP-ngram algorithm, this process has been covered in a previous deliverable (Laine et al., 2008) and is briefly outlined here for clarity. The input utterance is processed with all internal episodic templates ($X = \{x_1, \ldots, x_m\}$) stored in LTM. Figure 2.14 displays the accumulative quality score matrix (Q_{x1}) for template x_1 against the input utterance, the darker areas show similar acoustic stretches between the two sequences and longer stretches will accrue a higher final quality score

(q). By backtracking from the highest quality score (q_{max}) we can retrieve the longest local alignment, and repeating this process will give us a list of local alignments in order of highest quality (i.e. length). Once all the local alignments have been retrieved, up to a specified quality threshold, they are clustered into acoustically similar units through hierarchical agglomerative clustering. This clustering process is used as it does not require initializing the number of clusters a priori. Therefore, the system will be discovering and classifying large-scale patterns, such as sentences or words, and also small-scale patterns, such as syllables or phones. Each cluster is represented by the cluster centroid, which is the alignment with the shortest distance from all the others within the same cluster. We name the cluster centroids internal episodic templates (X).

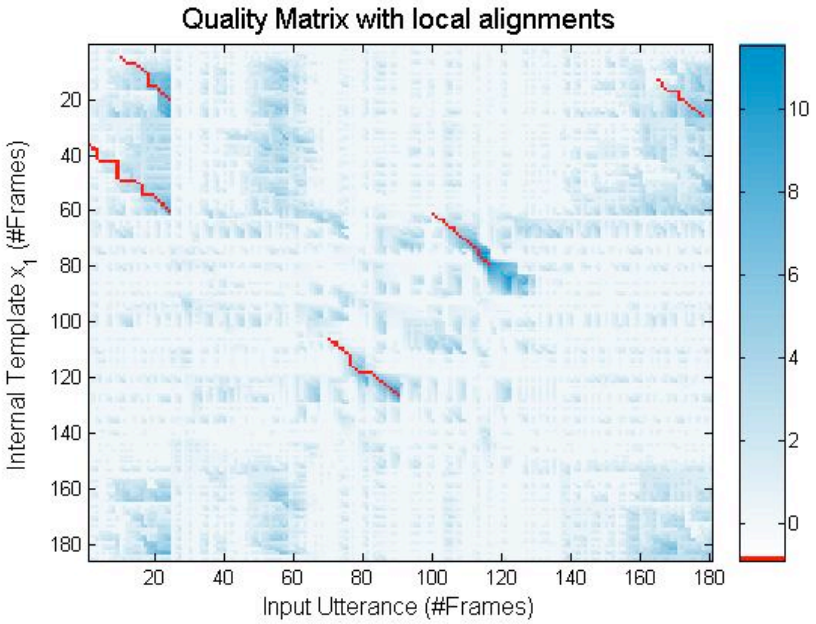

Figure 2.14: Quality matrix obtained from comparing the input speech signal and an internal representation from memory. The darker areas show higher quality scores and the red lines display discovered local alignments.

2.3.3 Recognition

Recognition is carried out by finding the optimal path through the input utterance using X. The quality matrix, calculated in the discovery stage, of each internal episodic template (Q_X) is used in order to keep the recognition and discovery process unified and reduce additional parameters. The optimal path through Q_X is calculated using dynamic programming, however, instead of finding the minimum cost path we search for the maximum accumulative quality score. In order to accumulate the quality score across templates we allow the score at the end of a template to be carried over to the beginning of the next.

Figure 2.15 displays the optimal path (the red continuous line) through the input utterance 'on the floor next to the bottle'. For clarity of this description, the example shown has been carried out on the orthographic representation of the speech. The x-axis displays the input utterance and the y-axis displays the set X. Template boundaries have been marked out using a dotted line and it can be seen that template jumps can only occur

at a boundary. The bottom frame of the y-axis is an additional 'unknown state' which is used when a portion of the input cannot be explained. Sequences of 'unknown states' are then stored in LTM for future use, thus allowing the system handle the 'out- of-vocabulary' problem by exploiting it.

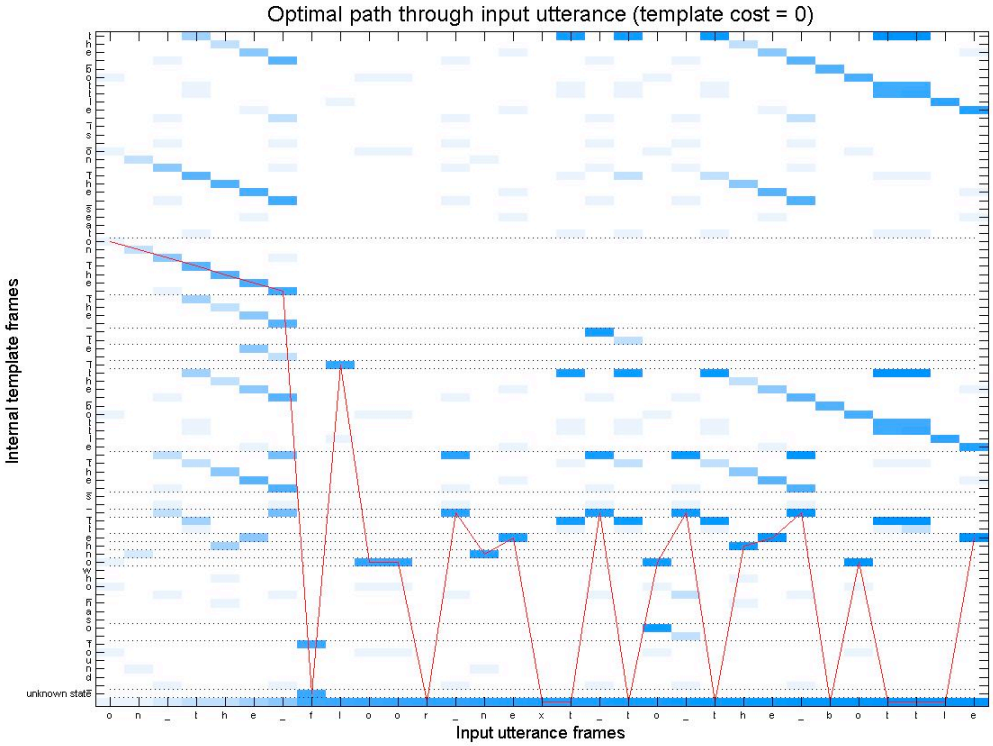

Figure 2.15: Optimal path through the input using the quality score matrix from all internal episodic templates.

Miller's notion of chunking states that frequently occurring groups are preferred to less frequent ones, and big ones are preferred to little ones (Miller, 1956). However, closer inspection of figure 2.15 shows that the system does not have a preference for longer templates as the portion 'the bottle' of the input utterance could have been explained with a single template (x_7). This problem can be solved by adding a cost for jumping out of a template. Figure 2.16 shows the optimal path through the input utterance with an additional template jump cost of -1. From the figure we can see that the system has now discovered the optimal path according to Miller (1956).

Figure 2.17 displays the optimal path after the template jump cost has been increased to -2. The system prefers to be in an 'unknown state' than use short templates, this shows that there is a balance between efficiency and the ability to differentiate meaning.

Figure 2.18 displays the optimal path through the input utterance using the set X on the acoustic signal. This path has been plotted after observing the same utterances as the orthographic examples and we can see that the system has successfully recognized the portion of speech containing 'the bottle' whilst allowing for temporal distortion.

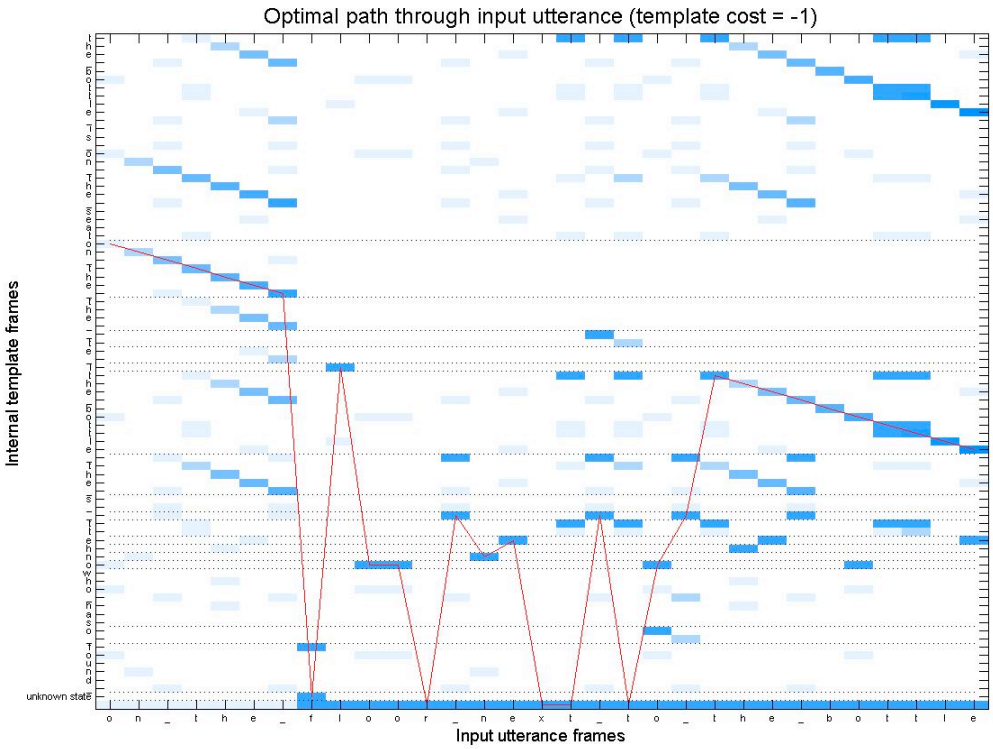

Figure 2.16: Optimal path through input utterance using internal templates with additional template boundary jump cost.

2.3.4 Active forgetting

Now that the system has a preference for longer internal templates we need to implement Miller's other criteria, that frequently occurring groups are preferred to less frequent ones. By implementing an active forgetting mechanism of not useful clusters we can prune our internal representations. According to Miller (1956), we can use frequency of occurrence as our 'usefulness' measure. This means that the templates that are commonly being used in the recognition stage of the learning process are reinforced or not forgotten. After some experience the system will possess a set of optimal templates that allow it to explain the input. The size of this set will be dependant on two variables:

1. The amount temporal distortion we allow during the discovery process

2. The cost of jumping between templates during the recognition process

Wolff (1982) suggests how we can measure the effectiveness of this set:

"The effectiveness of a grammar for compressing data (its 'compression capacity' or CC) is defined as (V-v)/V, where v is the volume, in bits, of a body of data after encoding by the grammar and V is the volume of the data in uncompressed form."

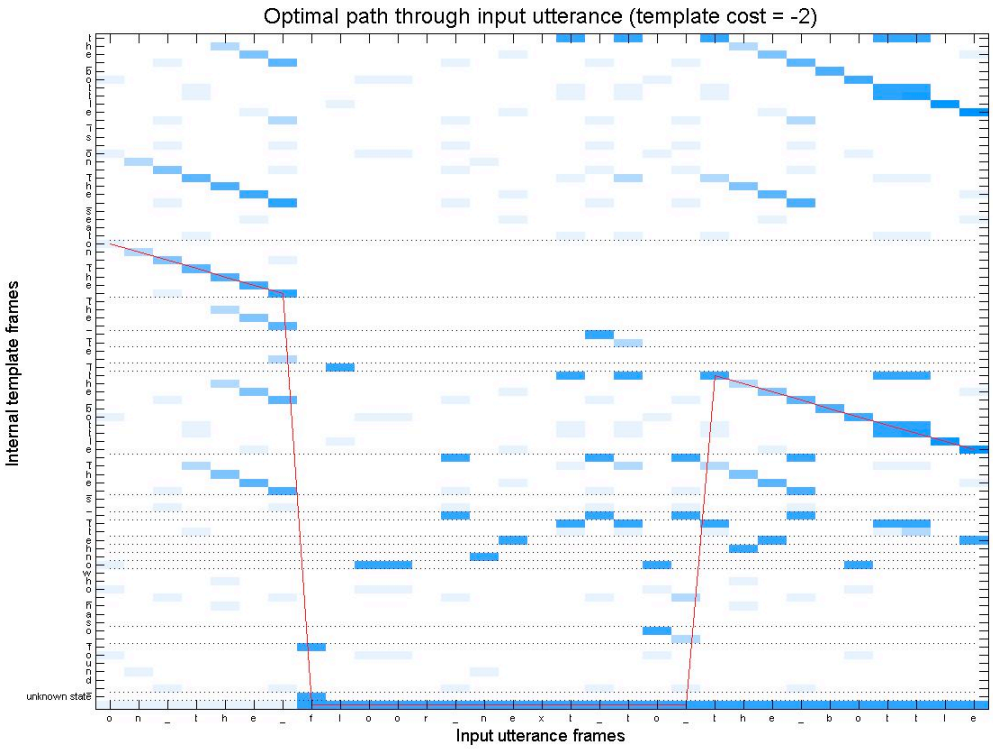

Figure 2.17: Optimal path through input utterance using internal templates with additional template boundary jump cost.

2.3.5 Discussion

This work proposes a general statistical learning mechanism for automatically discovering the fundamental units of speech. The system is not constrained to a pre-specified lexical units and is able to handle 'out-of-vocabulary' input, which are both huge problems for current state-of-the-art automatic speech recognition technology.

The DP-ngram process allows the system to build a suitable lexicon for its native language (as well as others), taking into account speech variation. The fundamental units arise an emergent property of the system interacting with its environment and striving for efficiency without compromising its ability to differentiate meaning. With experience, the systems internal representations are also allowed to dynamically evolve with the addition of the active forgetting mechanism.

Transcribing the input as a sequence of templates allows the system to store all or parts of it in an abstract form. In this abstract form it would be possible to build probabilistic models of these sequences, for example through ngrams, in order to use higher-level knowledge to predict future templates or force additional confidence weights.

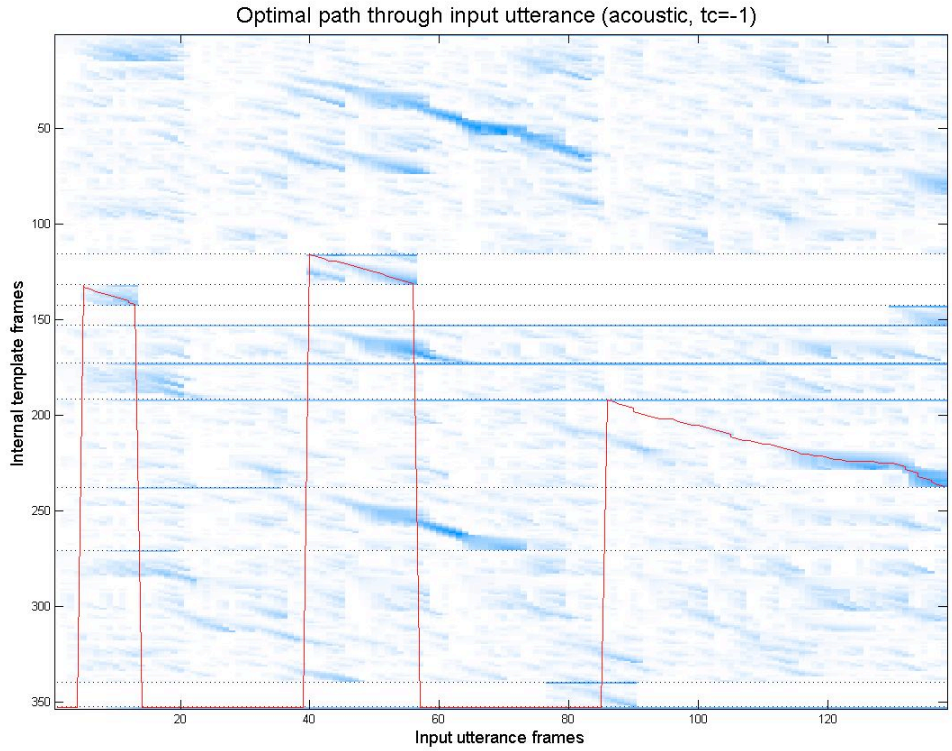

Figure 2.18: Optimal path through input utterance using internal templates with additional template boundary jump cost on the acoustic signal.

3. Computational Mechanics

Task 2 of ACORNS WP2 on Signal Patterning is concerned with investigating pattern discovery applications of a method known as computational mechanics. The approach comprises two parts: a theoretical representation of stochastic processes through so-called causal states, and a practical algorithm to infer the causal states of a process from empirical data.

3.1 Causal States

Computational mechanics (sometimes abbreviated CMM to avoid confusion with concept matrices CM) is a research area centering on the use of automata theory to describe patterns and the complex, stochastic processes that generate them. Central to these descriptions is the concept of *causal states* (Shalizi & Crutchfield, 2001; Shalizi, 2001). The causal states are equivalence classes defined over the possible *histories* $X_{-\infty}^{\ t}$ (sequences of observations from negative infinity until the current time t) of a given stationary, discrete-time stochastic process. Two histories belong in the same causal state if and only if they give the exact same beliefs about the future, i.e., if they imply the same probability distribution over all *futures* $X_{t+1}^{\ \infty}$ (sequences of observations from t+1 and on to infinity). The causal states are thus a partitioning of the set of possible histories.

One can show that the causal state representation $\varepsilon(X_{-\infty}^{\ t})$ is a *minimal sufficient statistic* for the observation sequence (Shalizi & Crutchfield, 2001); it retains precisely all information from past observations relevant for predicting the future, and nothing more. Thus ε satisfies $I(\varepsilon(X_{-\infty}^{\ t}); X_{t+1}^{\ \infty})=I(X_{-\infty}^{\ t}; X_{t+1}^{\ \infty})$. Moreover, appending a symbol to a history string gives a new history string that also belongs in some causal state. This way it is possible to define transitions between the states.

Interestingly, the states and their transitions together constitute a Markov process—even if the original process is not Markovian. Unlike HMMs (but similar to Markov chains), the current state can, in this description, be uniquely identified from the available sequence of observations. The entropy $H(\varepsilon(X_{-\infty}^{\ t}))$ can be used as a measure of the complexity of the X-process.

As an example, we will consider the even process of Weiss (Weiss, 1973) in Figure 3.1. This is a process over a two-symbol alphabet {0,1} which is actually a simple concatenation of strings chosen equiprobably from the set {0,11}. There are two causal states, A and B, which can be identified (almost surely) by whether an even or odd number of ones have been observed since the last zero; the states therefore contain the strings $A=\{*0(11)^n\}$ and $B=\{*0(11)^n1\}$, where n goes over all nonnegative integers. We notice that the causal state representation is very compact, even though this is a so-called *strictly sofic process*, meaning that it cannot be represented as a Markov chain of finite size.

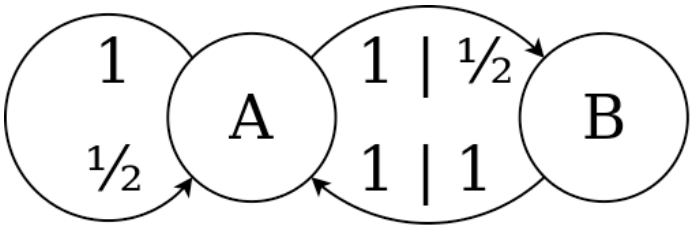

Figure 3.1: The even process due to Weiss (1973), which cannot be represented by a finite size Markov chain, yet has a compact causal state representation.

3.2 The CSSR Algorithm

Causal states are a theoretical construct that assumes full knowledge of the underlying process. Fortunately, in practice an approximation of the causal states can be learned from one or more empirical data sequences using the so-called *causal state splitting reconstruction algorithm* (CSSR) by Shalizi, Shalizi, and Crutchfield (Shalizi & Shalizi, 2004; Shalizi et al., 2002). Under some conditions, this procedure converges on the true set of causal states, given enough data. However, the algorithm only operates on sequences of discrete symbols from a finite alphabet.

Unlike the theory, practical algorithms can only access histories of finite length. CSSR, in particular, only considers the $L_{\max}$ most recent symbols at any given point in the data, known as a *suffix* of the history string, where $L_{\max}$ is a user-set memory length parameter. Despite this limit on suffix length, CSSR can actually learn certain processes with a non-fixed, potentially infinite memory, such as the even process of Weiss (Weiss, 1973). However, asymptotic convergence requires that the number of causal states is finite and that $L_{\max}$ is not set too low.

In brief, the CSSR algorithm consists of three main steps: parsing the data, homogenization, and determinization. These are outlined below (see also Shalizi & Shalizi, 2004; Shalizi et al., 2002).

1. To parse the data the algorithm simply counts the number of occurrences of all N-grams in the data with a length shorter than or equal to $L_{\max}+1$. These can be arranged into a tree, so CSSR belongs to the class of so-called *context tree* or *suffix tree* methods. This class also includes variable length Markov models like those discussed in Ron et al. (1996), and lossless compression algorithms such as Kennel & Mees (2002).
2. During homogenization CSSR iteratively looks at longer and longer suffixes, up until length $L_{\max}$, and collects these together into states based on what distribution they give for the next symbol X_{t+1}. The assignment is based on a statistical test, for example two-sample chi-squared, comparing the distributions. The test is carried out at a level α, a second user parameter. There is a bias for placing suffixes in the same state as their *parent suffix* (the suffix obtained by removing the oldest symbol from a given suffix), to prevent unnecessary splits. The result of homogenization is known as *precausal states*, since they can predict optimally one step into the future.
3. Determinization, finally, splits the precausal states to create a set of states that also has deterministic transitions: if suffixes in a state end up in more than one state after appending some positive-probability follower symbol, this is a non-

deterministic transition and the state has to be split, one piece for each possible follower state. This may cause other, previously deterministic states to become non-deterministic, but the procedure must eventually terminate. Determinization is necessary since causal states represent our beliefs about the future, and these change in a deterministic way with each new symbol we observe. Therefore transitions between causal states must be deterministic as well, given the next symbol. The set of states obtained after this step is able to predict the entire future optimally, assuming the precausal states were correctly partitioned.

Before and after determinization, CSSR also identifies any *transient states*, which are states that typically only are visited a finite number of times even if the output automaton is run for an infinitely long time. These cannot be true causal states and are therefore removed. The procedure chosen to identify state transitions and thus transient states (the "closure" mentioned in the ReadMe from Shalizi (2008) has an impact on the learnability of the even process.

The output of CSSR is a so-called *deterministic finite automaton*, or DFA, which is a discrete state Markov representation. Unlike HMMs, where transitions and observations are conditionally independent given the current state, the next state in a DFA is a deterministic function of the current state and the next observation. Once enough data is available to identify the current causal state, the state of the process is known exactly from then on. Also, in contrast to HMM training with the EM-algorithm, the number of causal states and their transition structure are recovered automatically, with few assumptions and without the need to specify a parametric model. CSSR thus performs unsupervised *pattern discovery*, not merely pattern recognition.

3.3 Limitations of CSSR

The WP2 ACORNS work has uncovered a number of practical limitations with the causal state concept and the CSSR procedure that reduce their usefulness in speech recognition and language acquisition. To begin with, CSSR is a batch learning algorithm that is difficult to adapt to the incremental learning scenario typical of language acquisition. While the parse tree is easy to update on-line, the number and nature of the statistical tests performed for each suffix during homogenization depend crucially on the outcome of previous tests. A small change in the parse tree statistics can change the outcome of a statistical test, bringing a chain of consequences that are difficult to assess without rerunning the algorithm. The determinization step is similarly dependent on previous decisions in a non-obvious manner.

Another problem is the computational requirements: since each node in the parse tree may have a full set of children down to the depth $L_{max}+1$, the worst-case requirements for data and computational power are exponential in L_{max}. On the other hand, since the data is read sequentially once, computational complexity is linear in the data sequence length.

As the rate of increase for the computational requirements depends on the cardinality of the symbol alphabet, it appears CSSR-based algorithms for speech would likely be most successful on a phone or phoneme level, where the dictionary is small and historical correlations not too strong (so that L_{max} can be kept low).However, in the first

year ACORNS recordings the number of words was similarly small and the sentence structure very regular, so it made sense to try CSSR on the word level for this particular case.

Figure 3.2 shows the successful results of applying CSSR to a transcript of these recordings, wherein each word (distinguished by spelling) was assigned a separate symbol, plus one additional symbol used to denote inter-utterance silence. This produced a data sequence of 4,295 symbols drawn from a 23-symbol alphabet.

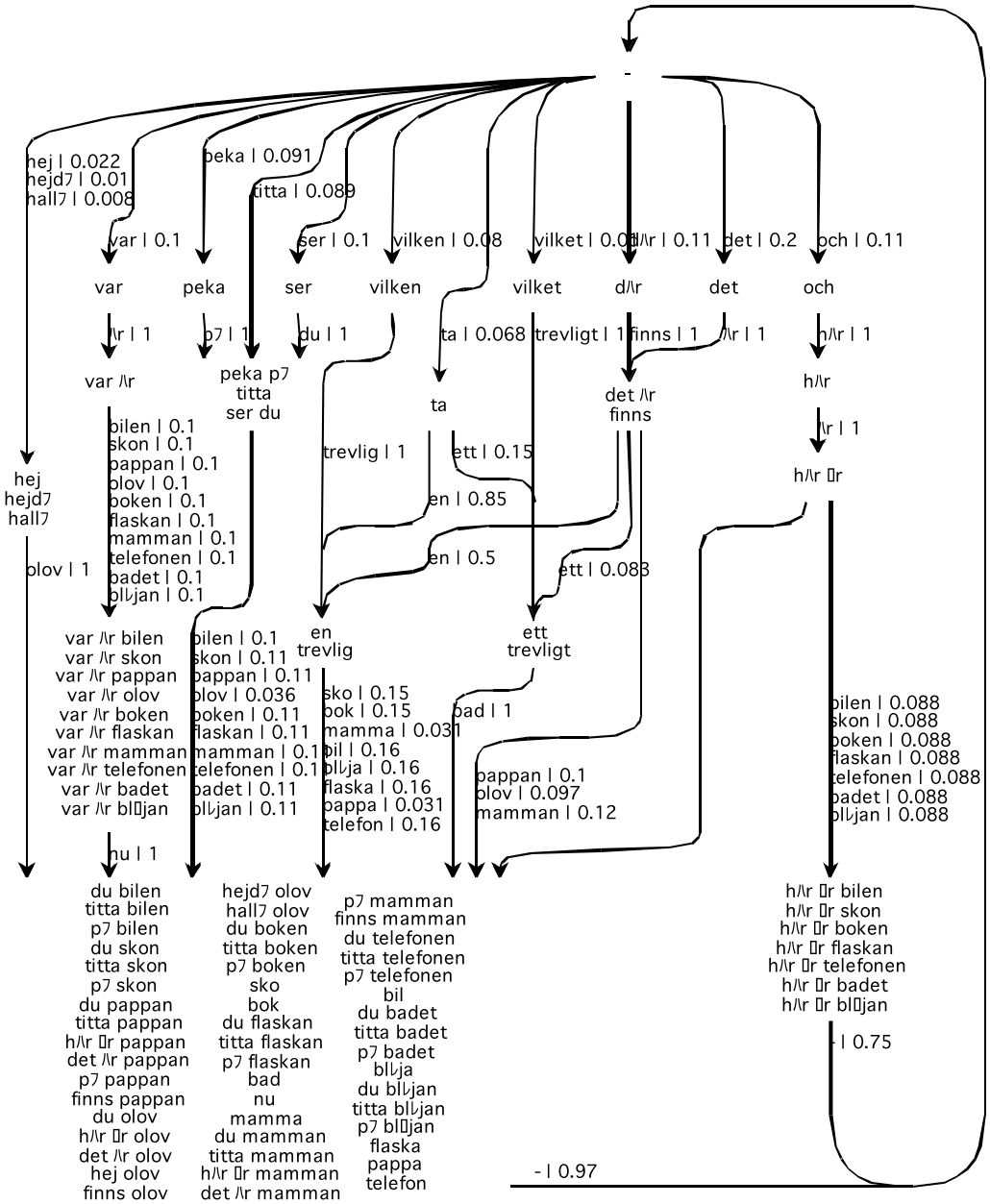

Figure 3.2: Reconstructed word-level automaton with Y1 Swedish data (L_{max}=4, α=0.002). The existence of two separate end states indicates a likely problem with the original CSSR implementation obtained at Shalizi (2008), necessitating a reimplementation of the algorithm for ACORNS.

As seen in the figure, CSSR learned a near-perfect automaton representation of a stationary stochastic process to generate the sentences in the observed data. Each state typically represented a specific word or position within one of the carrier sentences. Getting the desired conclusion required setting the algorithm parameters just right, but this problem diminished in importance if additional training data was randomly generated from the same bag of sentences.

Unfortunately, the same good results did not persist when CSSR was applied to a more realistic dataset, highlighting another significant weakness of causal states: poor robustness against noise. This was tested by applying CSSR to one million symbols from a simple model of speech as a slightly noisy sequence of randomly occurring words. Every word comprises a sequence of 'phones' (symbols) taken from a pre-generated random word dictionary of ten short symbol strings, each 4–8 symbols long. Eight different phones were used. The data was corrupted by low-probability (P=0.05) symbol substitution noise.

In stark contrast to the result on noise-free data, the algorithm here failed to converge on a limited set of causal states. Instead, the number of reconstructed states now grew steeply as larger and larger values for the memory length parameter L_{max} were considered, with no end in sight. The computing power requirements also increased prohibitively quickly. The same behavior persisted for similar language models with reduced word lengths, shorter word lists, and smaller alphabet sizes, as long as noise was present. Similar divergence with increasing L_{max} has also been observed by a group studying the applicability of CSSR for natural language processing tasks such as Named Entity Recognition, on a data corpus derived from Spanish text (Padró & Padró, 2005).

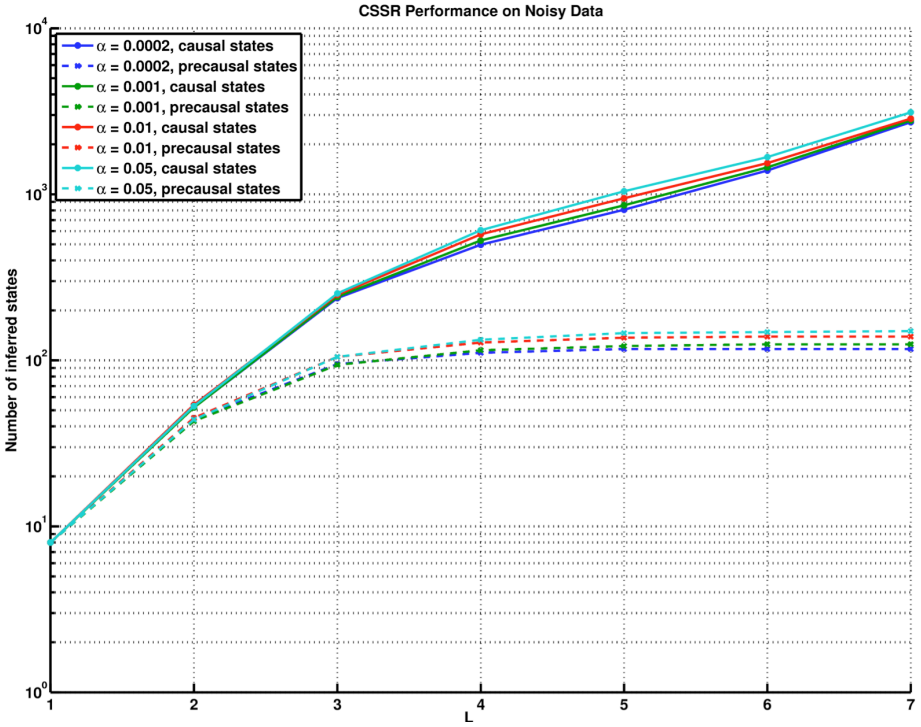

Figure 3.3: Unchecked growth in the number of reconstructed states as a function of $L_{\max}$ for a noisy test dataset with one million symbols from a size eight alphabet. The number of precausal states (before determinization) are dashed.

3.4 Causal States and Noise

The essence of the divergence problems of CSSR appears to be that the algorithm insists on capturing all the information from the past that is relevant for future behavior. Typically, the further back the algorithm looks at a dataset, small additional pieces of information comes to light that very slightly influence future behavior. All samples are thus informative, and memory length is infinite. Furthermore, the noisy speech model has infinitely many possible probability distributions for the future, all of which CSSR has to distinguish and represent by states of their own, even if their differences are so small that a human would label them as noise; the algorithm's definition of "pattern" versus "noise" is not the same as our own. In the end, this leads to an explosive increase in the number of output states as the memory length L_{max} grows.

Theoretical work within WP2 has established that this "infinite possibilities problem" occurs even with simple parametric models such as HMMs. We have developed a novel algebraic criterion that can be used to prove that many HMMs cannot be represented by a finite number of causal states. An example is the simple flip automaton in Figure 3.4 disturbed by random substitutions. The resulting noisy process can be described by a four-state HMM which provably cannot be learned by CSSR because its causal state representation is infinitely big. This highlights a subtle but important point that "structure" is not a singular concept and can be defined in many ways. In particular, the sense of structure dictated by causal states is often infinitely complex and difficult to use.

The large representations returned by CSSR are not only difficult to store in memory; there is also a clear risk of overfitting as the information in the sample data is divided between all the output states (while many of the causal states are typically quite similar, the base form of CSSR is not designed to take this into account). Results by Padró & Padró (2005) show that, whereas the number of states increased rapidly in $L_{\max}$, the best system performance was actually attained with a very small number of states at $L_{\max}$=3, the shortest memory length used. Evidently, a smaller representation that does not distinguish all causal states can achieve better performance for realistic sample sizes.

3.5 Robust Causal State Discovery

The problem with random substitutions, insertions, and deletions applied to an otherwise CSSR-learnable process, such as the one in Figure 3.4, can be traced to the next-step distributions that define the precausal states. Noise disturbs these distributions, causing suffixes previously in the same precausal state to separate. However, if the disturbances are not too great, these distributions will still be clustered closely together in the space of possible next-symbol distributions..

We have developed a modified version of the homogenization stage in CSSR, known as *robust homogenization*, where the resolution of the statistical test does not become infinitely sharp even as arbitrarily much data becomes available, but instead can be adjusted with a user-set parameter $n_{\max}$. This enables suffixes with similar next-step distributions—typically those that belonged in the same precausal state before being disturbed by noise—to be brought back together.

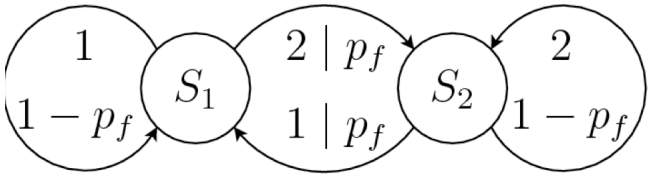

Figure 3.4: The flip automaton. A simple CSSR-learnable process that cannot be learned if disturbed by random symbol substitutions, even though the noisy process has a compact HMM representation.

Robust homogenization in combination with determinization as in regular CSSR forms the *robust causal state discovery* (RCS) algorithm. If the noise is not too large and the parameters are set correctly, this algorithm is provably able to recover the finite suffix partitioning corresponding to the causal states of the undisturbed, underlying process, albeit with slightly altered next-symbol probabilities. This is in contrast to the somewhat similar method in Schmiedekamp et al. (2006), which does not perform any determinization, and therefore often produces output with non-deterministic transitions that cannot be the causal states of *any* process.

We have prepared these results on learnability and robust causal state discovery into a paper, and are working on disseminating the information. We have additionally written a C++ implementation of the method, the source code of which will be made publicly available alongside the paper. This code is capable of performing both CSSR and RCS, and is also significantly faster than the CSSR implementation provided at Shalizi (2008) by the inventors of the original algorithm.

3.6 Continued Work

The RCS method is not a panacea, however, and is only certain to be useful in situations where a simple, CSSR-learnable process (i.e., one that satisfies the three criteria on page 4 of Shalizi & Shalizi, 2004) is disturbed by a moderate amount of random substitutions, insertions, and deletions. In the case of more realistic data or other noise conditions, divergent growth and overfitted models can still emerge.

Particularly troublesome in practice is the determinization stage, where the precausal states of natural systems typically have to be split many times before a fully deterministic configuration is reached, as seen in Figure 3.3. This complication is not easily bypassed by introducing resolution or stopping determinization early, since any non-determinism in the output takes away the interpretation that the current history (suffix) uniquely determines the current state, a fundamental assumption necessary for CSSR to learn the properties of the output states as collections of suffixes in the first place.

It seems that other ideas for finding approximate causal states are required. For this, we have turned our attention towards the concept of lossy compression. Clearly, by retaining all information that is relevant for prediction, CSSR has to learn an impossibly big representation that also includes properties of the input that we consider to be noise. It can be argued that learning and generalization, as opposed to memorization, must be a lossy process, where the goal is to identify information in the data that can be discarded, so that only the salient parts are kept. This will give a more compact (compressed) representation, that presumably also can be more useful, seeing how the results in Padró

& Padró (2005) show that a smaller non-sufficient statistic can achieve better performance than bigger models for realistic sample sizes.

The information bottleneck (IB) framework due to Tishby, Pereira, and Bialek (1999) is an approach for lossy prediction that appears to be an ideal candidate for obtaining an approximate extension of causal states. Similar to causal states it is grounded in information-theoretic considerations, specifically the information from a variable X that is predictive about another variable Y. However, it does not necessarily maximize this information, but instead seeks a statistic $T(X)$ that minimizes the functional $L(T)=I(X; T(X))-\beta I(T(X); Y)$ where the parameter β controls a trade-off between the predictive relevance of T and the "size" of T; the algorithm is willing to add 1 bit of additional complexity ($I(X; T(X))$) to T in exchange for β^{-1} or more bits of predictive information ($I(T(X); Y)$).

The minimal sufficient statistic, capable of optimal prediction, is obtained in the limit $\beta\to\infty$ (Shalizi & Crutchfield, 2002; Shamir et al., 2008), but finite parameter values may yield a smaller representation where only the most informative bits pass through the bottleneck. The other extreme, where $\beta\leq 1$, returns the trivial 0-bit predictor. Like the causal state ε, T can be considered a partitioning of the observation space of X. However, the partitioning here is "soft," i.e., potentially stochastic.

Apart from relations to causal states, information bottleneck also has interesting connections to source coding. Specifically, the optimal statistics for different β trace out a smooth, convex curve in the $(I(X;T), I(T;Y))$ plane, known as the information curve $I_X(I_Y)$, similar to the well-known rate-distortion function in lossy source coding. Furthermore, the Blahut-Arimoto algorithm, used to calculate a point on the rate-distortion curve in source coding, can also be adapted to iteratively find a (locally) optimal bottleneck statistic T. The resulting procedure is known as the *bottleneck equations*.

The IB framework is quite general and has seen many applications and specializations. However, only recently has it been considered to apply IB in the case of sequence data (for a Gaussian process), where X is the history and Y is the future (Creutzig et al., 2009). Interestingly, it appears that the case of discrete, stationary symbol sequences, which is the domain of causal states, has not yet been addressed.

We are currently working on combining the information bottleneck concept with the Markovian aspect of causal states, and believe this would be an excellent fit: Even if the information from the past relevant for predicting the future, $I(X_{-\infty}^{t}; X_{t+1}^{\infty})$, is infinite, the information passing through such a bottleneck may be finite and learnable. This looks like a promising approach to finally achieve practically useful approximate causal states. For our first step we are presently investigating IB applied to simple parametric models such as Markov chains.

References:

Aimetti G.: Modeling early language acquisition skills: Towards a general statistical learning mechanism. *Proceedings of the Student Research Workshop at EACL 2009*, pp. 1-9, Association for Computational Linguistics, 2009

Ali A. M. A., van der Spiegel J., and Mueller P.: Acoustic-Phonetic Features for the Automatic Classification of Stop Consonants. *IEEE transactions on speech and audio processing*, Vol. 9, No. 8, pp. 833-841, 2001

Aristoteles: VI, Metafysiikka. Gaudeamus, Helsinki 1990, pp. 324, ISBN 951-662-492-8 (in Finnish). See also: Aristotle, Metaphysics, http://classics.mit.edu/Aristotle/metaphysics.html

Bandt C.: Ordinal time series analysis. *Ecological modelling*, Vol. 182, pp. 229-238, 2005

Creutzig F., Globerson A., and Tishby N.: Past-Future Information Bottleneck in Dynamical Systems, *Physical Review E*, Vol. 79, No. 4, 2009

Feldman N. H., Griffiths, T. L., and Morgan, J. L.: Learning phonetic categories by learning a lexicon. *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, in press

Groth A.: Visualization of coupling in time series by order recurrence plots. *Physical Review E*, Vol. 72, No. 4, 2005

Hand D. J. and Bolton R. J.: Pattern discovery and detection: A unified statistical methodology. *Journal of Applied Statistics*, Vol. 31, No. 8, pp. 885–924, 2004

Hillenbrand J., Getty L. A., Clark M. J., and Wheeler K.: *Acoustic characteristics of American English vowels*. Journal of Acoustical Society of America, Vol. 97, No. 5, pp. 3099-3111, 1995

Keller K., Sinn M., and Emonds J. Time series from the ordinal viewpoint. *Stochastics and Dynamics*, Vol. 2, pp. 247-272, 2007

Kendall M.: A new measure of rank correlation. *Biometrika*, Vol. 30, No. 1, pp. 81–93, 1938

Kennel M. an. Mees A.: Context-tree modeling of observed symbolic dynamics. *Physical Review E*, Vol. 66, 2002

Kuhl P. K., Conboy B. T., Padden D., Rivera-Gaxiola M., and Nelson T.: Phonetic learning as a pathway to language: new data and native language magnet theory

expanded (NLM-e). *Philosophical Transactions B of the Royal Society,* Vol. 363, pp. 979-1000, 2008

Laine U.K., Räsänen O., Altosaar T., Driesen J., Aimetti G., and Henter G.: Methods for enhanced pattern discovery in speech processing. *ACORNS project deliverable*, http://lands.let.ru.nl/acorns/documents/Deliverables_Y2/Del%202.2.pdf, 2008

Miller G. A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, Vol. 63, pp. 81–97, 1956

Niyogi P. and Sondhi M.: Detecting stop consonants in continuous speech. *The Journal of the Acoustical Society of America*, Vol. 74, pp. 706-714, 2002

Padró M. and Padró L.: A Named Entity Recognition System Based on a Finite Automata Acquisition Algorithm. *Procesamiento del Lenguaje Natural*, Vol. 35, pp. 319–326, 2005

Ron D., Singer Y., and Tishby N.: The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length. *Machine Learning*, Vol. 25, No. 2–3, pp. 117–149, 1996

Räsänen O., Laine U. K., and Altosaar T.: A noise robust method for pattern discovery in quantized time series: the concept matrix approach. *Proc. Interspeech '09*, Brighton, England, 2009

Schmiedekamp M., Subbu A., and Phoha S.: The Clustered Causal State Algorithm: Efficient Pattern Discovery for Lossy Data-Compression Applications. *Computing in Science and Engineering*, Vol. 8, No. 5, pp. 59–67, 2006

Shalizi C.: Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata. *Ph.D. thesis*, University of Wisconsin, Madison, (As of 2009-10-24 available online at http://www.cscs.umich.edu/~crshalizi/thesis/), 2001

Shalizi C.: *CSSR: An Algorithm for Building Markov Models from Time Series*. Web page, accessed 2009-10-24 at http://www.cscs.umich.edu/~crshalizi/CSSR/. Last update in May 2008.

Shalizi C. R. and Crutchfield J. P.: Computational Mechanics: Pattern and Prediction, Structure and Simplicity. *Journal of Statistical Physics*, Vol. 104, pp. 819-881, 2001

Shalizi C. R. and Crutchfield J. P.: Information Bottlenecks, Causal States, and Statistical Relevance Bases: How to Represent Relevant Information in Memoryless Transduction. *Advances in Complex Systems*, Vol. 5, pp. 91–95, 2002

Shalizi C. R., Shalizi K., and Crutchfield J. P.: An Algorithm for Pattern Discovery in Time Series. Santa Fe Institute Working Paper 02-10-060 (As of 2009-10-24

available online at http://arxiv.org/abs/cs.LG/0210025)

Shalizi C. R. and Shalizi K.: Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences. In: M. Chickering and J. Halpern (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*, pp. 504–511, Arlington, Virginia, AUAI Press, 2004

Shamir O., Sabato S., and Tishby N.: Learning and Generalization with the Information Bottleneck. In: *ALT '08: Proceedings of the 19th international conference on Algorithmic Learning Theory*, pp. 92–107, Budapest, Hungary, Springer Verlag, 2008

Tishby N., Pereira F., and Bialek W.: The Information Bottleneck Method. In: B. Hajek and R. Sreenivas (eds.), *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377. September 1999

Van hamme H.: HAC-models: a Novel Approach to Continuous Speech Recognition. *Proc. Interspeech'08*, Brisbane, Australia, 2008

Varshney L. R. and Goyal V. K., Toward a Source Coding Theory for Sets, in Proceedings of the Data Compression Conference (DCC 2006), Snowbird, Utah, 28-30 March 2006.

Weiss B.: Subshifts of Finite Type and Sofic Systems. *Monatshefte für Mathematik*, Vol. 77, pp. 462–474, 1973

Werker J. F. and Curtin S.: PRIMIR: A Developmental Framework of Infant Speech Processing. *Language Learning and Development*, Vol. 1, pp. 197-234, 2005

Wolff J. G.: Language acquisition, data compression and generalization. *Language and Communication*, Vol. 2, pp. 57–89, 1982

Appendix A

Algorithm for alignment of multiple multidimensional time-series data

Input:

– N time-series $\mathbf{S_n}$ of dimension D and length l_n

– Maximum number of iterations

Output

– N aligned time-series $\mathbf{A_n}$ of dimension D and length l_A

– The amount of zero padding t_n added to each $\mathbf{S_n}$ to achieve the global alignment.

1) Take the longest time-series $\mathbf{S}_x$ as a reference.
2) Take $\mathbf{S}_y$ (y≠x) of length l_y and pad $\mathbf{S}_x$ with l_y zeroes at both sides.
3) Slide $\mathbf{S}_y$ over $\mathbf{S}_x$ frame by frame and compute dot-product of each frame in each window location to obtain distance vector $\boldsymbol{d}_{y,x}$.
4) Find maximum of $\boldsymbol{d}_{y,x}$ and compute required shift s accordingly.
5) If $s > 0$, zero pad $\mathbf{S}_y$ with s zeroes. If $s < 0$, zero pad $\mathbf{S}_x$ with s zeroes.
6) Add aligned $\mathbf{S}_y$ to the reference list with $\mathbf{S}_x$ and repeat steps 2-6 with a new $\mathbf{S}_{y+1}$, now computing distance to all aligned $\mathbf{S}$ in the reference list.
7) Once all time-series are aligned in the first pass, iterate steps 2-5 until convergence using all of the time-series from previous iteration as a reference.

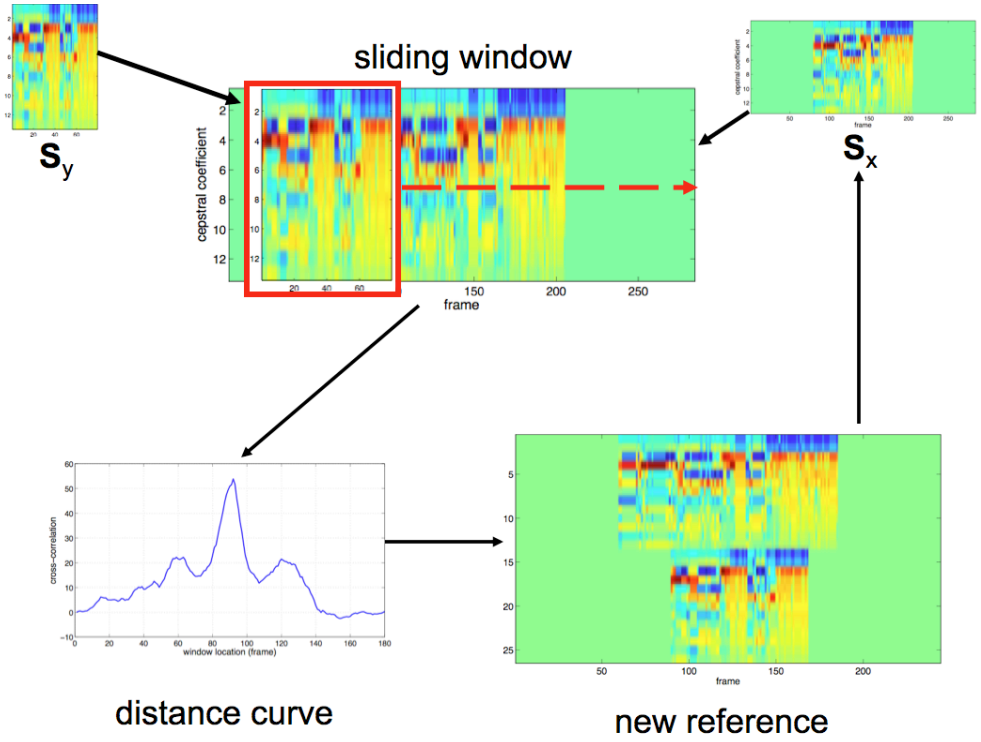

Figure A.1: A schematic for the first iteration in the alignment process. Spectrogram $\mathbf{S_y}$ slides over the reference $\mathbf{S_x}$ and their overall cross-correlation is computed at each point. Then the $\mathbf{S_y}$ is zero padded so that it will correspond to the point of maximum correlation in time. Finally, aligned $\mathbf{S_y}$ is added to the reference list with the previous reference $\mathbf{S_x}$. Now alignment can be performed for new input $\mathbf{S_{y+1}}$ by computing correlation to both references in $\mathbf{S_x}$.