A New Structure for Automatic Speech Recognition

by

Paul Duchnowski

S.B., Massachusetts Institute of Technology (1987)

S.M., Massachusetts Institute of Technology (1989)

Submitted to the Department of
Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1993

Signature of Author ____________________
Department of Electrical Engineering and Computer Science
August 31, 1993

Certified by ____________________
Louis D. Braida
Henry E. Warren Professor of Electrical Engineering
Thesis Supervisor

Accepted by ____________________
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

A New Structure for Automatic Speech Recognition

by

Paul Duchnowski

Submitted to the Department of Electrical Engineering and Computer Science
on August 31, 1993, in partial fulfillment of the requirements
for the Degree of Doctor of Science

Abstract

Speech is a wideband signal with cues identifying a particular element distributed across frequency. To capture these cues, most ASR systems analyze the speech signal into spectral (or spectrally-derived) components prior to recognition. Traditionally, these components are integrated across frequency to form a vector of "acoustic evidence" on which a decision by the ASR system is based. This thesis develops an alternate approach, *post-labeling integration.* In this scheme, tentative decisions or labels, of the identity of a given speech element are assigned in parallel by *sub-recognizers*, each operating on a band-limited portion of the speech waveform. Outputs of these independent channels are subsequently combined (integrated) to render the final decision.

Remarkably good recognition of bandlimited nonsense syllables by humans leads to the consideration of this method. It also allows potentially more accurate parameterization of the speech waveform and simultaneously robust estimation of parameter probabilities. The algorithm also represents an attempt to make explicit use of redundancies in speech.

Three basic methods of parameterizing the bandlimited input of the sub-recognizers were considered, focusing respectively on LPC and cepstrum coefficients, and parameters based on the autocorrelation function. Four sub-recognizers were implemented as discrete Hidden Markov Model (HMM) systems. Maximum A Posteriori (MAP) hypothesis testing approach was applied to the problem of integrating the individual sub-recognizer decisions on a frame by frame basis. Final segmentation was achieved by a secondary HMM. Five methods of estimating the probabilities necessary for MAP integration were tested.

The proposed structure was applied to the task of phonetic, speaker-independent, continuous speech recognition. Performance for several combinations of parameterization schemes and integration methods was measured. The best score of 58.5% on a 39 phone alphabet is roughly comparable to the published performance of traditional HMM systems and warrants further development. Potential sources of weakness of the approach, as implemented, are identified and improvements are suggested.

Thesis Supervisor: Louis D. Braida
Title: Henry E. Warren Professor of Electrical Engineering

Acknowledgements

I have been often told that my style of writing, especially on scientific matters, is too "folksy". So I started this page with high phrases of "intellectual challenge", "profound insights", and "unwavering support". And, strictly speaking, all of them, applied to the usual suspects, would have been perfectly accurate. But none of it sounded like anything yours truly would ever say except under duress.

I have no doubt that working for and (dare I say?) with Lou Braida made my graduate student experience better than that of many of my fellow students. Thesis advisors are granted certain, written and unwritten, privileges and, in return, are supposed to fulfill certain expectations. Lou rarely if ever made full use of the former and never failed in the latter.[1] And it was fun to be able to discuss multivariate analysis and Churchill's Balkan strategy all in the same conversation.

Al Drake, Cam Searle, and Rosalie Uchanski were the official readers of this thesis and the document is surely better for it. But the designation "readers" is incomplete at best. It's hard to explain this in a few lines (brevity being the soul of wit, which I appear to be lacking at the moment) so I'll just say that I learned a lot from each of them, long, in fact, years before the phrase `\chapter{Introduction}` appeared in my Emacs window.

If there is any blame to go around for this thesis having taken as long as it did, it must go to the seventh floor gang, past and present, known affectionately as the Sensory Communications Group. See, the place is just too good to leave. No incentives for a speedy departure here. However, since no good deed shall go unpunished, I am going to name a few names. Joe Frisbie, Matthew Power, and Kiran Dandekar many times "volunteered" their expertise on various software topics. Joe and Matt also kept `ollie` running smoothly and made sure he had unimpeded access to all the network goodies.

I want to acknowledge the support of the IMC, but Joe, as one of the co-founders, always reminds me that it's a shadowy organization eschewing publicity, so I better not.

Most of my graduate student career was aided and abetted by my office-mate Julie Greenberg. There have been a few lucky breaks in my life, and ending up in the same room with her for the last five years has definitely been one of them. Hey Julie, CDB !

A few more faces from outside the lab: Kathleen[2], Greg, Celia, and several Mikes. You know who you are and why you're here. Suzy, thank you for your friendship and the e-mail therapy.

Finally, thanks to my family who may not have always understood my motives but stuck by me through thick and thin, nonetheless.

[1] OK, there was a certain screen intensity incident, but that's now long forgotten.

[2] Who was the indirect inspiration for the flight of fancy on this page, having pointed out to me the bio of one John R. Davis, *Proceedings of the IEEE*, 67(6), p.950, June 1979.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The cues identifying speech tokens are distributed within the acoustic waveform in frequency and time. An implicit, intermediate goal of most automatic speech recognition schemes is the extraction or identification of these cues. Once the cues are obtained, deterministic or probabilistic matching rules are invoked that determine what utterance most likely gave rise to these particular cues.

As part of the matching process, one needs to specify the integration process that describes how the cues are combined in identifying speech elements. Focusing on the cues distributed in frequency, the traditional approach to cue integration is shown schematically in Figure 1-1. Individual cues are extracted from different regions of the spectrum. The cues are then combined in a multidimensional, *continuously valued* cue space. Finally, a mapping function matches the "point" in the cue space to a label that represents the final identification of the speech element.

The process illustrated in Figure 1-1 is quite general. It encompasses any cues that a system might extract from the acoustic waveform. The key characteristic is that cues from disparate frequency regions are combined together as continuously valued acoustic evidence prior to the system rendering an identification, a label, for the speech element. Consequently, this approach to automatic speech recognition will be referred to as *pre-labeling integration.*

Figure 1-2 shows an alternate approach. As before, cues are extracted from the acoustic signal spectrum. However, instead of integrating them in a cue space, the

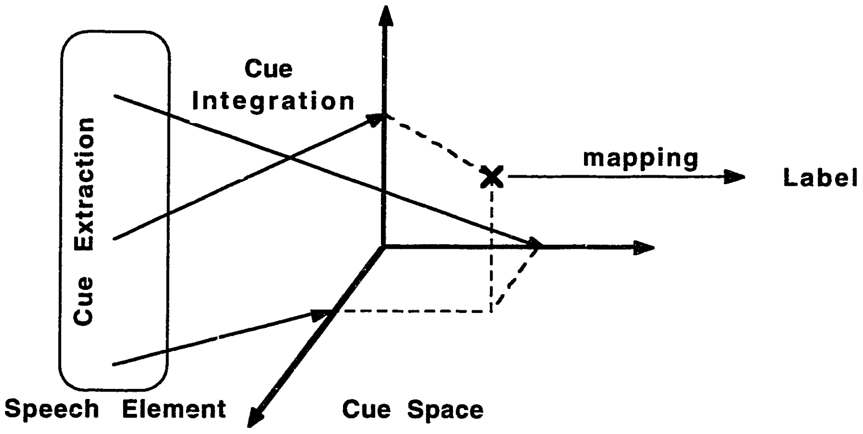

Figure 1-1: Pre-labeling Integration approach to ASR.

system immediately invokes labeling rules[1] for each cue (or possibly, subsets of the cues). This yields a set of *discrete valued* labels for the speech element that may be regarded as best guesses according to each cue. Finally, the labels are combined, again according to some rule, to produce the ultimate, single label for the utterance. This approach will be called *post-labeling integration.*

1.1 Rationale

At first glance, post-labeling integration would appear inferior to pre-labeling integration. With the former we lose the ability to compare cues across frequency or, put another way, to exploit across frequency correlation. A simple example might be the idealized vowel, characterized by the frequencies of the formants. These frequencies could constitute a perfectly reasonable set of cues. However, different vowels will exhibit very similar values of one or two formants. It is by comparing entire sets of formants that we can distinguish between different vowels, an approach not directly available in post-labeling integration. Nonetheless, there are a number of

[1] By "labeling rules" we mean any procedure that maps the continuously valued cues or parameters to a discrete label for the speech token. This procedure very well may be stochastic in nature.

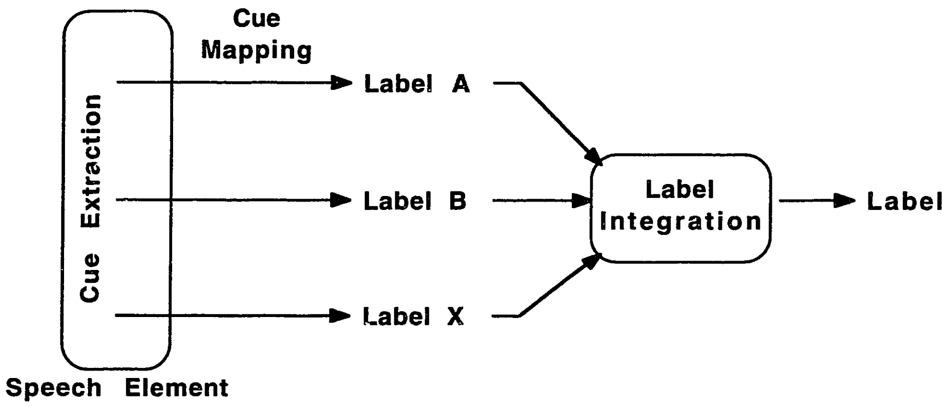

Figure 1-2: Post-labeling Integration approach to ASR.

considerations that suggest possible advantages of post-labeling integration.

In order to consider these tradeoffs, we have to be more specific about the manner in which we're going to separate the cues assumed to be distributed in frequency. A natural way to do this is to consider the band-limited speech waveform. Obviously, the signal so filtered will contain only a fraction of the cues present in the wideband signal. We may then consider how identification of speech elements based on the band-limited waveform would proceed. If we pick the bands such that together they span the entire speech spectrum then we will be satisfying the picture of Figure 1-2.

1.1.1 Human Performance

There have been relatively few studies of human recognition of bandlimited speech. There are at least two, however, whose results are relevant to the issue at hand. The first of these is the classic work of Miller and Nicely [53]. The stimuli comprised C-/a/ syllables where the consonant was one of sixteen most common in English. Depending on the test condition these stimuli were variously bandlimited. Each of five listeners was presented with 800 syllables for each condition and asked to identify the consonant. Table 1.1 lists the percentages of the consonants identified correctly

Frequency Band (kHz)	% Consonants Correct
0.2–0.6	49.5
0.2–1.2	57.2
0.2–2.5	72.8
1.0–5.0	73.1
2.5–5.0	38.1

Table 1.1: Average consonant identification by subjects in Miller and Nicely's study; 16 consonants in C-/a/ context.

Frequency Band	% Consonants Correct		
(kHz)	JG	JT	RM
0.7 lowpass	38	38	34
0.7–1.4	55	55	51
1.4–2.8	73	71	68
2.8 highpass	39	40	31

Table 1.2: Average consonant identification by three subjects in Milner's study; 24 consonants in CV context.

for some of the conditions tested.

The study by Milner *et al.* [54] also measured, among other conditions, band-limited consonant reception by normal hearing subjects. The stimuli consisted of 24 consonants paired with the vowels /a/, /i/ and /u/ in a CV context. The original, wideband stimuli had energy up to 4.5 kHz. They were presented to three subjects under four different filtering conditions. Table 1.2 summarizes the recognition scores achieved.

These results point to surprisingly high recognition rates achievable by humans even for relatively narrowband speech signal. While it is well known that speech remains perfectly intelligible when either lowpass or highpass filtered[2] at around 2500 Hz, what makes these data remarkable is the total lack of contextual information in the stimuli. Since the stimuli are purely nonsense syllables the only information available to the listeners is acoustic. This, in turn, leads to a conclusion that band-

[2] Consider, for instance, standard telephone transmission.

limited speech waveform still contains a large number of cues necessary to identify the speech elements without the aid of grammar, semantics, or context.

1.1.2 Modeling of Human Cue Integration

We may treat the foregoing as an existence proof of the feasibility of labeling of speech tokens based on cues from limited frequency regions. The question remains whether the labels so generated can be effectively combined. In a still larger context, can post-labeling integration compete with pre-labeling integration?

Braida [14, 15] developed and compared these two models to describe cue integration by humans. Originally this analysis was applied to integration of cross-modal cues. Specifically, the models sought to predict human audio-visual speech reception from performance on audio and visual stimuli separately. However, the methods are directly applicable to the more general question of how combination of cues from distinct channels of information may occur.

As suggested above, under the post-labeling integration model it is assumed that the listener makes decisions on the cues from each channel of information separately and then combines the decisions. Therefore, the multi-channel response should be predictable from responses in the constituent "uni-channel" modes. Suppose the stimulus $\mathcal{S}_i$ elicits the responses $\mathcal{R}_m$ and $\mathcal{R}_n$ when only information in channels A and B, respectively, is made available to the listener. The response to cues in a given channel is assumed independent of the other channels (including their presence or absence). The probability that the response pair produced when both channels are present is $(\mathcal{R}_m, \mathcal{R}_n)$ is:

$$\Pr{}^*_{AB}(\mathcal{R}_m, \mathcal{R}_n \mid \mathcal{S}_i) = \Pr{}_A(\mathcal{R}_m \mid \mathcal{S}_i) \times \Pr{}_B(\mathcal{R}_n \mid \mathcal{S}_i) \tag{1.1}$$

This procedure extends in an obvious fashion to more than two possible input channels.

The decision on the final identification $\mathcal{R}_j$ that results in highest identification accuracy follows the maximum likelihood rule[3]: $\mathcal{R}_j$ should be the identity of the stimulus for which $\Pr^*_{AB}(\mathcal{R}_m, \mathcal{R}_n \mid \mathcal{S}_i)$ is greatest.

In contrast, pre-labeling integration assumes that each stimulus channel evokes a continuous valued, noisy vector of cues $\vec{X}$. When only one channel is presented, the stimulus is identified as the *response center* $\vec{R}_k$ closest to $\vec{X}$. When multiple channels are available, the separate cue vectors are assumed to combine in one vector whose probability density is the "Cartesian product" of the individual channel cue densities. This compound vector is then compared to response centers in the multichannel cue space to determine the response. Cues are further assumed to combine optimally across channels without masking or interference.

Given listener performance on separate information channels, these two models can be used to predict response accuracy when the channels are presented simultaneously. To the extent that the predictions are close to actual multi-channel performance it may be said that a model accurately represents the process of cue integration by the subject.

Braida applied the models to human consonant reception, with the consonants in various CV contexts. The performance data are usually available as confusion matrices. The probabilities of Equation 1.1 can be readily estimated from these, simply as frequencies of responses given the stimuli. The problem of estimating the parameters of the pre-labeling models from confusion matrices is not completely solved but a procedure exists [13, 14].

As employed in the models, the concept of a channel is quite general. For example, they may represent different input modalities which is what was done in [14]. Here channel A represented the auditory component of the stimulus, while channel B stood for the visual part. The predictions of the pre-labeling models were found to be in closer agreement with actual audio-visual human performance than the predictions of the post-labeling model. Quite consistently, the latter tended to underestimate the audio-visual recognition.

Of more interest to the current research was the comparison of the two models

[3] We assume that a priori stimulus presentation probabilities are equal.

Frequency Bands	% Consonants Correct		
(kHz)	Pre-Label.	Post-Label.	Human Wideband
0.2-1.2 1.0-5.0	89.3	83.2	83.3
0.2-2.5 2.5-5.0	83.5	77.9	83.3

Table 1.3: Consonant scores from Miller and Nicely, compared to pre- and post-labeling predictions.

Frequency Bands	% Consonants Correct		
(kHz)	Pre-Label.	Post-Label.	Human Wideband
0.7 lowpass + 0.7-1.4	82.1	69.4	63.0
0.7 lowpass + 0.7-1.4 + 1.4-2.8	95.2	89.6	87.9
0.7-1.4 + 1.4-2.8 + 2.8 highpass	96.4	90.0	90.1
1.4-2.8 + 2.8 highpass	89.7	79.7	76.3
0.7-1.4 + 1.4-2.8	92.6	84.9	82.4

Table 1.4: Consonant scores from Milner (subject JG, 60-70 dB SPL), compared to pre- and post-labeling predictions.

when the information channels were chosen as non-overlapping frequency bands of the now purely audio stimulus. Tables 1.3 and 1.4 summarize the results of this comparison [15, 16]. In contrast to the audio-visual simulations the data here are more equivocal. Neither of the two models is clearly superior. Significantly though, the post-labeling model's predictions are in nearly all cases quite close to the actual multi-band performance. This seems surprising but one has to recognize that in order to effectively use the pre-labeling integration strategy, a listener has to be able to make accurate comparisons across frequency. It is not obvious that humans can perform such comparisons with sufficient accuracy.

Certain limitations of this demonstration have to be kept in mind, however. First, the stimuli whose recognition was well predicted by post-labeling cue integration were limited to isolated syllables. It is not immediately obvious how these observations would generalize to more complex speech input. Second, the success of the integration models does not gurantee that it is enough to achieve individual channel performance of the level in, say, Table 1.2 to be assured of the predicted multi-channel accuracy.

The integration process relies on a complementary distribution of errors among the individual channels, i.e., on the structure of the confusion matrices. For instance, if error patterns from two channels were perfectly correlated, the joint performance could not be better than with either channel alone.

On the other hand, the study does demonstrate that acoustic information alone[4] is sufficient for high rate of recognition and that it is feasible to consider combining the labels assigned to a speech element based on individual frequency bands in order to produce the final identification. Post-labeling integration is a viable option.

1.1.3 Potential Benefits for Automatic Recognition

The evidence that humans may accomplish speech recognition in the post-labeling mode does not of itself show whether an automatic recognizer operating in this mode would be competitive with the usual approach: pre-labeling integration.

Automatic speech recognizers usually derive cues from the acoustic speech waveform in the form of parameters such as filterbank energies, spectral or cepstral coefficients, etc. [21, 22, 59]. Just as the abstract cues referred to above these are continuous valued quantities. The task of the recognizers is then to find a partition of the parameter space that assigns appropriate regions within it to each speech element in the vocabulary or alphabet of interest.[5] The manner in which the partitioning occurs is the defining characteristic of different ASR algorithms.

In general, recognition systems attempt to "learn" the decision regions from known, manually-labeled speech. This is true of all three general ASR classes: Dynamic Time Warping (DTW) [68], Hidden Markov Models (HMM) [2, 4, 66, 69] and connectionist or neural network methods [47, 55]. A common problem is the relative paucity of realistically available training data in relation to the complexity of the parameter space that has to be partitioned. As a result the recognition systems must compromise between accurate representation of the acoustic input (requiring

[4] It bears stressing again that all the stimuli concerned were nonsense syllables.

[5] More realistically, one usually maps *sequences* of parameter sets since a speech element will span more than one. Insofar as a sequence may be considered as one large parameter set, the general picture of parameter space partitioning still holds.

many parameters) and robust estimates of parameter regions corresponding to specific speech elements (requiring many "samples" of the parameter space).

A good example of this tradeoff is the discrete HMM approach where the parameter sets are vector quantized and mapped to one of a finite number of prototype vectors.[6] The quantization process involves a loss of information, the so-called VQ-distortion. The more prototypes available, the smaller this distortion. However, the HMM formalism requires (roughly) the estimation of the probabilities of the occurrence of all available prototypes for all possible speech elements. For a constant amount of training data the robustness of these estimates decreases as the number of prototypes is increased.

The situation is quite different if we consider post-integration labeling. Suppose cues from different frequency regions are represented by parameters derived solely from the correspondingly band-limited acoustic waveform. Clearly, fewer parameters will be necessary to characterize a band-limited signal than the original wideband waveform. On the other hand the available amount of training data remains unchanged. The result should be a more robust estimate of the partitioning of the now reduced parameter space. Admittedly, within each band-limited cue group the potential for confusion among certain speech elements would increase. These would be the elements distinguishable mainly through information in other frequency regions. These, however, are also the confusions we expect might be reliably recoverable in the decision integration process.

The expected advantage may be viewed as an attempt to exploit the known redundancy of speech [79]. Pre-labeling integration recognizers are forced to suppress this redundancy in order to arrive at parameter sets small enough to be reliably classified. Post-labeling integration allows us potentially to retain more of the cue information at the cost of foregoing the ability to compare across cues. Instead we must compare across *labels.* One of the chief goals of this study was to assess the usefulness of this option.

One might argue that in order to get the benefit of reduced parameter set di-

[6] More detailed discussion of this method may be found in Sec. 3.1.

mensionality, one could simply compute wideband parameters as done in traditional speech recognizers, then divide the parameters into smaller sets and perform independent recognition on each of these subsets. However, there are no guidelines for how the assignment of vectors into subsets should be performed. On the other hand, we do have evidence from the human recognition experiments discussed above, of potentially high recognition rates for the narrowband cue groups.

1.1.4 Other Considerations

The notion of sub-recognizers acting on cues from independent frequency regions creates a potential for improved noise immunity of the entire system. In particular, unless the noise spectrum matches the speech spectrum, the noise would tend to affect separate sub-recognizers differently. If the character of this noise were known, the decision integration procedure could explicitly discount the output of the most affected sub-recognizer. Even if no such specific provision were made, one might still expect that the unaffected channels might compensate for the probably erratic guesses of the suspect channel. Nonetheless, this aspect of the recognition process was not investigated in this study.

The structure of the proposed approach might also be attractive for real-time implementation. The parallel nature of generating labels from independent cues increases the leverage to be gained from faster and/or more advanced hardware.

1.2 Motivation

This investigation of the feasibility of post-integration labeling approach to ASR was conducted in a specific context, i.e., a task with defined constraints, vocabulary, and a performance metric. A communication aid for the hearing impaired being studied in the Sensory Communications Group at MIT provided such a context. The aid's goal is to enhance speechreading by providing the listener with supplemental cues difficult to obtain from the visual signal.

1.2.1 Cued Speech

Speachreading is one of the most common methods used by individuals with hearing impairment for real time communication. Normal hearing individuals likewise use speechreading to assist communication in the presence of noise. The term refers to the observation of the movements of the talker's mouth (hence the sometimes applied term "lipreading") in order to discern what is being said. Unfortunately, speechreading, even under optimal conditions, is generally insufficient for reliable communication [80]. It results in mental strain for the listener and is also, alone, inadequate for language acquisition. In order to render speechreading more useful and informative some method of enhancement is necessary. Such a method, by resolving the ambiguities inherent in the "signal" available to the speechreader, would provide a clear and dependable system allowing learning of English as a spoken language.

A number of studies have addressed the issue of how much and what kind of information can be reliably extracted by a lipreader without any further aid. These investigations concentrated on establishing the minimal unit for speechreading analogous to the acoustically oriented phoneme. The visual counterpart has been dubbed the *viseme* and is defined as any individual and contrastive visually perceived unit that constitutes a minimal unit for visual speech perception [26, 37]. Just as for phonemes these units encompass allophonic variations although the differences among these are not so well understood as for acoustic phones. Each viseme tends to include several phonemes that appear the same to the observer. For example, the lips come together for each member of the /p, b, m/ viseme and are puckered for /u, U, ow/. This then, in the simplest terms, is the source of ambiguity for lipreaders.

While no universal system of visemes has been established, general categories, indicating most confusable visually phonemes, have been found by several investigators [10, 60, 84, 85]. These results suggest a straightforward approach to improving communication via speechreading: design a way of conveying which member of a particular viseme is spoken at a given moment. Cornett [18, 39] developed a method exactly along these lines. In his system phonemes of spoken English are divided into visually contrastive sets which are signalled to the speechreader manually. Consonants are divided into eight groups and vowels into four. Hand shape is used to define the consonant and hand position relative to the face of the talker signifies the vowel. In this way a consonant and subsequent vowel are signalled simultaneously, making the CV syllable the basic unit of Cued Speech. Diphthongs are expressed by moving the hand between the positions assigned to the initial and final vowels.

Manual Cued Speech has been shown to significantly improve speech reception for its users [17, 58, 81]. However, the system's usefulness and applicability are limited since in order to employ it, the talker must be able to produce the cues while speaking. To remedy this situation, Cornett and Beadles have been developing (since 1969) a device to analyze the speech received from the talker automatically and to provide the speechreader with the appropriate cues [19, 73]. The latest data available indicate that the speech recognition performance of this system is not yet good enough to result in a significant benefit to the user [8].

1.2.2 Recognition Task

The application outlined above served to specify the task for which the system would be designed. In principle one might merely require the recognition of cue groups. However, the optimal composition of these groups is still the subject of studies. It is likely that the optimal groupings would constitute a compromise between "disambiguating potential" and minimizing the adverse effect of recognizer errors. Accordingly it was decided to construct a phonetic speech recognizer from whose output any number of cue systems could be derived. In addition, there exist useful benchmarks for the performance of phonetic recognizers.

Further specifications on the recognizer arise from the consideration of practical aspects of an automatic cueing system. The recognizer should function in speaker-independent, vocabulary-independent, continuous speech mode. It also must be reasonably "real-time". While implementation issues were not dealt with directly here, this last limitation does constrain any potential algorithm from allowing any significant delays in order to benefit from "future" information. The delays that would be detrimental to a lipreader are hard to estimate since no appropriate study exists.

However, studies on audio-visual speech recognition [51, 61] found that delays of as little as 40 ms cause intelligibility degradation. The algorithm was developed using the 40 ms mark as maximum allowable "built-in" delay.

1.3 System Overview

Given the general philosophy of the recognition scheme and the context in which to test it, a more concrete structure may be suggested. Figure 1-3 shows the block diagram of the recognition system that was investigated in this study.

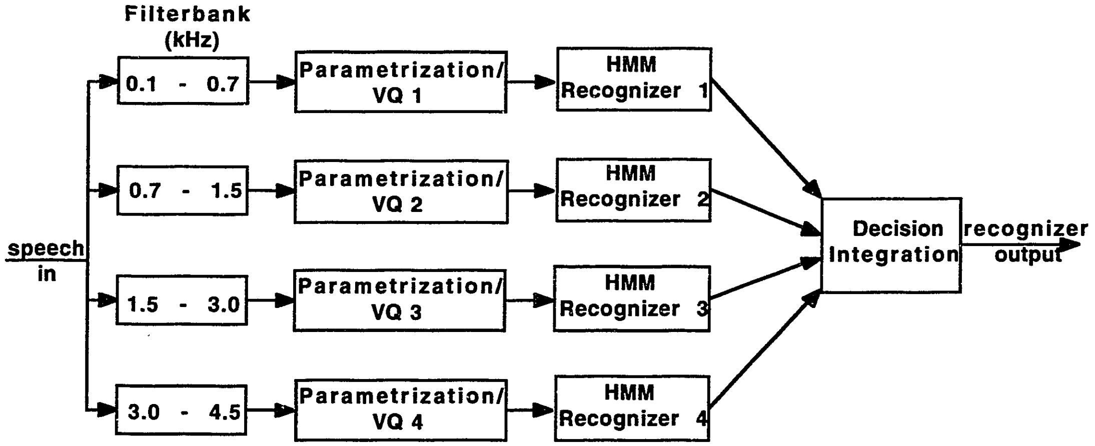

Figure 1-3: Block diagram of the proposed recognizer

In the initial filterbank stage the speech signal is bandpass filtered into four non-overlapping frequency bands. This represents the splitting of frequency-distributed cues. The bandwidths of the filters are given in the figure. They essentially match the bandwidths used in Milner's study (see Table 1.2). Additionally, the bandwidths correspond roughly to formant frequency regions. Formant locations being one of the principal established speech cues, it was deemed appropriate that a post-labeling system treat them separately.

At this point each "channel" of information is treated independently of the others.

In each one a data reduction step takes place to represent the acoustic input with a stream of parameter sets. This operation is common to virtually all speech recognizers, the difference here being again the parallel and relatively narrow-band nature of the parameterizations.

The parameter streams serve as input to the sub-recognizers corresponding to the *labeling* stage of the overall structure. In Figure 1-3 these are left unspecified. In principle we are free to use any classification algorithms at this stage as long as the outputs can be combined to yield the final answer. For instance, it would be inadvisable to use word recognition algorithms at this stage if the global goal is phonetic recognition. On the other hand, phonetic output from the sub-recognizers is perfectly compatible with the word recognition.

In the work presented here, the discrete Hidden Markov Model approach was used to implement the sub-recognizers.[7] This technology is currently achieving some of the best results on tasks similar to the one motivating our recognizer. It also offers computationally efficient algorithms for the training and especially recognition phases.

The outputs of the sub-recognizers (i.e. the labels) are then fed to the label integration stage, the final step in post-labeling integration. Here a combination rule is invoked to merge the individual channel outputs and produce the final stream of recognized phones. Note that in our application this process must not only integrate labels but also make timing and segmentation decisions. This is due to the independence of the sub-recognizers and the resulting likelihood that their decisions will not be synchronized or even that they will not produce the same number of labels.

[7] The structure also allows mixing of algorithms: the sub-recognizer in channel 1 may be of different type than the one in channel 2. However, there were no compelling or obvious factors that would suggest why a certain type of recognizer should work better for a certain frequency range. In the experiments performed here only the HMM-based recognition was used.

TIMIT Set	# of Speakers	Sentences	Sentences per Speaker	Use
TRAIN	462	SX	5	Sub-recognizer and Decision Integration training
		SI	3	Decision Integration training
TEST	168	SX	5	Testing at all levels

Table 1.5: Breakdown of TIMIT sentences as used in the current study.

1.4 Database

It has been said that when it comes to building statistically-oriented speech recognizers "there is no data like more data" [52]. This is true for HMM systems employed here as the sub-recognizers and, as will be seen, holds even more emphatically for the decision integration problem. The most readily available, large, multi-speaker database that has been phonetically transcribed is TIMIT [27, 40].

In this investigation the NIST edition of TIMIT [56], available on CD-ROM was used. The database contains a total of 630 speakers, each saying 10 sentences. The speakers represent 8 general dialect regions of American English. There are roughly twice as many males as females. Of the three sentence types the SA sentences (2 per speaker) could not be used since they are the same for all speakers. This would bias the materials towards certain specific phones and phone contexts. The remaining data consist of the SX (5 per speaker) and SI (3 per speaker) sentences. The former were specifically designed to provide a good coverage of pairs of phones while the latter were selected from existing written sources to add diversity to the corpus. Each of the SX sentences is spoken by seven speakers but each SI sentence occurs only once.

The NIST convention of dividing the database into the training and testing parts was followed here. Table 1.5 outlines the characteristics of the subdivisions of the corpus and how they were used.

The specifics of the database use will be described further in experimental sections. It should be noted that the training and testing materials were entirely disjoint: no speaker or text was included in both.

1.5 Implementation Note

Most of the computation described in this thesis was performed on DECstation 5000 and 3100 computers. In the development stage, especially parameterization of the acoustic input, a signal processing software package SPUD [65] was used.[8] Otherwise, all C code necessary for the implementation of the various algorithms, including the HMM procedures, was written and implemented by the author.

[8] This software, developed at the Sensory Communications Group at MIT, is available commercially as N!Power from Signal Technology Inc. of Goleta CA.

Chapter 2

The Front End

The components of a recognition system involved in initial processing of the speech waveform and its parameterization are often referred to as the front end. Its function is data rate reduction and extraction of the most salient, information bearing features.

Work done with HMM-based systems over the last decade has resulted in general acceptance of several signal processing steps that the front end comprises. These procedures frequently include pre-emphasis, segmentation of the speech waveform into frames, windowing of the frames, and computation of a set of parameters describing a particular frame. Some of these steps were directly applicable in the proposed system. In these cases the most frequently used methods in the field were chosen. On the other hand, given the new structure of the recognizer, certain aspects of the procedures had to be changed. A new parameterization scheme was also investigated.

In a sense, the current system used several front ends, one for each of the independent channels. Their functions included the initial filtering operation, division of the signal into frames, and generation of a data/parameter vector for each frame. However, except for channel-specific parameters like filter bandwidths, the computations themselves remained largely the same. Unless explicitly stated otherwise, this applies to all procedures described below.

Prior to processing by the recognizer, the TIMIT sentences were digitally lowpass filtered at 4.5 kHz and resampled for an effective sampling rate of 10 kHz.

2.1 Filterbank

HMM recognizers routinely pre-emphasize the input signal in order to flatten the spectrum and give equal weight to different frequencies in subsequent parameterization. This is largely superfluous in the present scheme since different frequency bands are parameterized independently. Thus the first operation performed by the recognizer is the splitting of the signal into these frequency bands, i.e., the filterbank.

The filterbank consisted of four digital FIR bandpass filters whose bandwidths[1] are shown in Figure 1-3. The filters were designed using the method of Schafer and Rabiner [75]. The prototype filter was a truncated impulse response of fifth order IIR Butterworth low-pass filter. This prototype was appropriately modulated and summed to produce the required band-pass filters. The method used allowed for minimal ripple and linear phase in the filterbank. The delay incurred was equivalent to 11.3 ms which was judged acceptable in this application.

The bandwidths of the filters were chosen to correspond directly to those used in the human recognition experiments summarized in Table 1.2. They were further considered appropriate since each encompasses roughly one formant region.

The cut-off characteristics and transition region of the filters are relatively sharp, similar to the filters used by Miller and Nicely and by Milner. The former study used filters with slopes of 24 dB per octave, the latter specified the stop-band of -60 dB to be reached in 200 Hz from the edge of the passband.

2.2 Parameterization

The popular practice of time-synchronous acoustic data processing was used here. In this method parameter vectors are computed for constant-length, usually overlapping segments of the acoustic waveform. In the current system the segments, often referred to as frames, were 20 ms long and overlapped by 10 ms. These durations give enough resolution to capture most articulatory events. On the other hand they are short

[1] Given the 10kHz sampling rate of the analog signal being filtered.

enough so that the speech signal within each frame may still be treated as stationary.

The ultimate goal of the perfect parameterization is to preserve the *invariant features* of speech, i.e. the metrics that remain constant for different realizations of a given utterance but are consistently different for different utterances. Unfortunately such a parameter set is still unknown. In the current research, generally agreed upon guidelines were followed in establishing the parameters.

A variety of parameterization schemes appears in the literature. They include energies in frequency bands [2, 21, 82], linear prediction (LPC) coefficients [72], reflection coefficients, low-time cepstra, and mel-weighted cepstra [43, 63, 76]. In a study by Davis and Mermelstein [22] the mel-weighted cepstrum parameterization yielded the best results. Other researchers have found it useful to supplement this information with parameters measuring power in the signal and with so-called delta parameters [28, 41, 42]. The latter approximate the rate of change of the static parameters in the vicinity of the frame.

Without exception these parameterizations have been applied to wide-band speech.[2] They were, therefore, not directly applicable to the task at hand. For instance, it makes little sense to mel-weight coefficients within bands as narrow as the ones proposed. On the other hand, the character of the generally used parameters suggests the type of parameterization needed.

It was concluded that the recognizer ought to be provided with three types of information about any given frame of speech: energy profile, spectral characteristics, and dynamic evolution of the first two. These will be referred to as energy, structure, and delta parameters. A number of arrangements were considered. Below are described the ones tested in the full system. All the parameters were computed on frames that had been scaled by the Hamming window $w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1})$, where N is the length of the frame in points. Since the data used was sampled at 10 kHz, $N = 200$.

[2] Telephone speech recognizers do limit the bandwidth but not to the extent of the channel bandwidths explored here.

2.2.1 Energy Parameters

Two parameters tracking the energy in the signal were used. The first, e_1: channel gross power, simply measured the log power in a given frame, normalized by the average power of the entire utterance. Thus, for an utterance s of length T and frame segment $x[n]$ of length N, parameter e_1 was computed as:

$$e_1 = \log(\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]) - P_{avg} \tag{2.1}$$

where P_{avg} is just:

$$P_{avg} = \log(\frac{1}{T}\sum_{n=0}^{T-1} s^2[n])$$

Since the training and testing materials consisted of sentences, this definition of average power was convenient and meaningful. In running speech one would compute the average power over a reasonable interval immediately preceding the frame under consideration, a few seconds say.

The second energy parameter, e_2: channel energy share, represented the only information about the wideband speech signal available to a given channel recognizer. It was computed as the ratio of the energy in the bandlimited frame to the energy of the wideband signal within the same time segment, the "wideband frame". Denoting this wideband frame signal by x_w, e_2 was given by:

$$e_2 = \frac{\sum_{n=0}^{N-1} x^2[n]}{\sum_{n=0}^{N-1} x_w^2[n]} \tag{2.2}$$

As might be expected, e_2 was mostly high, i.e. around 0.9, for the first channel because of the absence of pre-emphasis. However, since it was not being compared in any way to e_2's of other channels this was not a concern.

2.2.2 Structure Parameters

The purpose of this set of parameters is to capture the finer details of the waveform within a frame. If we accept the source-filter model of speech [67], we are lead to the conclusion that the signal in each channel is determined primarily by one or at most two resonances of the vocal tract. Furthermore, twelve to sixteen coefficients (for instance, LPC or cepstrum) are generally found to describe well the spectral characteristics of a wideband speech frame. These observations suggest that four is a sensible number of structure parameters.

As mentioned above, several possibilities were examined. Given the assumption of a resonant vocal tract filter, all-pole modeling via linear prediction was a natural method. Another parameterization used in conventional recognizers that could be adapted to this method were non-weighted (or linear frequency) cepstrum coefficients. Finally, a somewhat ad-hoc, purely time domain set of parameters was conceived.

LPC Parameters

The computation of the LPC coefficients would be entirely routine except for the bandpass nature of the signal in a given channel. Direct fourth order LPC modeling of such a signal would result in coefficients describing primarily the bandpass filter used to isolate the signal for each band.

Prior to LPC analysis the signal has to be modified, preserving the shape of its spectrum in the passband but removing the influence of the filter. This can be done by "stretching" the spectrum of the passband to fill the entire $-\pi$ to π frequency range. The process involves shifting the signal down to baseband and then downsampling.

The shifting operation can be effected by multiplying the time domain signal by an appropriate complex exponential, followed by a lowpass filter preserving only the low frequency "copy" of the spectrum. Alternatively, the DFT coefficients can be re-indexed such that the left edge of the passband falls at zero frequency. A potential problem is posed by the need to fill in the now missing high frequency coefficients. Setting these coefficients to zero is roughly equivalent to frequency-domain filtering.

In practice there were only minor differences between these two methods of spectrum shifting, affecting the LPC coefficients below three significant figures. Since the direct frequency domain shifting method was more expedient it was used in the system development.

The modification of the signal described above could be performed for each individual frame and probably would be in a real-time implementation. For development purposes it was convenient to combine the bandpass filtering and frequency shifting operations in each channel. Figure 2-1 gives an example of the pre-processing steps leading to the computation of the LPC parameters for a frame. In successive panels it shows the spectrum magnitude of an unfiltered frame, bandpass-filtered in channel 2, shifted to baseband, and expanded to fill the entire available frequency range by downsampling in time domain. The LPC coefficients were calculated from the signal corresponding to the spectrum shown in the bottom panel of Figure 2-1. The energy parameters in this scheme were computed prior to the downsampling.

The fourth order all-pole approximation to the spectrum of the frame was computed using the autocorrelation method and the Levinson-Durbin algorithm [50, 67]. The four coefficients constituted parameters lp_{1-4}.

Cepstrum Coefficients

Davis and Mermelstein [22] compared the performance of a continuous speech recognizer under several parametric representations. Mel-weighted cepstrum coefficients yielded the highest scores. However, as argued above, mel-weighting does not make sense for the band-limited input of the sub-recognizers. On the other hand the cited study found that linear frequency cepstrum coefficients (LFCC) were also measurably superior to LPC coefficients. Therefore, this parameterization was also evaluated for the sub-recognizers.

The same caveat applied here as for the LPC computation: the low time cepstrum of the channel signals would be largely determined by the characteristics of the bandpass filters. Therefore, the same shifting to baseband and downsampling procedure as in the foregoing section was applied to the signal prior to the LFCC calculation.

The downsampled frame $x[n]$ was zero-padded to the next power of 2 length and its DFT, $X[k]$ computed. The cepstrum coefficients lc_{1-4} were then computed according

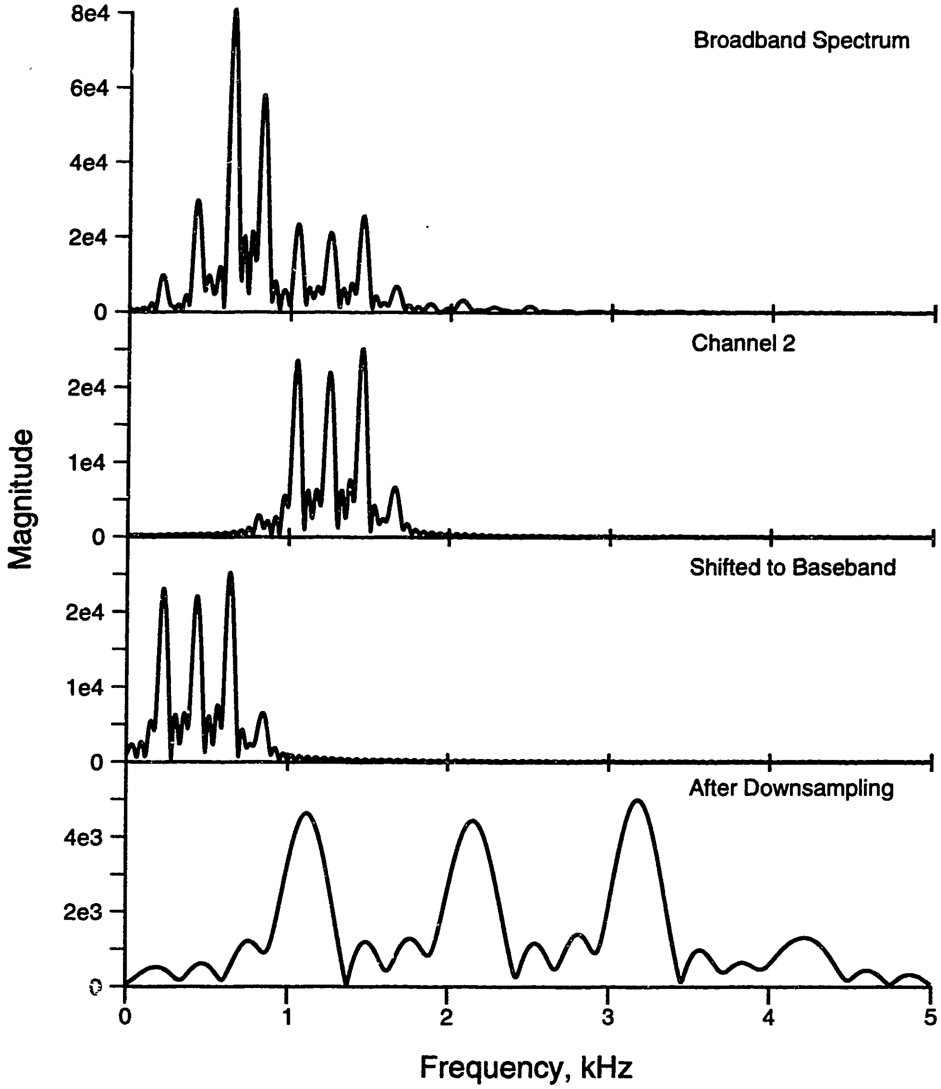

Figure 2-1: Spectra in successive stages of pre-processing of a frame of voiced speech prior to computation of LPC or cepstrum coefficients. See page 30 for more details.

to:

$$lc_i = \sum_{k=0}^{k=\frac{N}{2}-1} \log^2 |X[k]| \cos\left(\frac{\pi i k}{N/2}\right) \qquad (2.3)$$

where N is the length of the padded frame (and the DFT).[3]

Autocorrelation Parameters

The two parameterization schemes described in the foregoing sections are very similar to those commonly used in wideband speech recognizers with modifications that adapt them for band-limited speech. The third proposed parameterization sought to obtain a novel signal representation, specifically geared towards a band-limited signal.

In general, the speech signal in each of the four bands will be shaped largely by one resonance. Representing this resonance is a primary goal of the structure parameters. The two metrics we attempted to characterize are the frequency and the bandwidth (or damping ratio) of the resonance.

One could derive these parameters from the Fourier transform of the band-limited frame. However, in this approach resolution is limited, at least for voiced speech. There the spectrum consists mainly of peaks at integer multiples of the fundamental frequency. If the frequency of the largest of these peaks is taken as the resonant frequency one could incur an error as large as $\frac{F_0}{2}$ where F_0 is the fundamental frequency. For the two lower frequency channels and female speech especially, this would constitute a significant inaccuracy.

To obtain a better estimate more robust spectral estimation could be used. Alternatively the estimation can be performed entirely in the time domain. Aside from avoiding the harmonic problem, time domain parameterization is interesting for other reasons. Formant frequencies are coded quite effectively in the time domain on the cat auditory nerve [77]. Some “auditory based front-ends” for recognition have made use of time domain analysis [29, 78]. Finally, the LPC parameter set already effectively performs a spectral fit of the principal resonance.

[3] The magnitude was left squared since the logarithm operation merely converts the exponent to a constant scaling factor.

One way to estimate the principal frequency component directly from the time waveform in a frame is based on counting zero-crossings [5, 57]. A potentially more robust approach considered here, however, derives parameters from the autocorrelation function of the bandpass filtered speech signal which lends itself to analysis more easily than the raw waveform.

Figure 2-2 shows, as an example, a one frame segment of band-limited speech and the segment's autocorrelation. The original waveform resembles the output of a single resonance driven by a train of impulses. However, the waveform clearly contains other frequency components that might interfere with a direct time-domain analysis. By comparison, the autocorrelation function seems much more dominated by a single frequency component.

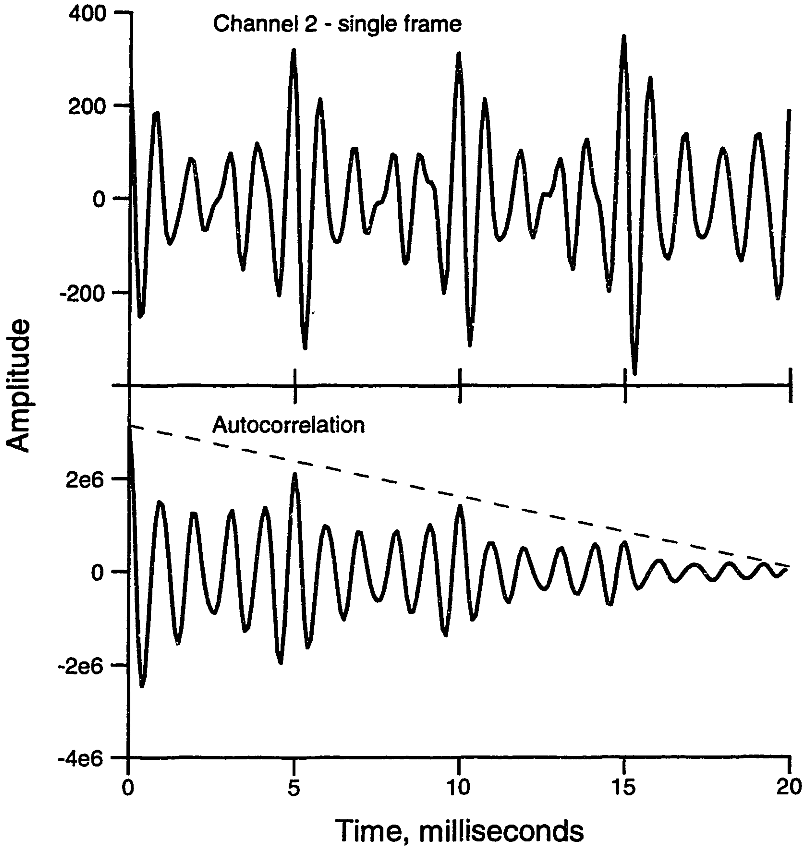

Figure 2-2: Example of a frame of voiced speech in channel 2 and its autocorrelation.

The effect is easily explained, considering that autocorrelation in time domain corresponds to multiplication by the complex conjugate transform in frequency domain. Thus the magnitude of the resulting spectrum is the square of the original, increasing the magnitude difference between large and small frequency components.

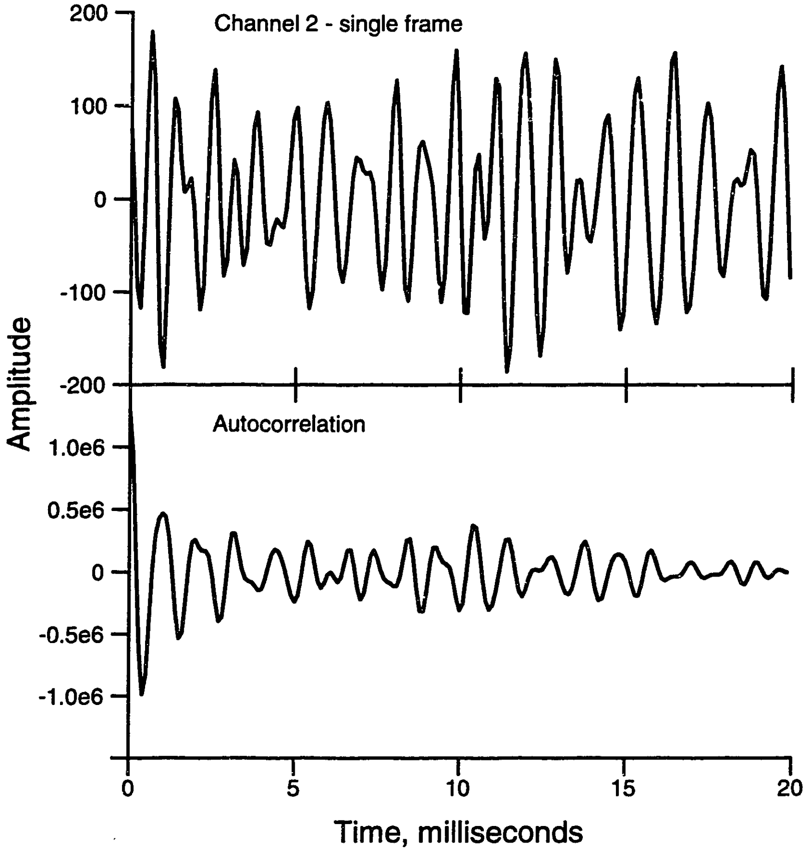

Figure 2-3: Example of a frame of unvoiced speech (/s/) in channel 2 and its autocorrelation.

Alternatively, the use of the autocorrelation is related to matched filtering (for example [62]). There, crosscorrelating a noisy signal with a (usually shorter) waveform of interest helps to detect the latter in the corrupted signal. This view may be especially appropriate in the case of continuant, unvoiced consonants, an example of which is shown in Figure 2-3. The input to the vocal tract in this case is not a train of impulses but rather noise. Consequently the waveform does not exhibit periodicity or a readily apparent damped resonant response. On the other hand, there is a dominant frequency component, determined by the shape of the vocal tract and it is this component that the autocorrelation function accentuates.

Having decided to focus on the autocorrelation function, a somewhat ad-hoc set of parameters was developed. In order to be able to compare directly the performance with this set to performance with the sets previously described, the number of parameters was fixed at four. They were computed as follows:

- The first parameter was designed to measure the principal frequency component in the waveform.[4] First, local maxima were located. Since the processing is done digitally it was judged necessary to obtain more accurate position estimates than by simply picking the largest sample. Specifically, a quadratic was fit to the locally largest sample and the surrounding two samples. The maximum of this parabola was then used as the interpolated maximum.

  Maxima are computed only up to and including the first peak that is larger than the preceding one[5] in order to minimize the interference from peaks due to voicing and fundamental frequency (F0). In general, the location of these peaks will not be related to formant frequencies (see Figure 2-2). This restriction also confines analysis to the part of the autocorrelation derived from significant overlap of the frame. We thus obtain a set of K peaks, indexed from 0 to $K-1$ and $K-1$ interpeak distances τ_i indexed from 1 to $K-1$. Then the first structure parameter, r_1, is given by:

  $$r_1 = \left[\frac{\sum_{i=1}^{K-1} \tau_i}{K-1} \right]^{-1}$$

  that is, it is the inverse of the mean of the interpeak distances. In the limiting case of a pure sinewave the interpeak distances will give us the period and the

[4] Here "waveform" refers to the autocorrelation function. The raw band-limited signal was not considered during the computations.

[5] Since the autocorrelation always has its absolute maximum at lag zero, we are guaranteed at least two additional "eligible" maxima.

inverse of their average the frequency. For the more complex signal r_1 will be an estimate of the principal frequency present, roughly analogous to the carrier in a frequency-modulated (FM) signal.

- When resonance is changing rapidly (for instance, during release of stop consonants) r_1 is less likely to be a robust value. Specifically, we would expect the interpeak distances τ to be more variable under those circumstances. To evaluate the quality of the estimate of the main frequency provided by r_1 a measure of τ variability is used as the second parameter. A convenient metric, and the one that was actually employed, is the standard deviation:

$$r_2 = \sqrt{\frac{1}{K-1}\sum_{i=1}^{K-1}(\tau_i - \overline{\tau})^2}$$

where $\overline{\tau}$ is the average interpeak distance.

- In order to convey information about the bandwidth of the resonance, the third parameter is calculated from the amplitudes of the K peaks described above. Specifically, r_3 is given by:

$$r_3 = \frac{1}{K-1}\sum_{i=1}^{K-2}\frac{v_{i-1} - v_i}{v_{i-1}}$$

where v_i is the amplitude of the ith peak. This parameter then is the average fractional decrease in successive autocorrelation peak amplitudes.

For voiced speech, highly damped resonances will exhibit peaks whose amplitude decrease relatively rapidly. The opposite will be true for narrow resonances. r_3 will capture this effect. It is perhaps less well suited to unvoiced speech.

Following the example of r_1 and r_2 it might seem reasonable to include the standard deviation of the peak decreases as the fourth parameter. This was tried. However, the standard deviation appeared to be highly correlated with r_3 and in preliminary experiments did not improve recognition performance.

Therefore it was not used.

- The final parameter was designed to address the strength of voicing within the scope of the current frame. As seen in Figures 2-2 and 2-3, when speech is voiced there are large peaks in the autocorrelation at multiples of the pitch period. Detecting these peaks has been long used as a method of estimating fundamental frequency [67].

  It was assumed that the large peak in the autocorrelation function would occur at a lag larger than about 3 ms, corresponding to a pitch frequency of 333 Hz, close to the highest normally observed. The magnitude of this peak relative to the zero-lag autocorrelation may be one measure of the degree to which glottal pulses are present. Following this reasoning, the last parameter, r_4, was computed as:

$$r_4 = \frac{ac[0] - v_P \frac{T}{T - \tau_P}}{ac[0]}$$

  where v_P and τ_P are the "voicing peak" amplitude and location. This peak was defined as the largest local maximum located at a lag greater than 3 ms.[6] T is the length of the frame and $ac[0]$ is the magnitude of the autocorrelation at zero lag. The scaling of v_P was performed to compensate for the declining amplitude of the autocorrelation due simply to smaller portions of the signal overlapping. The dashed line in the bottom panel of Figure 2-2 is the autocorrelation of a rectangular window of 20 ms length, normalized to the peak amplitude of the speech frame autocorrelation. This window autocorrelation function is the upper bound on the amplitude of the signal autocorrelation and illustrates the declining amplitude effect.[7] Without the amplitude compensation on v_P, r_4 would be influenced by the frequency of the fundamental: all else being equal,

[6] Quadratic fitting was used to obtain a better estimate of these quantities, just as described for r_1.

[7] Since each frame was Hamming-windowed prior to parameter computation, the "linear" compensation used was admittedly approximate.

higher F_0 (i.e., lower pitch period), would produce a larger r_4. This parameter would then behave differently for male and female voices, an undesirable feature in a speaker-independent recognizer.

Figure 2-4 shows an example parameterization of a part of a TIMIT sentence using the autocorrelation approach. Note, for example, the low values of parameters r_{2-4} during the phones "r" and "ao", indicating a steady, high-Q, voiced region. Parameter r_1 serves as an indicator that two different phones are present, remaining steady through the "r" and then increasing into the "ao". The transition into "ow" is marked by sudden jumps in parameters r_{2-4}. Similar behavior is seen when transitioning into the "n" although r_4 remains relatively low indicating a continuing presence of a voicing peak. The picture is less clear perhaps at the beginning of the utterance where a number of short phones occur. The initial silence is well marked by low energy (e_2) and high r_{2-4}. The phone "d" is also indicated by a spike in e_1 and r_2 and r_3, corresponding to the plosive burst.

2.2.3 Delta Parameters

A major shortcoming of the parameters described above is the limited scope of the signal that they describe. Specifically, they convey no information about the signal surrounding the current frame. Given the dynamic nature of speech it is not surprising that inclusion of parameters that depend on the signal outside of the current frame improves performance [28, 42, 41]. In general, these so-called delta parameters seek to describe the time evolution of the static parameters, i.e. the ones dependent solely on the current frame.

In the present context the potential utility of delta parameters is especially apparent. Each of the channels is dominated roughly by a single resonance of the vocal tract. It is well known that the temporal evolution of these resonances carries significant information about the identity of the underlying phones. For instance, the place of articulation of stop consonants, weak fricatives, and nasals can often be identified by the spectral transitions before and after the phone proper [59, 23].

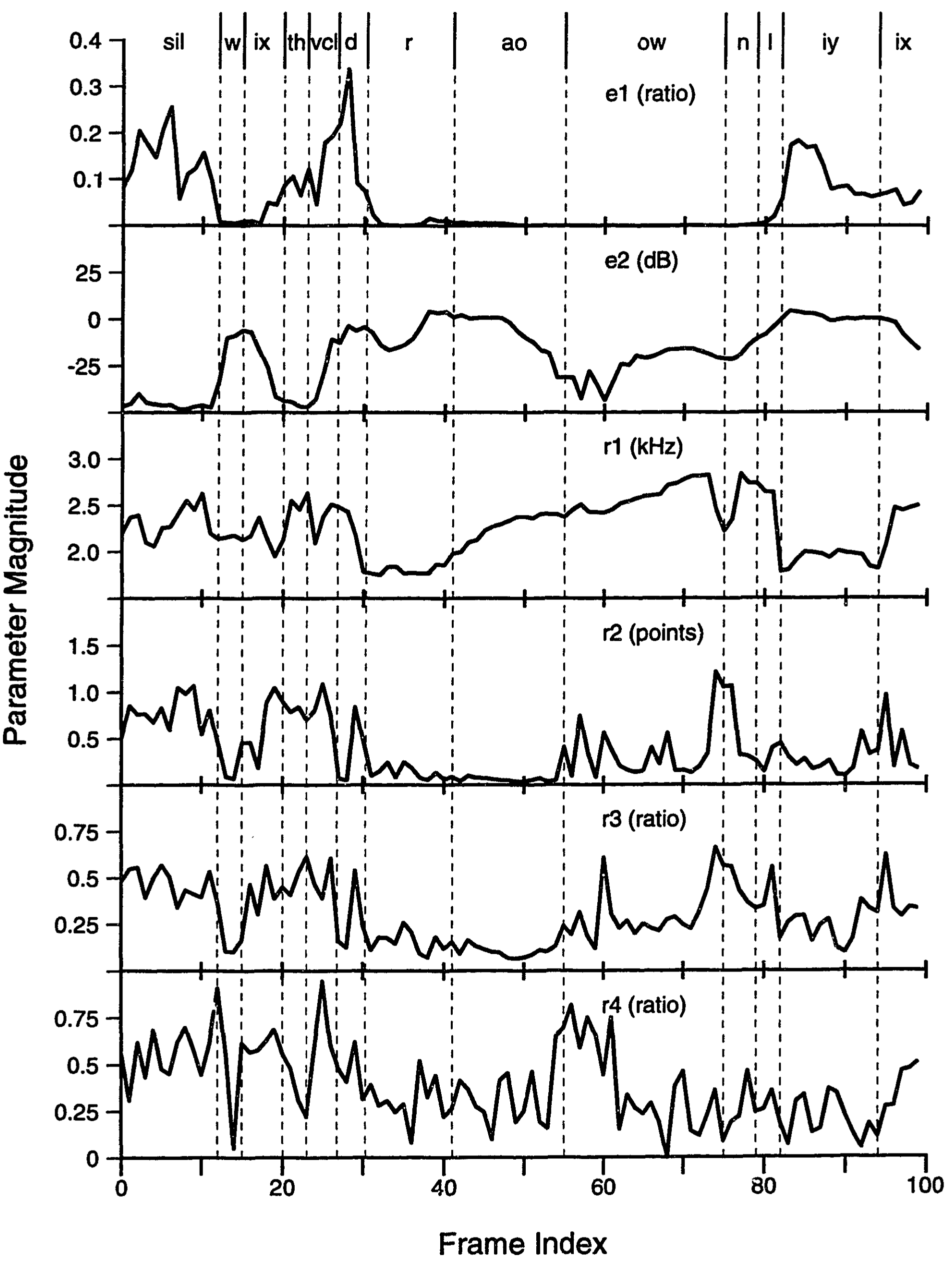

Figure 2-4: Example of autocorrelation parameterization of the utterance "withdraw only a(s)". Channel 3. Labels indicate parameter and its units; "ratio" signifies a unit-less parameter. See text for more details.

A number of methods of computing the dynamic parameters have been proposed and still more could be envisioned. However, it does not appear that the more complex approaches have yielded a consistent and significant improvement [32]. Therefore it was decided to employ a relatively simple measure. A delta parameter for frame ν, $\Delta p[\nu]$ is calculated from the corresponding sequence of static parameters $p[n]$ as:

$$\Delta p[\nu] = p[\nu + \delta] - p[\nu - \delta]$$

For all experiments in this study $\delta = 2$.

Chapter 3

Parameter Grouping and Quantization

The parameter vectors described in the previous chapter constitute the acoustic evidence available to the HMM-based linguistic decoder during recognition. In each channel a vector is generated every 10 ms. The recognition process centers on finding the probability that a spoken sequence of phones gives rise to the observed sequence of these vectors. To make them more amenable to a probabilistic characterization, further modeling or processing of the vectors is necessary.

Discrete parameter recognition systems require that the acoustic evidence be presented as a sequence of symbols drawn from a finite alphabet. Continuously-valued parameter vectors, therefore, must be converted by being mapped to the closest prototype vector taken from a finite set, according to some quantitative distance metric. This process, known as vector quantization, converts the sequence of parameter vectors to a sequence of symbols corresponding to the appropriate prototypes.

Training procedures estimate the probability mass function of observing these discrete symbols on each transition of the Markov model of a speech unit. This allows non-parametric modeling of arbitrary distributions of the speech data vectors. However, the quantization process entails a possibly severe loss of information since distinct vectors are represented by the same prototype and are thus indistinguishable as far as the linguistic decoder is concerned. The distortion may be alleviated simply by increasing the number of prototype vectors (i.e., the size of the codebook). Unfortunately, this proportionately increases the number of HMM parameters that have to be estimated. For finite training data, reliable estimation becomes difficult and performance suffers.

Continuous parameter recognition avoids this problem by assigning the vectors a multivariate probability distribution [70]. During training the parameters of these distributions are estimated. Since no quantization takes place this method may better preserve the acoustic evidence. The cost paid is two-fold. First, a distribution has to be forced on the parameter vectors. For reasons of tractability Gaussian distributions are preferred. These, however, have been found too constraining. For instance, they can only model unimodal behavior. Mixtures of Gaussians allow more flexibility but increase both the number of parameters that need to be estimated and the amount of computation during recognition. Tied mixture modeling (also known as semi-continuous modeling) compromises by sharing a relatively small number of distributions among all the models. This last approach appears to have been most successful [9, 34, 64].

Even tied mixture modeling increases computation substantially. On the other hand the dimensionality of the parameter vectors in the sub-recognizers of the proposed multiband system can be made small by design. This lessens the expected VQ distortion while keeping the number of prototypes small enough so that probabilities can be estimated robustly. The computational savings over continuous methods are important for the motivating real-time application. Following this reasoning, discrete parameter recognition was used.

3.1 VQ Codebook Generation

In order to perform vector quantization an appropriate codebook of prototype vectors must be constructed. The goal of this step is to obtain a set of vectors most representative of the expected speech data. In practice this amounts to minimizing the expected distance of input speech vectors to the closest prototype in the codebook.

A number of distance measures have been applied in similar contexts in speech processing. However, no clear winner has emerged for HMM applications. Therefore, for simplicity of computation, the Euclidian distance was used. The distance of a vector being quantized $\vec{x}_n$ from a prototype vector $\vec{r}$ was thus given by $||\vec{r} - \vec{x}_n||^{\frac{1}{2}}$.

The algorithm of Linde, Buzo and Gray [46] was used to compute the prototype vectors. Appendix A outlines the full procedure as implemented for this work. The *k-means* algorithm [49] was also evaluated but the differences in performance, as measured by average expected distance of vectors from prototypes, were insignificant.

Both algorithms require training data from which to compute the prototypes. These data should be representative of the speech that the vector quantizer will process but do not have to be identified or marked in any way. In this work 390 sentences from the training portion of the TIMIT database (see Sec. 1.4), representing proportionately the eight dialect regions, were used to generate all codebooks. For a codebook of depth 128 these data gave on the average 880 training vectors per prototype.

3.2 Quantization

Once a codebook is available, the process of vector quantization is simple. Each of the parameter vectors $\vec{x}_n$ is assigned the label μ of the closest prototype vector $\vec{r}_\mu$:

$$\mu = \arg \min_{1 \leq m \leq M} [||\vec{r}_m - \vec{x}_n||^{\frac{1}{2}}]$$

where M is the number of prototype vectors in the codebook.

Application of this formula converts the sequence of continuously-valued parameter vectors to a sequence of discrete labels. The latter constitutes the input to the HMM recognizer.

As mentioned above, the optimal number of prototype vectors results from a compromise between accuracy of representation of the acoustic waveform and robustness of the estimates of HMM parameters. Commonly used codebooks contain several hundred prototypes [1, 42, 69]. Rabiner *et al.* [69] reported that increasing the number of prototypes beyond 256 gave little improvement. The multiband system employs comparatively low-dimensional parameter vectors and consequently would be expected to require smaller codebooks for optimum performance. During the initial stages of development, codebooks of size 32 to 1024 were compared. In general it was found that increasing codebook size past 128 yielded very small improvements in individual channel recognition rate. These improvements were judged insufficient to justify the longer processing times and increased memory requirements associated with larger codebooks. Consequently, codebook size of 128 was used for all subsequent experiments.

The parameters that were included in the same vector were often of very disparate scales. This was especially true of the autocorrelation parameters; for instance, r_1 (frequency in Hertz), and r_2 (standard deviation of interpeak distances, in points). In order to obtain a meaningful quantization, parameters in a given vector had to be normalized. In this case this was accomplished by scaling each parameter so that the variance of its distribution was 100,[1] similar to the approach taken in [41]. The variances were estimated by collecting histograms of each parameter from the SX sentences in the TRAIN portion of TIMIT.

3.3 Multiple Codebooks

In the most straightforward application of Vector Quantization in an HMM recognizer, all parameters extracted from a given speech frame are combined in one vector. Quantization of this vector based on an appropriate codebook yields the label of the closest prototype. This label is then used as the acoustic evidence by the recognition algorithm.

Gupta *et al.* [31] first proposed the use of multiple codebooks as a means of obtaining a more faithful representation of the parameter vectors. Instead of combining all parameters in a single vector, they are divided among two or more vectors. For

[1] The value 100 was naturally arbitrary.

each of these parameter sets a separate codebook is created. When processing a frame each parameter "subvector" is quantized separately, using the corresponding codebook. Consequently, two or more labels per frame constitute the input to the linguistic decoder.

The principal advantage of the multiple codebook scheme comes from the lower dimensionality of the subvectors. As a result, *for the same number of prototype vectors* one incurs a smaller distortion in the quantization process, in turn better preserving the acoustic evidence for the recognizer. The implications for the HMM processor are discussed in Section 4.4.1. Using multiple codebooks requires making certain assumptions about the data being quantized. Performance will not improve if these assumptions are violated.

3.4 Parameter Grouping

The preceding chapter described the three parameter classes under consideration in the front end of each channel recognizer: energy, signal structure, and delta. Furthermore, three different structure parameter sets were proposed. It remained to be decided what combination of these parameters was optimal for narrowband recognition. In addition, division of the parameters among several codebooks was considered. Wideband recognition efforts have generally found that larger parameter vectors can be beneficial as long as there is sufficient training data [42, 72]. With that in mind, three general, progressively more complex groupings of parameters were considered.

The static parameters describing the energy content and signal structure (Sec. 2.2.1 and 2.2.2) were always included. The first parameterization option then comprised six parameters. Since three different structure sets were available three different static sets could be tested; they would share the two energy parameters but differ in the structure parameters.

The acoustic evidence can then be augmented with the addition of delta parameters described in Section 2.2.3. Potentially there exists one delta parameter for every static parameter, yielding a total of twelve parameters per frame. This is a fairly

Structure	static only	static + delta	
Parameters	1 codebook	2 codebooks	3 codebooks
LPC	e_{1-2}, lp_{1-4}	e_{1-2}, lp_{1-4} Δe_{1-2}, Δlp_{1-4}	e_{1-2}, Δe_{1-2} lp_{1-4} Δlp_{1-4}
Cepstrum	e_{1-2}, lc_{1-4}	e_{1-2}, lc_{1-4} Δe_{1-2}, Δlc_{1-4}	e_{1-2}, Δe_{1-2} lc_{1-4} Δlc_{1-4}
Autocorrelation	e_{1-2}, r_{1-4}	e_{1-2}, r_{1-4} Δe_{1-2}, Δr_{1-4}	e_{1-2}, Δe_{1-2} r_{1-4} Δr_{1-4}

Table 3.1: Parameter groupings for quantization.

large number to combine and quantize as one vector, comparable to dimensions of vectors in wideband recognizers. Since static and dynamic parameters are only weakly correlated (mostly at the extreme values) they were split into separate vectors with separate codebooks. Preliminary experiments confirmed that performance with two six-element vectors was superior to that with one twelve-element vector.

The third refinement of the parameter grouping attempts to further take advantage of the multiple codebook approach. The same twelve parameters (static + delta) are now divided into three vectors. As with any application of the multiple codebook scheme, care must be taken that separate vectors (in the same frame) be as uncorrelated as possible.

Since three different structure parameters are available, each of the above parameter groupings translates into three different parameterization schemes. Table 3.1 summarizes the parameter arrangements that were further considered. Section 4.4.1 describes the specific parameter arrangement, i.e. distribution among codebooks, that was chosen.

Chapter 4

HMM Sub-Recognizers

The last two chapters described the processing that occurs in each channel to derive the final acoustic evidence provided to the HMM sub-recognizers. This chapter will focus on the HMM recognition engine that was applied to this evidence in each channel to render four intermediate phone estimate streams.

4.1 Alphabet

The first task in specifying a recognition procedure is the establishment of an alphabet: the set of speech elements from which an utterance will be composed. In the present study the motivating application immediately suggests the use of phones.[1]

The exact set of elements was largely determined by the database on which the experiments would be conducted, namely TIMIT (Sec. 1.4). The phonetic transcription of this database uses 64 symbols. These were collapsed to a set of 48, identical to the set used by Lee and Hon [43] and very similar to sets popularly used in speech recognition [48]. This reduction mainly merges groups of closures. For instance, closures preceding /b/, /d/, and /g/, counted as separate elements in the full alphabet,

[1] Strictly speaking, the motivating application, i.e. the cueing protocol, calls for identification in terms of phonemes. However, the same phoneme can have quite distinct acoustic realizations (allophones). Consider, for instance, the phoneme /t/ in "stop" versus that in "butter". On the other hand it is usually straightforward to map phones to phonemes. It was decided, therefore, to use phones as basic recognition units.

Phone	Example	Merged	Phone	Example	Merged
iy	beet		en	button	
ih	bit		ng	sing	eng
eh	bet		ch	choke	
ae	bat		jh	joke	
ix	debit		dh	then	
ax	about		dx	muddy	
ah	but		b	bee	
uw	boot	ux	d	day	
uh	book		g	gay	
ao	bought		p	pea	
aa	cot		t	tea	
ey	bait		k	key	
ay	bite		z	zone	
oy	boy		zh	usual	
aw	bout		v	van	
ow	boat		f	fin	
l	lay		th	thin	
el	bottle		s	sea	
r	ray		sh	she	
y	yacht		hh	hay	hv
w	way		cl	*unvoiced closure*	pcl,tcl,kcl,qcl
er	bird	axr	vcl	*voiced closure*	bcl,dcl,gcl
m	mom	em	epi	*epinthetic silence*	
n	noon	nx	sil	*silence*	h#,#h,pau

Table 4.1: List of phones in the alphabet used throughout this study. The *Merged* labels designate elements distinguished by TIMIT that were treated as equivalent to the given phone.

are collapsed into the voiced closure, /vcl/. Several fine distinctions among the nasals and a few other phones were also merged. The glottal stop "q" was eliminated from the transcriptions entirely. Table 4.1 lists the phones used.

4.2 HMM Topology

In their most general form, Hidden Markov Models may represent a sequence of speech (or other) events with so-called fully connected networks. In such a model, transitions are allowed between each state and any other state. Timing constraints in speech suggest a more structured topology [69]. For instance, since speech events are not by nature cyclical, feedback connections between states are not required.

Several HMM topologies appropriate for phone models have been proposed [2, 7, 42, 76]. The present system uses the simplest form of a *Bakis* model shown in Figure 4-1, based on that suggested in [7]. This model consists of three states meant to roughly represent, in order, the initial transition into the phone, the steady state interval, and the transition out of the phone. The arrows indicate possible transitions; absence of a connection between two states indicates that that particular transition is not allowed. The dashed arrow represents a *null* transition, explained below. The model also includes the "entry" and "exit" dummy states. These do not model the phone as such but rather are used to provide connections between successive models as explained below.

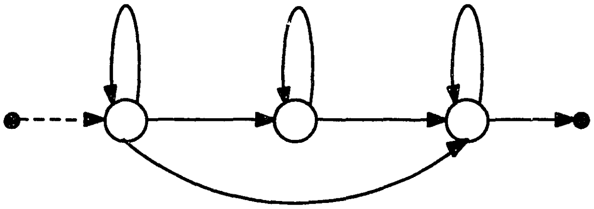

Figure 4-1: Markov model of a single phone.

As with any discrete HMM model two sets of parameters have to be specified[2]:

- the transition probabilities a_{ij}^m - the probability of moving to state j given that the current state is i and we are within the model for phone m.

- the observation probabilities $b_{ij}^m(k)$ - the probability of observing the symbol k when making the transition $i \rightarrow j$ within the model m.

[2] For a general HMM one would also need the initial state probabilities, often denoted π_i, indicating the likelihood of entering a model at each state. Given the structure of the models, all initial probabilities are effectively set to zero, except for the leftmost state where it is identically one.

In the structure chosen here each transition was assigned a different observation probability distribution. In other words, there was no tying effected among the transitions.[3] In our case the training data was sufficient because there were relatively few phone models and transitions within the models, and a small number of possible observation symbols (128, c.f.Sec. 3.2), relative to the size of the training database.

The individual phone models can be concatenated to yield a language model that may be fit to any continuous utterance. This is illustrated in Figure 4-2. The 48 available phone models appear in parallel. Their final transitions all connect to an "exit" dummy state. A null transition, which does not emit an observation symbol and takes no time, connects back to the "entry" dummy state. From this state in turn, null transitions allow transfer to any phone model.[4] Clearly, this arrangement can describe utterances consisting of an arbitrary number of phones.

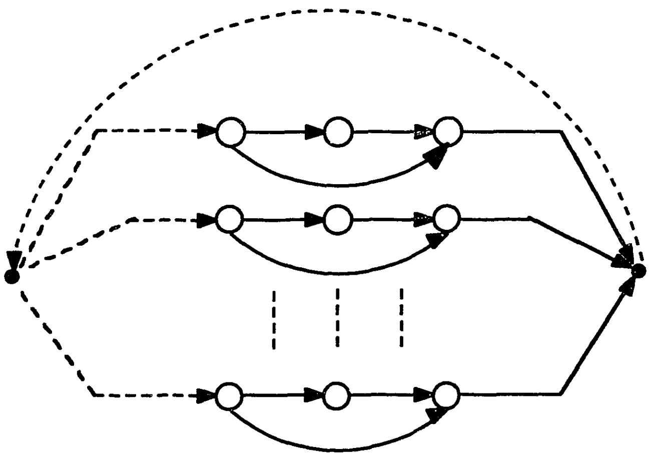

Figure 4-2: Connections of individual phone models to form the language network. Self-transitions omitted in the figure.

[3] Tying, where different transitions are constrained to have the same observation distributions, is used to produce more robust estimates when training data is scarce compared to the number of b parameters.

[4] The successive null transitions are, strictly speaking, redundant. The exit state could be connected directly to the first state of each phone model. Computationally this is completely equivalent. The picture given here was chosen for symmetry and clarity.

Since phone models are concatenated via null transitions, no observation probability distributions need to be trained for these transitions. However, one still needs to specify the transition probabilities (the a_{ij}'s) for the entry state. The structure of Figure 4-2 admits two general approaches: the zero-gram and the uni-gram language models. In the former, the transition probabilities are simply $\frac{1}{M}$, where M is the number of phone models, i.e. all phones are assumed to be equally likely. The uni-gram model, in contrast, weights the probabilities of entering successive phone models according to an estimate of the frequency with which the phones occur in speech. As would be expected, the latter approach can result in improved accuracy [42, 43, 63].

However, the output of a sub-recognizer is not the final result desired. The outputs from the four channels need to be combined. Including the frequency-weighted transitions biases the output towards the more often encountered phones. This makes the recovery of less frequent phones difficult during the decision integration process. The situation is aggravated by the relatively poor matches between the acoustic evidence and the phone models in the band-limited sub-recognizers. Under such conditions the output of a sub-recognizer is unduly dominated by the intra-phone transition probabilities.

In preliminary experiments it was found that the uni-gram model produced scores slightly higher than the zero-gram model, but at the cost of a large number of deletions (see Sec. 6.1 for description of error types). Evidently, in the absence of strong matches between the models and the data, the sub-recognizers were "reluctant" to leave the favored phones. The high deletion rate also makes it more difficult to integrate the individual channel decisions. On the other hand, as will be seen, the decision integration procedure can readily incorporate the a priori probabilities and is perhaps the more natural place to do so. Consequently, sub-recognizers used the zero-gram model exclusively.

4.2.1 Model Training

Training refers to the process of estimating the model parameters based on labeled acoustic data. The training method employed here was the well known Baum-Welch or forward-backward algorithm [6, 38, 42, 69]. This is a two-step, so-called expectation-maximization procedure. In the first step of each iteration, the current set of HMM parameters (i.e., the a's and b's) $\mathcal{D}$ is used to compute the expected likelihood of each arc in the model being taken and each observation symbol being emitted *at each time tick*, given the training data $\vec{Y}$. $\vec{Y}$ represents the sequence of (in our case quantized) speech parameter vectors. In the maximize step these expected values are used to recompute the HMM parameters, resulting in a new set $\hat{\mathcal{D}}$. At the next iteration $\hat{\mathcal{D}}$ is used as the initial parameter set. The procedure, which is similar to a gradient hill climb, is guaranteed to increase the likelihood of observing the training data given the model up to a local maximum, i.e.:

$$\Pr(\vec{Y} \mid \hat{\mathcal{D}}) \geq \Pr(\vec{Y} \mid \mathcal{D})$$

If the local maximum coincides with the global maximum, $\hat{\mathcal{D}}$ converges to the maximum likelihood estimate of the HMM parameters.

In the system implemented here, the reestimation procedure was initiated with a *flat start*, i.e. by setting all probabilities equal at the start, subject to the constraint of summing to one. While more sophisticated initialization procedures have been proposed, the flat start has been found to work well for continuous speech tasks [63].

The Estimate-Maximize steps are repeated until some criterion of convergence is met, or a fixed number of iterations may be run. The stopping condition used here considered the average increase in the conditional probability of producing a sentence (or more accurately, the sequence of quantized vectors $\vec{Y}$, representing the sentence) in the training corpus. The training was stopped when the log of this improvement reached 5. In other words the model $\hat{\mathcal{D}}$ was accepted as final when:

$$\frac{1}{N}\sum_{i=1}^{N}[\log \Pr(\vec{Y_i} \mid \hat{\mathcal{D}}) - \log \Pr(\vec{Y_i} \mid \mathcal{D})] \leq 5$$

where the summation index i covers all the N sentences in the training set.

In practice this resulted in roughly six iterations of the forward-backward algorithm. Increasing the number of iterations had a negligible effect on the recognition rate.

Discrete HMM systems have to contend with the *unseen observation* problem. A label that never occurs during training within the scope of a particular phone model will have its observation probability set to zero. Should it occur during recognition it will eliminate that phone from contention even if the rest of the label sequence matches the phone well. This has been found to degrade performance. Since the context-independent models used here were, in general, well trained, the simple solution of setting the unseen probabilities to a minimum non-zero value was utilized. This value was picked as 10^{-5} following the data in [69].

Given a phonetically labelled and manually segmented database such as TIMIT, one could train the individual phone models on their acoustic realizations excised from the sentences. An alternate strategy, which was used in the current system, is to fit the appropriate sequence of phone models to each entire sentence. The phone model sequence is made to match the known phonetic transcription but the location of phone boundaries available from the manual segmentation is ignored. This allows the training procedure to converge to its own phone boundaries. For continuous speech recognition the latter approach is superior because it allows phone models, especially their transition states, to capture some coarticulatory influences by including information in the transition regions. Although not directly relevant here, this technique also allows the training to proceed on phonetically labelled but unsegmented databases which alleviates the effort of preparing the training material.

4.2.2 Recognition

During the recognition phase, we want to identify the sequence of speech elements (phones) most likely to have produced the observation sequence $\vec{Y}$ derived from the acoustic waveform of the the unknown utterance. For the particular task here $\vec{Y}$ corresponds to the parameter vector labels calculated over a sentence. The Viterbi algorithm [69, 83] was used to find the corresponding phone sequence, This decoder finds the path q through the state network of Figure 4-2 that maximizes $\Pr(\vec{Y} \mid q, \mathcal{D})$ where $\mathcal{D}$ is the set of Markov model parameters we have trained. It does so in a time-synchronous fashion, finding at time tick t for every state in the network the most likely predecessor, given the predecessor's score at time $t-1$ and the transition to the current state. At the end of the utterance the best state sequence is recovered starting at the state with the final highest score.

The structure of TIMIT made it natural to perform recognition one sentence at a time. In other words, the Viterbi algorithm would align the most likely path through the state network to an entire sentence. In recognizing running speech the algorithm would be applied to the last few seconds of the acoustic input and the identity of the current phone reported according to the final state of the Viterbi alignment. Some, probably heuristic, post-processing might be required, for example to deal with an alignment whose final state is in the middle of a phone model.

A real-time implementation might also make use of a still more efficient variation of the Viterbi search. In this method, called the *beam search* [76], only the states with sufficiently high current scores are considered as potential predecessors at each time tick. The minimum eligible score is given as a constant offset below the maximum probability achieved by any state at the given time. The beam search reportedly can reduce the number of possible paths to be explored by up to two orders of magnitude without seriously degrading performance. However, since recognition took place off-line in this study, full Viterbi search was used.

4.3 Scoring

The output of a phonetic recognizer is generally evaluated by the number of phones identified correctly. A dynamic programming algorithm is used to match the output phone sequence to the manually produced one [43]. However, the relatively low recognition rate at the sub-recognizer level made this alignment problematic. Instead, since the test database is segmented, and the Viterbi algorithm assigns each data frame to a state in a phone model the sub-recognizer performance was scored as the percentage of *frames* identified correctly.

In computing the scores the true label of each frame in the test sentence was compared to the corresponding label in the sub-recognizer output and the two frames on either side of that frame. The frame was considered to be correctly classified if any of these output frames were labelled the same. This was done in order not to penalize slight differences in segmentation and number of frames assigned to each phone between the manual and sub-recognizer transcriptions. It should be kept in mind that the frame-oriented method of scoring gives greater weight to the recognition of longer phones. Therefore, the scores are mostly relevant as a relative rather than an absolute measure of the sub-recognizers' performance.

4.4 Experiments

A myriad of variations on the basic sub-recognizer can potentially be tested. These include the type and number of parameters to be extracted from the acoustic waveform, the number of codebooks in the VQ stage, the number of VQ prototypes, the arrangement of states in the phone HMMs, number of iterations of the training algorithm, etc. It makes little sense to study all the combinations. As has been indicated in the foregoing sections, at each stage of development many of these alternatives were eliminated from further consideration by performing experiments with the other characteristics held fixed.

To recapitulate, the following were the salient characteristics of the recognition process in each channel:

- All codebooks contained 128 prototypes, providing relatively well trained phone models. Performance increased only incrementally for larger codebooks.

- The HMM topology was fixed as shown in Figures 4-1 and 4-2.

- The forward-backward algorithm was used for training and the Viterbi algorithm for recognition (c.f. Sec. 4.2.1 and 4.2.2).

- The SX sentences of the training portion TIMIT database were used as training data. Including the SI sentences did not improve performance, suggesting that the complexity of the models was not excessive.

Whereas it was not likely that the narrowband system would require, for instance, a different training algorithm, it might respond differently to the type of the parameter set, the number of parameters and the number of codebooks. Experiments at the sub-recognizer level concentrated on these system characteristics.

4.4.1 Parameter Arrangement

As explained in Section 3.4, once the method of finding the signal structure parameters was chosen, we had twelve potential parameters per frame from which to choose. The LPC parameters were picked as the structure set to test three alternative arrangements of these parameters in all four channels.

The three arrangements were:

- *static only:* only the six static parameters: e_{1-2} and lp_{1-4} quantized with one codebook.
- *full - 2 codebook:* six static: e_{1-2} and lp_{1-4}, and six delta: Δe_{1-2} and Δlp_{1-4} parameters quantized with separate codebooks.
- *full - 3 codebook:* all twelve parameters split into three four-element vectors and quantized from three codebooks as follows:
  - energy: static and delta, e_{1-2} and Δe_{1-2}
  - structure: static, lp_{1-4}
  - structure: delta, Δlp_{1-4}

Table 4.2 summarizes the results obtained. They are consistent with previous experience with discrete HMM recognizers. For all four channels the augmentation of the parameter set with dynamic parameters improves recognition, on average, by

Parameter	Channel Score			
Arrangement	1	2	3	4
static only	28.8	30.7	28.6	23.9
full - 2 codebook	35.4	36.1	33.2	27.7
full - 3 codebook	39.8	40.5	37.6	30.8
full - 3 codebook random assignment	35.5	36.1	32.7	27.0

Table 4.2: Percent frames correctly identified for three different parameter arrangements and a random "demonstration" arrangement. LPC Structure parameters. See text for details.

5.1 percentage points. Furthermore, splitting the full parameter set into three vectors rather than two, results in still further improvement of almost 4 percent.

While the full set of experiments represented by Table 4.2 was not repeated for the other two signal structure computations (autocorrelation and cepstrum), selected channels were tested. The trend shown above was not contradicted in any way.

Division of the twelve parameters among the three codebooks was not accidental. The key to the effectiveness of the multiple codebook approach is the assumption that the resulting multiple labels obtained per frame have independent observation probability distributions. Thus, for three labels k^1, k^2, k^3, their joint observation probability for a given model transition $i \rightarrow j$, is:

$$b_{ij}(k^1, k^2, k^3) = \prod_{l=1}^{3} b^l_{ij}(k^l)$$

This assumption allows us to train the probabilities of three quantized vectors as robustly as the probability of one vector. At the same time the VQ distortion of the three vectors is smaller than when all the elements are combined in one vector.

This technique works only insofar as the independence assumption holds. In practice one tries to split the parameters among the subvectors so that there is little correlation among the vectors. The split here was chosen heuristically to minimize such correlation. To demonstrate that correlation can degrade the performance of a multiple codebook system, also in a narrowband recognizer, the three-codebook

Structure Parameters	Channel Score				Average across Channels
	1	2	3	4	
LPC	39.8	40.5	37.6	30.8	37.2
cepstrum	39.3	40.9	37.9	31.4	37.4
autocorrelation	39.7	38.7	35.6	31.5	36.3
Average across Parameters	39.6	40.0	37.0	31.2	

Table 4.3: Percent frames correctly identified for three different waveform structure parameter sets. 3-codebook arrangement as listed on page 57.

system was re-tested, this time assigning parameters to the three codebooks more or less at random. The results are also given in Table 4.2. As can be seen recognition rates drop from those attained by the designed split.

4.4.2 Structure Parameters Performance

The experiments above demonstrated that the best performance, given the overall pattern of the parameterization, was obtained with full (static and delta) parameter sets split into three codebooks. The three proposed types of structure parameters could be employed in this context: LPC, cepstrum, and autocorrelation. The results are given in Table 4.3. Note that the results for LPC are repeated from Table 4.2 to facilitate a comparison.

The cepstrum parameters tend to result in generally the highest recognition rate, being best for channels 2 and 3, and second for channel 4, and having the highest average score. The ad-hoc autocorrelation parameters tend to have lowest performance. The differences, however, are fairly small.

4.5 Conclusion

Increasing the number of parameters per frame and dividing them into separate codebooks was found to yield consistently and significantly superior results at the sub-recognizer level. Subsequent experimentation concentrated, therefore, on the 3-

codebook parameter grouping, including all three classes of parameters: energy, signal structure, and delta.

The differences among individual channels resulting from different structure parameters were more equivocal. In particular, different parameters proved best in different channels. Also the scores were relatively close. Therefore, the combination of channels had to take all three signal structure parameterizations into account.

When compared to human performance on analogous narrowband stimuli, the channel scores appeared high enough to continue with the problem of label integration without altering other aspects of the sub-recognizers, such as topology or the vector quantizer algorithm.

Chapter 5

Decision Integration

Each sub-recognizer's output is a sequence of phones that best matches the acoustic input according to the pre-trained model. In addition, the timing of the transitions between phones is specified to within the spacing between successive frames, in our case 10 ms. Figure 5-1 shows an example of the output from the four subrecognizers over a space of several phones.

Not unexpectedly the outputs of the channels are not identical and do not always agree with the "true" underlying sequence.[1] On the other hand they tend to agree with the basic series of vowel, closure, stop consonant, fricative, closure. The challenge before the integration procedure is to optimally arbitrate among the individual decisions or possibly to choose a phone not identified by any of the channels.

Several complications are immediately apparent. The channels rarely agree on the timing of the transitions between successive phones. In some cases, as illustrated in Figure 5-1, the differences can be on the order of 50 ms or more. Alone this problem would not be fatal, however it becomes more serious considering that individual channels may add or delete phones. Examples of insertions occur in channels 2 and 3 in the figure.

Phone level decision integration can be seen as involving two issues. First, the procedure must decide on the boundaries of the underlying phones and second, on

[1] Serendipitously the first channel happens to be correct as to the sequence in this case, although even it does not match the exact phone boundaries.

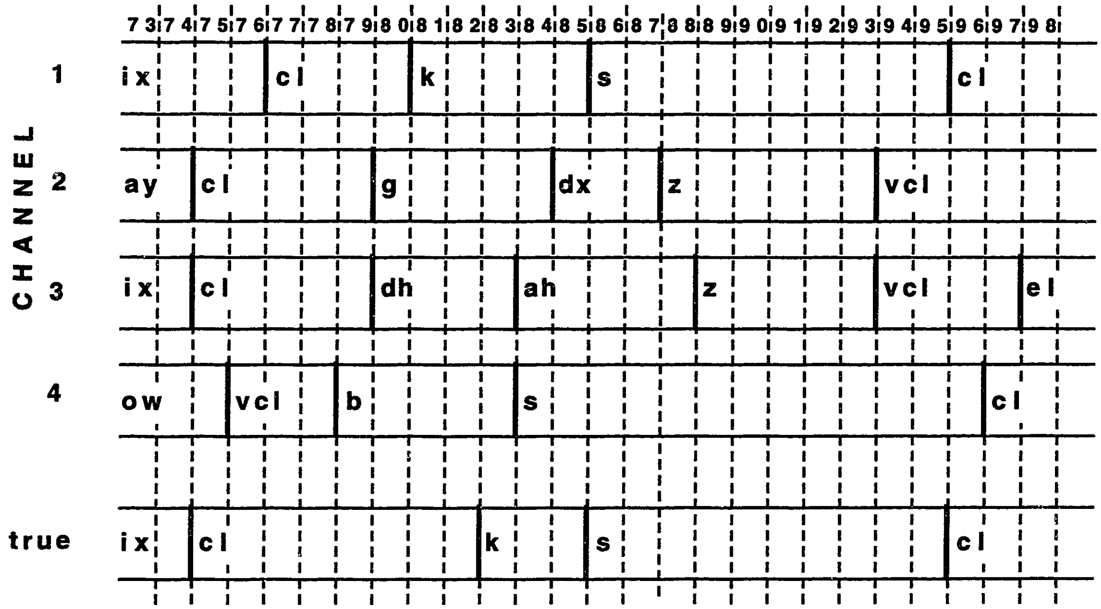

Figure 5-1: Sample output of the four sub-recognizers. LPC parameters, 3 codebooks. Dashed lines indicate frame boundaries. The underlying phone sequence (labelled *true* in the figure) is the initial part of the word "excluded".

their identity. Fundamentally these do not have to be separate processes. However, separating them makes the problem more tractable.

In particular, the problem of combining four labels is readily amenable to statistical approaches if the problem of synchronizing these labels is ignored. This is true if we consider combining the channel decisions at the frame level where channel outputs are perfectly synchronized. Also, there is no such problem as insertion or deletion of a frame.

The main shortcoming of the frame-level integration is that the integrated frame stream is likely to switch frequently among several phones close to the transition between two "true" phones. This is a direct result of inconsistent segmentation by individual channels. Therefore a second stage of the integration process is necessary to provide the final segmentation and to "clean up" the frame stream. The decision integration set-up is depicted in Figure 5-2.

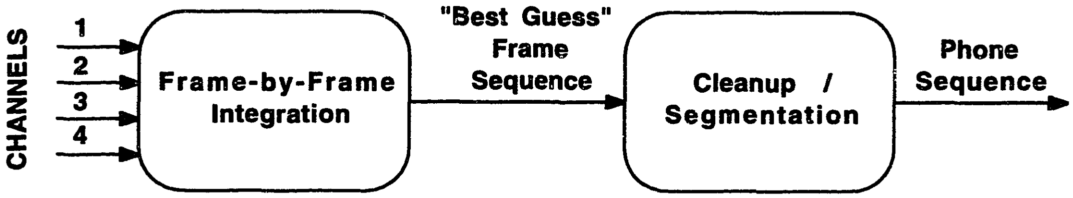

Figure 5-2: Block diagram of the label integration process.

5.1 Frame Level Integration

At the frame level the problem may be viewed as one of probabilistic detection [62]. We have to decide which of a finite number of phones gave rise to the fourtuple of best guesses. The conditions of the problem immediately suggest employing Maximum A Posteriori (MAP) probability detection method. Accordingly, we classify each frame as belonging to phone $\hat{q}$ given by:

$$\hat{q} = \arg \max_{q \in Alphabet} \Pr[q \mid c_1 c_2 c_3 c_4] \tag{5.1}$$

where c_i is the frame classification according to the *i'th* channel.

Equation 5.1 can be rewritten, using the definition of conditional probability and the fact that $\Pr[c_1 c_2 c_3 c_4]$ does not depend on q, as:

$$\hat{q} = \arg \max_{q \in Alphabet} \Pr[c_1 c_2 c_3 c_4 \mid q] \times \Pr[q] \tag{5.2}$$

Here we show explicitly the dependence of the maximum on the a priori probability of q.

As Equation 5.2 indicates, to apply the MAP criterion we need two sets of probabilities: the probability of observing a certain four-fold combination of channel outputs given an underlying phone, and the probability of the phone occurring without regard to the current channel output.

5.2 A priori Probabilities

One way to estimate the a priori probabilities is to assume that all phones are produced with equal frequency. This would effectively remove $\Pr[q]$ from Equation 5.2 and simplify the MAP criterion to the maximum likelihood (ML) criterion.

This assumption is of course unlikely to be correct for any realistic speech corpus [20]. Therefore, in order to maximize the likelihood that $\hat{q}$ is correct a better estimate should be used. The use of statistics reported in [20] was considered. However, the phonetic alphabet used there is not completely compatible with the 48-element alphabet derived from the TIMIT label set as used in this study. On the other hand the training portion of TIMIT provides ample data from which the frequency of phone occurrence could be estimated by simple counting.

There remains an additional issue, however. That is whether $\Pr[q]$ should refer to the frequency of occurrence of phones or phone frames. This is an important distinction. Choosing the frequency of phone frames naturally favors longer duration phones whose frame counts will be higher compared to short phones that may occur just as often. For instance, under this policy fricatives would be likely assigned higher a priori probabilities than stop consonants. This type of tradeoff would not be unreasonable if our goal was to maximize recognition of frames or, equivalently, the fraction of *time* that the recognizer is correct.

For most applications, however, the relevant aspect of recognizer performance is the percentage of *phones* that are correctly recognized. Consequently, the frequency of phone occurrence used as the estimate of $\Pr[q]$ should yield the highest recognition rate at the phone level. This hypothesis was empirically confirmed (see Sec. 6.2). Table 5.1 lists the phone a priori probabilities as estimated from the SX TRAIN sentences of TIMIT.

Phone	Probability	Phone	Probability	Phone	Probability
iy	0.0345	l	0.0380	g	0.0104
ih	0.0261	el	0.0069	p	0.0197
eh	0.0223	r	0.0389	t	0.0247
ae	0.0175	y	0.0088	k	0.0253
ix	0.0468	w	0.0179	z	0.0231
ax	0.0267	er	0.0363	zh	0.0019
ah	0.0148	m	0.0280	v	0.0131
uw	0.0113	n	0.0466	f	0.0183
uh	0.0043	en	0.0039	th	0.0053
ao	0.0168	ng	0.0069	s	0.0402
aa	0.0188	ch	0.0062	sh	0.0085
ey	0.0146	jh	0.0071	hh	0.0084
ay	0.0142	dh	0.0182	cl	0.0844
oy	0.0032	dx	0.0112	vcl	0.0517
aw	0.0042	b	0.0190	epi	0.0072
ow	0.0118	d	0.0152	sil	0.0608

Table 5.1: Phone a priori probabilities.

5.3 Conditional Probabilities

The second, crucial component of Equation 5.2 is $\Pr[c_1c_2c_3c_4 \mid q]$ - the probability of observing the given four-way combination of individual sub-recognizer outputs, given that the frame fell within phone q. The natural method of estimating these probabilities is from analysis of sub-recognizers' outputs on known (i.e. labeled) training data. Specifically, we could estimate the a posteriori probability as:

$$\Pr[c_1c_2c_3c_4 \mid q] = \frac{x^q_{c_1c_2c_3c_4}}{N_q} \tag{5.3}$$

where $x^q_{c_1\ldots c_4}$ is the number of frames the sub-recognizer output combination $c_1c_2c_3c_4$ occurs when phone q is spoken and N_q is the total number of frames of phone q present in the training data.

Unfortunately this direct approach proves impractical for any realistic application. An excessively large database would be required to estimate the probabilities by Equation 5.3. For a 48 phone Alphabet there are 48^4 or well over 5 million possible combinations of individual channel outputs. Furthermore, the conditional a posteriori probability of each combination for each of 48 phones is required (a total of almost 250 million combinations). The training portion of the TIMIT database contains only about 1 million frames.[2] This is clearly insufficient to estimate 250 million probabilities.

The problem is most obvious for a combination that is *never* seen in the training data but occurs in speech to be recognized. In that case Equation 5.2 would be unable to render a decision. The difficulty with the simple method is more general, however. In addition to the combinations that do not occur in the training data, many combinations occur only a few times. In practice these turned out to constitute a large fraction of the combinations seen during recognition. Clearly, in these cases the estimate provided by Equation 5.3 will not be very reliable.

A robust estimate of the a posteriori probabilities is indispensable for the proposed frame-by-frame integration scheme; in fact, it is its key element. Consequently, several methods of estimating these probabilities from the available data were developed and tested. Their theory is presented below. Some of the practical aspects, arising in application are addressed with experimental results in Section 6.4.

5.3.1 Two-at-a-time Method

While it is unlikely that sufficient data could be obtained to estimate the probabilities of the four-fold combinations directly, it is much more realistic to perform such an estimate for any two-channel combinations. There we have 48^2 (2304) possible outputs and a reasonably good coverage with roughly 700,000 frames from the training part of TIMIT SX set. In particular, the problem of unseen combinations does not occur. The coverage for some less frequently occurring phones may be insufficient if we were interested in fine statistical differences. However, the goal is to find the reliable maximum a posteriori probability and for this purpose the training data are sufficient.

The approach this suggests is to first separately two pairs of channels and then

[2] Excluding the SA sentences; these two sentences are spoken by all speakers, using them might skew the statistics. See Sec. 1.4.

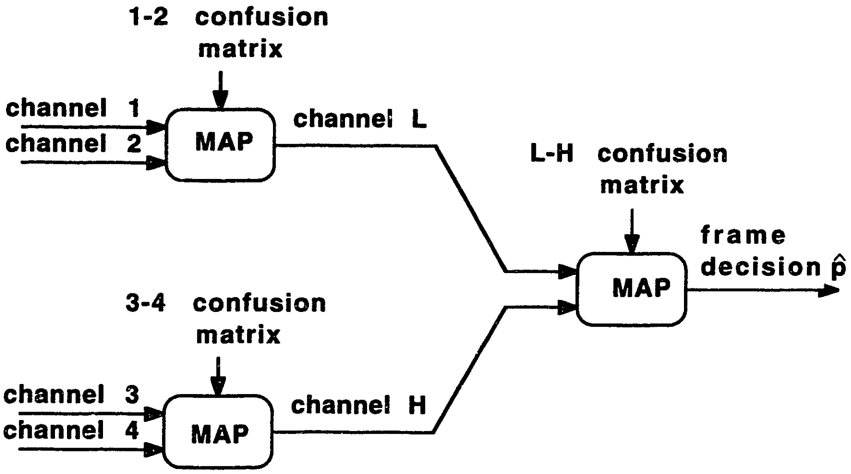

Figure 5-3: Block diagram of the two-at-a-time frame integration method.

to combine the resulting two "compound channels". This concept is illustrated in Figure 5-3. In this implementation MAP detection is applied to the channel pairs 1 and 2, and 3 and 4 separately. The results of the combination of each pair form new streams of frame labels, called channels L and H. It is not a fundamental requirement that adjacent channels be combined first. However, the method requires us to effectively estimate joint probabilities of two channels. These estimates will be better and more useful for channels that are correlated and these in turn are likely to be the adjacent ones.

Channels 1 and 2 are thus integrated according to:

$$\hat{q}_L = \arg \max_{q \in Alphabet} \Pr[c_1 c_2 \mid q] \times \Pr[q] \tag{5.4}$$

Here the a posteriori probability $\Pr[c_1 c_2 \mid q]$ is computed as given by Equation 5.3, modified for only two channels. Channels 3 and 4 are combined in an analogous fashion to produce $\hat{q}_H$.

To obtain the final decision channels L and H are combined according to:

$$\hat{q} = \arg \max_{q \in Alphabet} \Pr[q_L q_H \mid q] \times \Pr[q] \tag{5.5}$$

In this scheme then, the four-fold probability is approximated as:

$$\Pr[c_1 c_2 c_3 c_4 \mid q] = \Pr[\hat{q}_L \hat{q}_H \mid q]$$

5.3.2 Independence Assumption

Another straightforward method of estimating the a posteriori probabilities relies on the assumption that the individual channel decisions are statistically independent of each other. Under this assumption we may write:

$$\Pr[c_1 c_2 c_3 c_4 \mid q] = \prod_{i=1}^{4} \Pr[c_i \mid q] \tag{5.6}$$

Thus the problem is reduced to estimation of the individual channel decision a posteriori probabilities. There is certainly enough data in TIMIT to do this by simply counting the appropriate frames, i.e.:

$$\Pr[c_1 \mid q] = \frac{x^q_{c_1}}{N_q}$$

which is simply Equation 5.3 applied to a single channel.

The independence assumption has several advantages. It is very easy to implement, involving only simple computations. The individual a posteriori probabilities are likely to be very robust, with no danger of unseen combinations. And it corresponds directly to the post-labeling integration model of human perception described in the Introduction.

5.3.3 Log-linear Model Fitting

The previous sections avoid the problem of missing output combinations by assuming complete independence among sub-recognizers or groups of sub-recognizers. One might argue, however, that the training data provides enough information to take advantage of some, albeit reduced level of interaction among the channel outputs. One approach is through log-linear modeling of the data.

The data from which we want to estimate the a posteriori probabilities, $\Pr[c_1c_2c_3c_4 \mid q]$, forms a set of 48 (for the chosen alphabet) four-dimensional *contingency tables.* Each of these tables is conditioned on a different phone q and contains the counts of the number of occurrences of all the possible sub-recognizer output combinations when the particular phone was spoken $x^q_{c_1c_2c_3c_4}$. The sparseness of these tables is the problem.

We may consider the entries of the contingency table as being the result of sampling the available N frames of phone q present in the training data. The probability of a given combination $\{c_1c_2c_3c_4\}$ of outputs is precisely $\Pr[c_1c_2c_3c_4 \mid q]$. Simplifying the notation for the probability to p_{ijkl} and the count in the table cell to x_{ijkl}, the counts have the multinomial probability density function:

$$f(x_{ijkl}) = \frac{N!}{\prod_{i,j,k,l} x_{ijkl}!} \prod_{i,j,k,l} {p_{ijkl}}^{x_{ijkl}} \tag{5.7}$$

The individual combination probability is related to the expected count, m_{ijkl} in the corresponding cell by:

$$p_{ijkl} = \frac{m_{ijkl}}{N}$$

Finding the expected values of the cell counts is therefore equivalent to finding the desired probabilities.

We can obtain estimates of these expected cell counts that are more robust than the raw counts x_{ijkl} by fitting log-linear models [12] to the array of counts. The general, so-called saturated, model for the four-dimensional array is:

$$\begin{aligned} \log m_{ijkl} \;=\;& u + u_{1(i)} + u_{2(j)} + u_{3(k)} + u_{4(k)} + \\ & u_{12(ij)} + u_{13(ik)} + u_{14(il)} + u_{23(jk)} + u_{24(jl)} + u_{34(kl)} + \\ & u_{123(ijk)} + u_{124(ijl)} + u_{134(ikl)} + u_{234(jkl)} + u_{1234(ijkl)} \end{aligned} \tag{5.8}$$

This equation is analogous to analysis of variance (ANOVA) models. For an array with equal number of categories (L) of each variable, the first term is the "grand mean":

$$u = \sum_{i,j,k,l} \frac{g_{ijkl}}{L^4}$$

where $g_{ijkl} = \log m_{ijkl}$. The term $u_{1(i)}$ captures the deviation of cell values from the mean that are attributable solely to variable 1:

$$u_{1(i)} = \frac{g_{i+++}}{L^3} - \frac{g_{++++}}{L^4}$$

where the notation g_{i+++} indicates summation over the variables replaced by pluses, i.e. $g_{i+++} = \sum_{j,k,l} g_{ijkl}$. Analogous expressions hold for $u_{2(j)}, u_{3(k)}, \text{and} u_{4(l)}$, with appropriate marginal sums replacing g_{i+++}. Similarly, higher order terms in Equation 5.8 express the contributions due to multivariable effects.

The u-terms must sum to zero over all subscripted variables. Given this constraint the number of u-terms is exactly equal to the number of cells in the array, hence the term *saturated*; as it stands, the model does not estimate anything. However, it provides us with a formalism for eliminating higher-order interaction among the variables, precisely the quantities that we cannot estimate reliably from the sparse raw counts. Setting the high order terms to zero effectively reduces the number of parameters that describe the structure of the contingency table. The available data is then used to obtain maximum likelihood estimates of these parameters, rendering an *unsaturated* log-linear model of the table.

As shown in [12, Sec. 3.3] the sufficient statistics for the ML estimates are obtained by relating the log-likelihood of Equation 5.7 to the model for the expected cell counts, Equation 5.8. In general, these sufficient statistics are the marginal sums of the table corresponding to the highest order u-terms left in the approximate model. Furthermore, an iterative procedure exists that allows us to compute the ML estimates of the expected cell counts, $\hat{m}_{ijkl}$, without first finding the parameters of the log-linear model. It makes use of the constraint [11] that the ML estimates of the sufficient statistics (marginal sums) must be equal to the observed values. For instance, if

x_{ij++} is a member of the set of sufficient statistics then:

$$\hat{m}_{ij++} = x_{ij++}$$

This estimation procedure effectively computes cell counts that satisfy the marginal sums specified by the unsaturated model. These recomputed cell counts are the estimated expected values from which we can directly obtain the desired probability:

$$\Pr[c_1c_2c_3c_4 \mid q] = \hat{p}_{ijkl} = \frac{\hat{m}_{ijkl}}{N} \tag{5.9}$$

The procedure itself is outlined in Appendix B.

There are numerous approximate models we could try as long as they obey the hierarchy principle: the setting of any u-term to zero implies that all its higher order relatives are set to zero. However, since there is no compelling reason for treating one channel differently from others, only two approximations retain the appropriate symmetry. In the first, the highest order effects $u_{1234(ijkl)}$ are set to zero, in the second, all third order effects ($u_{123(ijk)}$, etc.) are eliminated.[3] The hierarchy principle requires, of course, that in the latter case the fourth order effect be set to zero as well.

A final issue that must be faced when using this method is the possibility that some elements of the marginal sums that we try to fit will themselves contain zeros. This is quite likely, especially for the unsaturated model eliminating only the fourth-order interaction term. Say, a cell $x_{+j'k'l'}$ is zero: the requirement that all cells be non-negative results in:

$$\forall i: \; \hat{m}_{ij'k'l'} = 0$$

A common statistical sleight of hand that avoids this problem is to add $1/2$ to *all* cells of the marginal counts and that is what was done.

[3] We could proceed further and eliminate the second order effects, leaving only the grand mean and the terms $u_{1(i)}$, etc. However, a model so simplified becomes equivalent to the independence assumption of Sec. 5.3.2.

5.3.4 Pseudo-Bayesian Analysis

In contrast to the method of Section 5.3.3, the approach described here does not attempt to model the structure of each output combination array. Instead, it tries to interpolate between the highly specific but mostly unreliable combination counts and some more robust but less well resolved estimates. The interpolation takes the form:

$$\hat{p}_{ijkl} = \frac{N}{N+K}(x_{ijkl}/N) + \frac{K}{N+K}\lambda_{ijkl} \tag{5.10}$$

where the notation of the previous section has been retained (see Equation 5.9) and λ_{ijkl} is the previous (presumably "over-smoothed") estimate of p_{ijkl}. K controls the proportions in which the two factors will be combined and must itself be estimated in some fashion.

Pseudo-Bayesian analysis [12, Chapter 12] offers one method for calculating a K dependent on the available data x and the initial estimates λ. The optimal K is calculated to minimize the expected value of the departure of $\hat{p}$ from the true probability p over all possible combinations:

$$\hat{K} = \arg\min_{K} N \sum_{i,j,k,l} \mathrm{E}[\hat{p}_{ijkl}(K) - p_{ijkl}]^2 \tag{5.11}$$

Solving Equation 5.11 leads to:

$$\hat{K} = \frac{1 - \sum_{i,j,k,l} p_{ijkl}^2}{\sum_{i,j,k,l}(p_{ijkl} - \lambda_{ijkl})^2} \tag{5.12}$$

Unfortunately, the result depends on the unknown p_{ijkl}. An estimate of the optimal value of K can be obtained by replacing p_{ijkl} with the direct frequency of the cell's occurrence:

$$\hat{K} = \frac{1 - \sum_{i,j,k,l}(x_{ijkl}/N)^2}{\sum_{i,j,k,l}(x_{ijkl}/N - \lambda_{ijkl})^2} \tag{5.13}$$

The denominator of Equation 5.13 is large when the previous estimate of the probabilities, λ_{ijkl} and the direct cell frequencies are widely divergent, indicating that new data, x_{ijkl}, disagrees with our original picture of the distribution. This leads to a small $\hat{K}$ which in turn places greater weight on the new data term in Equation 5.10. Conversely, close overall agreement between the cell frequencies and previous estimates favors λ_{ijkl}, thus discounting the (presumably few) cells of the matrix whose frequencies differ from their expected values.

The numerator of Equation 5.13 tends to be larger when the counts x_{ijkl} are spread out rather than concentrated in a few cells. For instance, it goes to zero if all counts occur in only one cell. Thus the estimate from cell counts is favored when the new data suggests extreme rather than smooth probability values p_{ijkl}. This agrees with the assessment of the risk of different estimators given in [12, Sec. 12.2.2].

Further analysis of asymptotic behavior of $\hat{K}$ under different assumptions about p_{ijkl} in Equation 5.12 [12, Sec. 12.7] leads to the estimate that was actually used in the current investigation[4]:

$$\hat{K} = \frac{N^2 - \sum_{i,j,k,l} x_{ijkl}^2}{\sum_{i,j,k,l} x_{ijkl}^2 - 2(N-1)\sum_{i,j,k,l} x_{ijkl}\lambda_{ijkl} + N(N-1)\sum_{i,j,k,l} \lambda_{ijkl}^2 - N} \quad (5.14)$$

By using Equation 5.14 in 5.10 we can obtain estimated probabilities for all output combinations.

The remaining issue is the source of the robust estimates λ. In this case a natural choice is to use the probabilities estimated via the independence assumption, Equation 5.6 and this was indeed done.

5.3.5 Occurrence-weighted Interpolation

The final method continues the concept of interpolating among estimates of the a posteriori probability. The interpolation proposed is of the form:

$$\hat{p}_{ijkl} = r_1 \frac{x_{ijkl}}{N} + r_2 \frac{x_{ij++}x_{++kl}}{N^2} + r_3 \frac{x_{i+++}x_{+j++}x_{++k+}x_{+++l}}{N^4} \quad (5.15)$$

Equation 5.15 combines probability estimates at three levels of robustness/specificity:

[4] This estimate differs only slightly from Equation 5.13 as can be seen by approximating $N-1$ by N and dividing through by N^2 in the latter.

- direct estimate from the full table of output combination counts
- product of estimates based on combining pairs of channels directly (i.e. the pairs, 1 and 2, and 3 and 4, are assumed independent)
- product of individual channel estimates: the independence assumption

The requirement that $\hat{p}_{ijkl}$ represent a valid probability measure constrains the values that the coefficients r can take. In fact, this interpolation admits only two independent coefficients. This leads to the alternative expression:

$$\begin{aligned} \hat{p}_{ijkl} &= \alpha\frac{x_{ijkl}}{N} + (1-\alpha)\left[\beta\frac{x_{ij++}x_{++kl}}{N^2} + (1-\beta)\frac{x_{i+++}x_{+j++}x_{++k+}x_{+++l}}{N^4}\right] \\ & 0 \leq \alpha, \beta \leq 1 \end{aligned} \tag{5.16}$$

This variation underscores the idea: we want to use as much of the direct estimate as possible, getting the remaining fraction $(1-\alpha)$ of the probability from the combination of the other two estimates. The parameter β determines their relative contributions.

The key problem is determining the magnitudes of α and β. They should reflect our relative confidence in the three available estimates. One way to gauge this is by inspecting the total number of times a particular channel output combination $\{i, j, k, l\}$ occurred in all of the training data. Qualitatively, if this combination occurs frequently, then the direct estimate of the a posteriori probability should be accorded a high weight even if it is *never* seen for the particular phone q on which we're conditioning the current probability. The resulting low probability will have meaning when used in Equation 5.2.

The parameters of Equation 5.16 can be expressed as[5]:

$$\begin{aligned} \alpha &= f_\alpha\left(\sum_{q\in Alphabet} x^q_{ijkl}\right) \\ \beta &= f_\beta\left[\min\left(\sum_{q\in Alphabet} x^q_{ij++}, \sum_{q\in Alphabet} x^q_{++kl}\right)\right] \end{aligned} \tag{5.17}$$

[5] The superscript notation x^q explicitly shows the conditioning of the counts on the underlying phone q.

Strictly speaking we might want to make the functions depend on the total count relative to the size of the alphabet (and thus the size of the count tables). However, since the latter was not altered throughout the investigation, these expressions are general enough.

There are certain obvious constraints on f_α and f_β, for instance, they should be monotonically non-decreasing and asymptote to one. However, the exact form can only be arrived at empirically. In this case the functions were arrived at by trial and error as described in Sec. 6.4.

5.4 Final Segmentation

Each of the five methods described above integrates individual frame labels. What is specifically ignored at this level is the question of segmentation as is the relationship between neighboring frames. As a consequence the frame stream contains, for instance, frequent single frames of different labels following one another or single vowel frames surrounded by frames assigned to a different vowel. The goal of the final segmentation is to "clean up" the frame stream and produce a sequence of phones.

Initially one might contemplate a heuristic approach, invoking such rules as minimum duration, merge of "mixed" stop consonant frames, etc. However, we have a ready system that is potentially capable of converging to such rules automatically: another HMM engine. For its purposes the input sequence of frame labels is indistinguishable from the output of a 48-prototype vector quantizer.

Figure 5-4 shows the phone model employed in this final HMM. It is similar to the model used in the sub-recognizer HMMs (Figure 4-1) but includes an additional state with no self-transition. This state is used to model the transition to the next phone as shown in Figure 5-5 which demonstrates the interconnections of individual phone models that define the effective language model. An additional transition from the initial state in the phone model to the new state is also added. This is done to maintain two state transitions as the shortest possible duration of phone; stop consonants frequently exhibit durations around 20 ms.

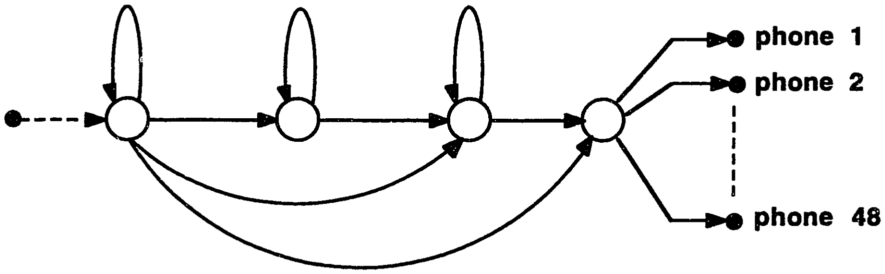

Figure 5-4: Single phone Markov model in the final HMM.

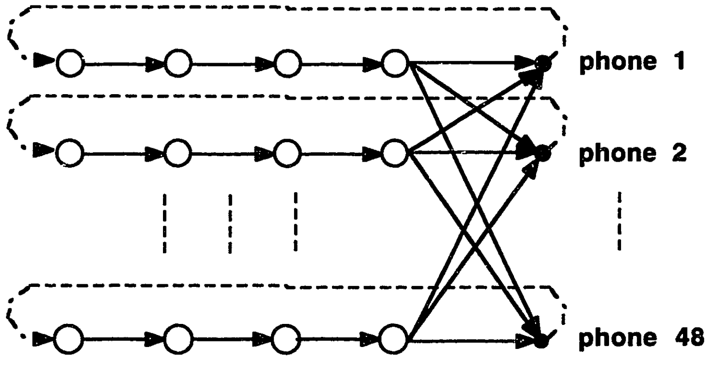

Figure 5-5: Connections of the phone models effecting the bi-gram language model of the final HMM. Some of the within-phone transitions have been omitted.

Unlike the sub-recognizer HMMs, the final HMM utilizes a bigram language model. The transitions between phones have different probabilities (the a_{ij}'s) depending on the particular pair of phones involved. Just as the other HMM parameters, these probabilities are estimated via the forward-backward algorithm from training data. Once the phone models are trained, the final recognition proceeds via the Viterbi algorithm as described in Section 4.2.2. The output of this stage constitutes the final output of the recognizer.

Chapter 6

Integration Results

The main focus of the experiments was to evaluate the proposed methods of frame label integration described in the previous chapter. However, there were other, ancillary, variations in the system configuration for which results were obtained. This included, again, choice of original structure parameters, and the effect of increasing the training data.

Except where explicitly stated otherwise, all training, parameter and probabilities estimation, etc. were performed using data of the SX sentences in the TRAIN portion of TIMIT. Performance results were obtained for the SX sentences of the TEST part of the database (c.f. Table 4.1).

6.1 Scoring

The first score of interest produced by the integration process is the fraction of the frames in the test data matched correctly by the frame label integration stage. Because of the relatively frequent "single frame" phones in the output of this stage, phone scoring is not feasible. The frame scoring rules employed in the sub-recognizer evaluation, Section 4.3, were also followed here.

On the other hand, the final output must be scored according to the correctness of the recognized phone sequence. In obtaining this score, published scoring rules for phonetic recognition were followed [42, 43, 76]. The recognized phone sequence was

A Priori Probabilities	Percent Correct			
	Two-at-a-time 2 codebooks		Occurrence-weighted 3 codebooks	
	Frames	Phones	Frames	Phones
all equal	48.4	43.6	55.4	50.1
frame frequency	53.3	45.1	59.0	51.8
phone frequency	52.7	46.1	58.5	53.2

Table 6.1: Results for different choices of a priori probability in the frame combination rule. LPC structure parameters. See text for details.

aligned to the correct phone sequence by a dynamic programming algorithm described in Appendix C. The *Percent Correct* score was calculated as the percentage of the phones in the TIMIT transcription that were matched with identical labels in the output of the recognizer. Thus substitutions and deletions are counted as errors but insertions are not. The latter were always between 10 and 11 percent of the input phones which is slightly lower than the 12% rate reported in [43] and [76].

The scores reported in most of this chapter refer to the 48-phone alphabet used throughout (Sec. 4.1). Phonetic recognizers using the TIMIT database often reduce this set further to 39 elements by merging several acoustically similar phones [43] and report the performance on this set, often referred to as the CMU/MIT reduced set. Accordingly, the last section lists the performance of the best versions of the current system using this reduced alphabet. It is formed by merging the following phones: [ih,ix]; [ah,ax]; [aa,ao]; [l,el]; [n, en]; [sh, zh] and [cl,vcl,epi,sil].

6.2 A Priori Probabilities

Equation 5.2 indicates that the frame combination rule depends on the choice of a priori probabilities. As explained in Section 5.2 three different approaches to obtaining this probability were considered: all equal, frequency of phone frame occurrence, and frequency of phone occurrence. Table 6.1 shows a comparison of the performance under these conditions.

Two combination methods were tested: the two-at-a-time method of Section 5.3.1 and the occurrence-weighted interpolation of Section 5.3.5. The former was used with 2-codebook input to the sub-recognizers, the latter with 3-codebook input. LPC structure parameters were used in both cases. Assuming all phones equally likely leads to the lowest score both for frames and phones. The frame scores are highest when frame occurrence frequency is used as the a priori probability estimate. This agrees with the analysis of Section 5.2.

As signaled in that section, however, a priori probabilities that lead to highest frame scores do not have to correspond to highest *phone* scores. The data of Table 6.1 bear this out. For both conditions, the phone score is highest when the phone occurrence frequency is used as the a priori probability. Consequently, when testing other aspects of the system, this was the estimate used when applying Equation 5.2.

6.3 Parameter Arrangement

As noted in Section 4.4.1, increasing the number of parameters used to describe a frame of input speech improved the performance of each sub-recognizer. Similarly, splitting these parameters into three rather than two codebooks resulted in higher scores. The ultimate measure of performance, however, rests with the output of the label integration stage. It is to be expected that better channel scores would result in an improved overall recognition rate. However, given the philosophy of the frame combination rule (Equation 5.2) the correctness of the sub-recognizers outputs is not as important as their consistency. In an extreme example, the sub-recognizers could be always wrong but if each of their combinations occurred uniquely for a given underlying phone the final decision would be always correct.

Table 6.2 shows the relevant results with the LPC waveform structure parameters. Compared are: the average frame correct rate for the four individual channels, percent correct of the frames after combination, and the percent of phones correct in the final output. The frames were combined using the independence assumption.

As the results indicate, the overall performance does seem to be reasonably cor-

Parameter	Percent Correct		
Arrangement	Average Channels	Combined Frames	Combined Phones
static only	28.0	50.6	43.3
full - 2 codebook	33.1	56.9	48.9
full - 3 codebook	37.2	58.3	52.3

Table 6.2: Average sub-recognizer, and overall performance for LPC structure parameters and various parameter arrangements. Independence assumption combination.

related with the performance at the sub-recognizer level. However, the magnitude of the improvement is not as clear: the increase in combined performance from 2 to 3 codebooks is only about 34% of the average individual channel improvement. The corresponding number when the change is from static to full parameter set is 124%. Since the combination was done using the independence assumption, this discrepancy cannot be blamed on artefactual effects of sparse training data. More likely it reflects the effect of different interactions of channel errors in the combination process.

The performance of the 3-codebook arrangement was thus seen to be superior both at the sub-recognizer and the the overall level. Subsequent experiments concentrated, therefore, on versions of the system using 3-codebook sub-recognizers.

6.4 Label Integration Methods

6.4.1 Frame Level Results

As seen in Section 5.3, the different methods of integrating the label streams produced by the sub-recognizers center on different approaches to estimating the conditional a posteriori probabilities of the possible output combinations. Five such methods were proposed and evaluated using LPC structure parameters.

It should be noted that once the probabilities of Equation 5.2 are fixed, the mapping between the four-fold combinations of channel outputs and the integrated label is completely determined. In practice this means that the frame-by-frame combination stage involves essentially no computation. Rather, a simple table lookup is required.

The minimization of computation at recognition time is of course highly desirable for a real-time application.

Two-at-a-time Method (1)

The collection of appropriate confusion matrices needed to merge the pairs of channels and then to combine the compound channels L and H was straightforward. One problem that did surface was the absence of a very small, but non-zero, number of channel pair combinations in the training data. To deal with such cases, the integration rule was amended such that a frame whose label pair had not been seen was mapped to the same phone as the immediately preceding frame. In the event fewer than 0.01% of the original channel pair combinations were missing, and only about 0.03% of the HL combinations. The ad-hoc fix thus seemed justified. This method gave 55.7% correct combined frames.

Independence Assumption (2)

No special provisions had to be made for this method; the individual channel confusion matrices were well covered. Frames were combined at the rate of 58.2% correct.

Log-linear Models (3)

Two versions of this method were implemented:

I fourth order effect set to zero

II all third order effects set to zero (hierarchy principle requires that the fourth order effect be eliminated also)

As explained in Section 5.3.3, 1/2 was added to all cells of the marginal sums to avoid the problem of zero counts. In the case of model I, about 99% of the marginal sums' cells were, in fact, zero. The corresponding figure for model II was around 40%.

Model I gave 53.5% frames correct, model II, 58.4%.

α		β	
Count Range	Value	Count Range	Value
< 10	0	< 20	0
11–20	0.1	21–50	0.2
21–50	0.3	51–200	0.5
51–100	0.7	201–500	0.9
> 100	1	> 500	1

Table 6.3: Ranges and values for the optimal weights for occurrence-weighted interpolation. This effectively constitutes the functions f_α and f_β of Equation 5.17.

Pseudo-Bayesian Analysis (4)

The preliminary, robust estimates of the a posteriori probability, λ_{ijkl} were obtained from the independence assumption analysis, whose result is given above. Averaged across the 48 phones, the weight in Equation 5.10 assigned to x_{ijkl}/N was 0.901. The percentage of frames identified correctly was 55.7.

Occurrence-weighted Interpolation (5)

This method, as described in Section 5.3.5, requires the specification of the interpolation weights α and β as functions of the total count of a given four-fold output combination (Equation 5.17). The approach here was to divide the count into several ranges and experiment with assigning α and β values to them. The optimal factors arrived at, using this approach, are listed in Table 6.3.

This manual optimization procedure used SI sentences to measure the performance of the alternative weight values. Accordingly, the "official" test data did not influence the choice of these values. The performance of this method seemed fairly insensitive to the choice of α and β, varying by tenths of percent. For the factors of Table 6.3 58.5% of test frames were identified correctly.

Figure 6-1 summarizes the frame recognition scores listed above. It also includes scores for two of the integration methods when the other two parameter sets were used in the sub-recognizers. These data indicate that three frame integration methods

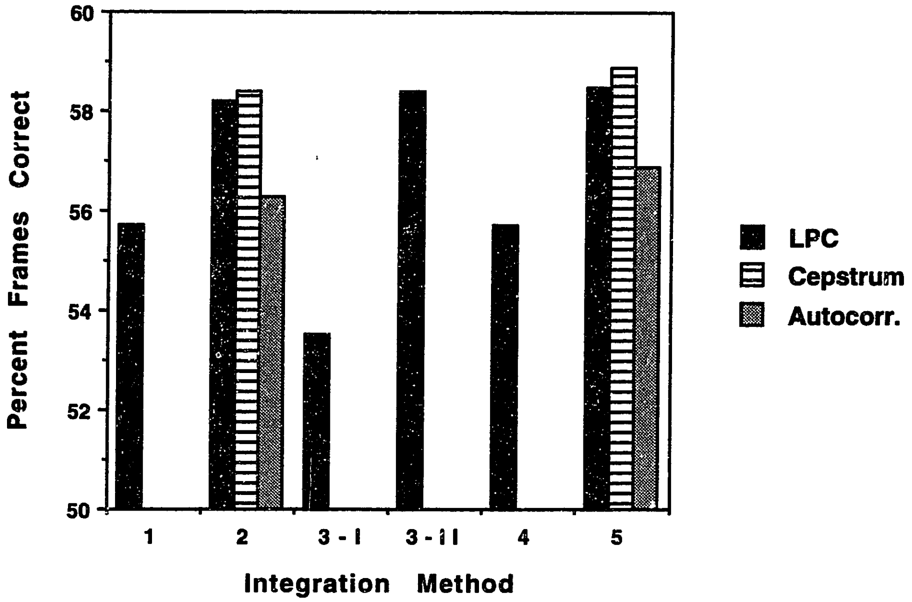

Figure 6-1: Frame integration results for different parameter sets and integration methods. Section 6.4 lists the integration methods corresponding to the indices indicated in the figure.

appear to achieve roughly comparable scores: independence assumption, log-linear modeling II, and occurrence-weighted interpolation. The last method achieves the highest scores by a slight margin.

For the two methods tested, the results for the different parameter sets follow the trends at the individual channel level (c.f. Table 4.3). Cepstrum parameters perform best, followed by LPC and autocorrelation. The differences remain quite small, however.

6.4.2 Phone Level Results

Integrating the frames of the original training data as labelled by the sub-recognizers produced frame streams appropriate for training the HMMs to be used as the final "cleanup" and segmentation stage. The system versions showing the most promising frame scores were evaluated for phone scores.

Frame Integration Method	Percent Correct	
	Frames	Phones
Independence assumption	58.2	52.3
Log-linear II	58.4	52.5
Occurrence-weighted	58.5	53.2

Table 6.4: Comparison of frame and phone scores for the three best frame-by-frame label integration methods. LPC structure parameters.

LPC Parameters

Table 6.4 lists the results of the final segmentation results for the three best integration methods when LPC structure parameters were used. The frame level scores are also given. Again, no dramatic differences are evident. The best overall frame and phone scores belong to the occurrence-weighted interpolation method.

Occurrence-Weighted Interpolation

The performance under the occurrence-weighted method of frame integration was evaluated for the three developed parameter sets. Table 6.5 lists the results. The scores consistently favor the cepstrum coefficients over LPC and autocorrelation. The differences, however, remain quite small.

The table lists also a "hybrid" system, resulting from combining the outputs of sub-recognizers operating on different parameter sets. The individual channel results in Table 4.3 indicate that while the cepstrum parameters perform best on average, they are not uniformly superior in every channel. The hybrid system, therefore, combined the outputs of sub-recognizers displaying the highest frame recognition rate: channel 1 using LPC parameters, channels 2 and 3 using cepstrum, and channel 4 using autocorrelation. The performance of this system failed, however, to surpass the strictly cepstrum system.

Waveform Structure	Percent Correct	
Parameters	Frames	Phones
LPC	58.5	53.2
Cepstrum	58.9	53.4
Autocorrelation	56.9	52.1
Hybrid	58.9	53.4

Table 6.5: Comparison of frame and phone scores for the three parameter sets. Occurrence-weighted interpolation method.

	Percent Frames Correct				
Data	Channel				
	1	2	3	4	Integrated
Test	39.3	40.9	37.9	31.4	58.9
Train	44.8	45.9	43.0	36.3	69.8

Table 6.6: Comparison of frame scores for test and train data. Cepstrum parameters. Occurrence-weighted integration.

6.5 Increasing Training Data

In all the evaluations reported up till now the SX TRAIN sentences were "recycled" at each training stage. There are three such stages: sub-recognizer training, estimation of the a posteriori channel combination probabilities, and the training of the final segmentation HMM. Such re-use may result in the system parameters at each stage being fit too closely to the training data instead of following the distribution of general speech.

Table 6.6 illustrates this effect. As expected, running the recognition process on the data used to generate the recognizer parameters results in higher scores than on test data, unseen during training. At the sub-recognizer level this is not necessarily a problem as such, as long as the models are not *overtrained.* With too many iterations of the training algorithm the model probabilities become tuned to the idiosyncrasies of the training data and performance on test data actually declines [42]. While no systematic attempt to optimize the number of iterations was made, initial development results did not indicate that overtraining was a concern.

The problem of estimating the conditional a posteriori probabilities is different. There is often no obvious point in the estimation procedure at which one can check the intermediate probability estimates and stop iterating if recognition of test material suffers from further fitting. An exception is the occurrence-weighted interpolation method where α and β are adjusted for maximum overall recognition. If the SX TRAIN sentences were used to establish these parameters, we would draw the conclusion that α should be 1. In fact, training data integrated this way gives over 90% frames correct.[1] It was, therefore, necessary to use the SI sentences, when optimizing α and β.

Because of the large number of conditional probabil'ties relative to the available data, the estimates are likely to be sensitive to any peculiarities of the training data distribution. The SX TRAIN sentences likely exhibit different statistics when processed by the sub-recognizers than do the TEST sentences as suggested by Table 6.6. It should be beneficial to include additional training data when estimating the conditional a posteriori probabilities.

The only additional corpus of data available for training were the SI sentences of the TRAIN section of TIMIT.[2] The effect of including or substituting these data on overall performance was investigated using cepstrum parameters and the occurrence-weighted frame label integration method. Table 6.7 compares the results obtained when the data from which the a posteriori probabilities are derived is varied. The final "cleanup" HMM was trained in all cases on the SX sentences. Note also that the sub-recognizer scores on the SI sentences were very similar to those achieved for the SX TEST material.

Again, the scores do not change dramatically. However, observe that the frame score with the SI sentences is basically identical to that obtained with just the SX sentences. This is significant since there are only three-fifths as many of the former

[1] Note that in this case the problem of a four-fold combination not seen during "training" does not occur.

[2] The SI sentences of the TEST section could not be used because of the overlap of speakers with the designated testing data.

Probability Estimation	Percent Correct	
Sentences	Frames	Phones
SX	58.9	53.4
SI	59.0	53.3
SX+SI	59.4	53.7

Table 6.7: Comparison of performance when different data sets were used to estimate the conditional a posteriori probabilities. Cepstrum parameters. Occurrence-weighted integration.

Structure	Percent Phones Correct	
Parameters	48-phone	39-phone
SX estimation sentences		
LPC	53.2	58.0
Cepstrum	53.4	58.3
Autocorrelation	52.1	56.4
Hybrid	53.4	58.2
SX + SI estimation sentences		
LPC	53.4	58.3
Cepstrum	53.7	58.5

Table 6.8: Phone recognition results for the 48 and 39-phone alphabets and two data sets used to estimate the a posteriori probabilities for occurrence-weighted frame integration.

as there are of the latter. That the a posteriori probabilities are estimated just as well from this scarcer data is most likely due to its unskewed statistics. On the other hand, using both the SX and SI sentences does slightly improve the performance, the increased amount of data apparently offsetting some of the "re-cycling" effect.

6.6 Reduced Alphabet

The best versions of the recognizer from those tested were rescored using the 39-phone alphabet (Sec. 6.1). Table 6.8 compares the results.

The merged phones account for roughly 4.6 to 4.8 percent of the errors counted in the 48 phone alphabet. Qualitatively, the results remain unchanged: the cepstrum parameters have a slight edge over LPC which in turn are better than autocorrelation. Numerically, the differences remain quite small.

Chapter 7

Discussion

7.1 Performance

When discussing performance, attention will be focused on the 39-element alphabet (Sec. 6.1), since scores in the literature are often reported for that set. Also, the finer distinctions of the 48-element alphabet are largely unnecessary for the autocueing application which motivated this work.

7.1.1 Error Pattern

Appendix D gives several confusion matrices obtained for different parameterization schemes in the sub-recognizers. While individual phone scores vary, the general recognition pattern is fairly similar across these system versions. Specific rates quoted below refer to the best performing system: cepstrum coefficients with both SX and SI sentences used in the probability estimation, Table D.1.

As shown in Table 7.1 the general phone groups of vowels, consonants and silence[1] are relatively rarely confused among each other. The deletion rate of 8.96% overall, is similar to that of Lee and Hon (see below), 9.72%.

There is not much structure evident in the error pattern of vowels. While they appear to be rarely confused with consonants, they are broadly confused among each

[1] Voiced and unvoiced closures are counted as consonants here.

Input	Output Phone (Percent)			
Phone	Vowel	Consonant	Silence	Deletion
Vowel	83.2	8.2	0.3	8.3
Consonant	3.9	85.5	0.8	9.8
Silence	0.2	5.2	91.8	2.8

Table 7.1: Confusion matrix for general phone categories. Cepstrum parameters, occurrence-weighted interpolation, SX+SI estimation sentences.

other. The diphthongs "ey", "ay", and the vowel "iy" display best accuracy. A few of the more frequent mistakes are not surprising, for instance between "eh" and "ae". Diphthongs and longer vowels are generally less likely to be deleted.

The overall recognition rate for the fourteen vowels and diphthongs is 48.0%, for the 24 consonants[2] 52.4%. Part of the problem for vowels may be the sub-recognizers' difficulty in segmenting vowels, especially sequences of vowels. Unlike obstruent boundaries, vowel boundaries are usually less clearly defined and even more so within the relatively narrow bandwidths of the sub-recognizers. Inconsistent segmentation at the channel level likely leads to difficulties when integrating the individual labels. Furthermore, the human recognition experiments that lead to consideration of the proposed system tested exclusively consonant reception. As far as we know no studies have established analogous performance on vowels. It may be that narrowband recognition of vowels is simply impractical.

More perhaps may be said about the consonant confusions, which tend to cluster in more obvious ways. The recognition of glides and liquids proves to be quite good. The main errors here are, in fact, not substitutions but deletions. In addition, "r" is most often (almost 20%) confused with "er" and vice versa. Inspection of several test sentences revealed that the error often involved "er" being recognized as "eh" followed by "r". Similarly, "eh-r" would be contracted to "er". For the motivating application these errors probably would be quite acceptable.

Nasals are confused almost exclusively among themselves. In general, "n" appears

[2] After taking into account the merges that produce the 39-phone set.

to be the most frequent substitution for the others. The two affricates, "ch" and "jh", also form an often confused group, as do the stops. Of the latter unvoiced stops are recognized better and are also less likely to be identified as voiced whereas the main error for a voiced plosive is misrecognition as the corresponding unvoiced one. Stops are recognized correctly 48.1% of the time compared to 58.1% reported by Lee and Hon [43].

Fricative recognition is quite high for the unvoiced case with "f", "s", and "sh" scoring above 70%.[3] The same cannot be said for the voiced fricatives. Of these "z" was confused almost exclusively with "s", while the major source of "v" and "dh" errors was deletion. Overall fricative score was 56.2% compared to Lee and Hon's 66.0%. The closure/silence elements were identified at a high rate: 90.1% (Lee and Hon: 92.1%), with most of the errors occurring as deletions.

The observed confusions are consistent for the most part with what is known of acoustic similarity between different phones. It is unclear whether at the present level of performance the system could be used effectively for autocueing. A recognition rate of better than 70% is probably needed for significant benefit to the speechreader [81]. This benefit is dependent, however, on the interaction between the error pattern of the recognizer and phoneme grouping imposed by the cueing protocol. Judicious choice of cue groups can alleviate the effect of some recognizer errors.

Voicing identification, important for the autocuer application, remains inadequate. Overall, 14.7% of voiced consonants are mistaken for unvoiced, and 13.5% of unvoiced are identified as voiced. Of particular concern are the voiced stops: of those identified as stops, 29.1% were mislabeled as unvoiced. By contrast, the analogous figure for unvoiced stops was only 9.4%, a level probably compatible with useful cue production. The assymetry may be caused by the generally longer duration of unvoiced stops which increases the number of frames from which to estimate the conditional probabilities and improves their recognition. The performance on fricatives reflected the same trend: the voicing feature of 32.9% of voiced and 10.7% of unvoiced fricatives was reversed.

[3] In the 39-phone alphabet, where confusions between "zh" and "sh" are not counted as errors.

7.1.2 Overall Performance

The novel recognition structure proposed in this thesis was investigated in the context of speaker-independent phonetic recognition. Since it embedded HMM recognition engines within the overall scheme, its performance is most directly comparable to systems that also use HMMs. Two relevant studies are those by Schwartz *et al.* [76] and Lee and Hon [43]. The former obtained accuracy of 62% for single-speaker recognition on a 550 sentence database, while the latter was 64.1% accurate in speaker-independent mode on TIMIT. Both systems used vector quantization to represent the parameter vectors. The cited scores were obtained with *context-independent* HMMs, using the same model for a phone, regardless of the the phones that surround it in an utterance just as did the sub-recognizers in the current study.

As additional reference, Huang [33] reported 53.2% phoneme accuracy for a single speaker task where training and testing data comprised the same text (although recorded separately). The training data was limited to 98 sentences and dynamic parameters were apparently not used. Zue *et al.* [86] reported a phonetic recognition score of 55% (including insertions as errors) on the SUMMIT system and Digalakis *et al.* [24] achieved 70% although their scoring rules were somewhat different.[4] The last two recognizers are segment-based, stochastic systems and thus cannot be compared easily to the HMM recognizers.

The best score obtained from the proposed multiband system was 58.5% which is within the range of performance achieved by established phonetic recognizers. Nonetheless, it does not constitute an improvement. Since the system of Schwartz *et al.* was only tested in *speaker-dependent* mode, the most useful comparison could be made with the results of Lee and Hon whose system was similar to that studied here in some respects (not overall structure). They also split their waveform parameters[5] into three codebooks and used a bigram language model during recognition. However, their HMM phone model topology was somewhat more elaborate, including two

[4] For instance, substitution of a closure-stop pair by the corresponding single closure or stop was not counted as an error.

[5] Mel-weighted cepstra, differenced cepstra, and power.

additional states. They also used a technique of *co-occurrence smoothing* to prevent the observation probabilities (the b_{ij}'s) from vanishing. They found the performance of models trained with this method to become equivalent to the simpler floor method used in the current system when the training data contains over 64 speakers. However, it is not clear whether the same would be true for a narrowband recognizer.

7.2 Improvements

The feasibility of the proposed recognizer structure was evaluated using a relatively rudimentary implementation of the HMM sub-recognizers. Attention was concentrated instead on the system behavior under different parameterizations and on the question of integrating the sub-recognizer outputs. The approaches to improving the performance of this recognizer may be roughly divided between those that would increase the sophistication of the sub-recognizer HMMs and alterations of the overall structure itself.

7.2.1 HMM Sub-recognizers

A number of improvements over the discrete, context-independent HMMs have been reported. Since the sub-recognizers function independently and are limited only by the bandwidth of their input data, virtually all these enhancements could be included in the revised sub-recognizers.

Context-dependent Models

An effective means of attacking the problem of coarticulation has been the introduction of context-dependent phone models. A phone is represented by a different state-transition network depending on the immediately neighboring phones. This method greatly increases the number of model parameters that need to be trained necessitating the use of various interpolation and smoothing techniques [38, 42]. Nonetheless, recognition rates are improved. In the study cited above, Schwartz *et al.* achieved 81% correct with both left and right context modeled, while Lee and Hon reported

73.8% using only right context. Aside from increased complexity of training, the main drawback of this approach is a significant increase in computation during the recognition phase since more possible paths have to be searched. Also, since a phone model specifies its “future' neighbor, delays are introduced and real-time recognition becomes problematic.

In general, by allowing a greater delay during recognition, better recognition accuracy is obtained. However, at least in the autocueing application, the delay incurred may interfere with the listener's reception of the cue. The exact allowable delay is not yet known. Of course, different delays may be acceptable for other applications.

Parameter Vector Modeling

The recognition of band-limited speech signal was conceived in part to alleviate the problem of efficient and robust estimation of the observation probabilities of the parameter vectors (Sec. 1.1.3 and Chap. 3). However, as the results of Table 4.2 indicate, VQ-distortion still has a measurable effect on performance. This is evidenced by an increase in sub-recognizer performance when the parameter set is split among three rather than two codebooks.

An improvement might be effected by changing the normalization procedure when combining parameters in the same vector prior to quantization. Currently, the variances of all parameters are equalized. However, this may not be desirable for parameters such as cepstrum or LPC coefficients. Their relative variances are retained in [43] where different weights are used only for groups of parameters such as differenced cepstrum, power, etc. Alternatively, the entire parameter set may be transformed to a (possibly) lower dimensional, vector by matrix multiplication that normalizes the *covariance* statistics of the vector [25] or by analysis into principal components (Karhunen-Loeve expansion) [62].

Continuous HMMs, where parameters are not quantized but rather assumed to obey a specified probability density function, often show an improvement over discrete modeling although some of the results have been equivocal [9, 42]. On the other hand, semi-continuous models [9, 34, 64] which represent a compromise appear to consistently improve recognition rates. In particular, Huang found that phonetic recognition rate increased from 53.2% to 60% when semi-continuous models were used (c.f. Sec. 7.1.2).

Recognition phase computation load of semi-continuous HMMs is larger than for discrete models. Nevertheless, narrowband recognizers would likely require fewer distributions to describe their parameter sets than a wideband system. This lessens computational requirements and should also result in robust estimation of the distribution parameters.

Other

Numerous additional modifications to the basic HMM have been suggested. (e.g. [36, 44]). These include variations on the context-dependent phone model and explicit modeling of phone duration. A relatively recent alternative to the forward-backward training algorithm is corrective training [3, 45] which has shown promise. It is not immediately apparent, however, how these modifications would affect a narrow-band recognizer.

7.2.2 Structure-specific Modifications

Largely independent of the specific algorithms used to implement individual channel recognition, the overall structure is concerned with two main issues: what information is supplied to the sub-recognizers, and in what form, and how the outputs of the sub-recognizers are combined. Both these aspects are open to further exploration.

Approach to Sub-recognizers

Comparing the sub-recognizer results to human recognition rates (Table 7.2) reveals an interesting discrepancy. The sub-recognizers tend to achieve very similar scores in all channels, with channel 4 somewhat lower than the others. The experiments of Milner *et al.*, on the other hand, found human subjects performing significantly better on speech in channel 3 than any other channel, and better in channel 2 than

Channel	Percent Correct	
	Human Average Consonants	Sub-recognizer Frames
1	37	40
2	54	40
3	71	37
4	37	31

Table 7.2: Comparison of average scores by subjects in Milner's study [54] and sub-recognizer scores averaged across the three waveform structure parameters.

1.[6] This observation suggests that the current approach may not be taking sufficient advantage of information available in channels 2 and 3.

That impression is strengthened by the observation that combining just the lower two channels[7] leads to frame scores of roughly 50%. Since the best frame scores, when all four channels are combined, do not exceed 60%, we again appear not to be optimally extracting the cues from channel 3.

No obvious explanation is immediately apparent. It is possible that none of the tested parameterization schemes is appropriate to the narrowband task. This contention might be supported by the fact that the ad-hoc autocorrelation parameters result in scores only slightly lower than cepstrum or LPC, even though the former were developed using relatively loose heuristics. The assumptions that lead to these rules, may not hold sufficiently in all channels. For instance, tracking the third formant is generally harder than formants one and two, yet the autocorrelation parameterization employs the same method in all channels. Interestingly, while performance under the autocorrelation parameterization is essentially identical to LPC and cepstrum in channel 1, it drops relative to those methods in channel 3 by about 2 percentage points. It seems that a reassessment of the nature of cues present in channel 3 is

[6] Milner's experiments were only concerned with consonants. Nevertheless, the qualitative picture of sub-recognizer performance does not change if only consonants are considered. Because of the different tasks and scoring methods, however, direct comparison of human and sub-recognizer scores for a given channel does not hold.

[7] Using LPC parameters and direct counts from confusion matrices to estimate the conditional probabilities.

necessary.

Another possible explanation relates to the question of the appropriateness of vector quantization, as discussed above. Relatively low human performance in channel 1 indicates fewer available cues than in channels 2 or 3. Consequently, the resolution (VQ distortion) of channel 1 parameters may be less demanding, whereas extracting the cues from channels 2 and 3 would require more careful modeling. If that were the case, some of the suggestions of the previous section should prove profitable.

It seems, nonetheless, that the question of optimal narrowband parameterization remains open. In particular, it might be beneficial to depart from computation of parameters from single, fixed-width frames.[8] Tracking of resonances might be better performed by considering temporally wider waveform regions. On the other hand, such short events as plosive bursts might be better characterized by splitting a frame into smaller sections.

The overall structure in no way mandates the use of HMMs as the sub-recognizers. Some alternative methods appear to show superior results on wideband speech signals, for instance, the segment based system of Digalakis *et al.* [24]. Using such systems should improve performance since most of the results on the multiband system indicate that better performance at the sub-recognizer level translates to better overall recognition as well. One has to be cautious however; the hybrid system's performance (Table 6.8) was lower than all-cepstrum even though it drew from individually highest scoring sub-recognizers. The computational requirements of such alternative systems also have to be carefully considered. One of the attractive features of HMMs is the existence of several efficient algorithms for model alignment at recognition.

A potential advantage of the multiband system is the opportunity to "tune" each sub-recognizer to perform well on a specific class of speech elements, possibly to the detriment of others. The goal is to design sub-recognizers that commit complementary errors that can be eliminated through decision integration process. Such an option is not available to a wideband recognizer. This work did not attempt such an approach.

[8] Delta parameters are dependent on data in several frames; however, they are relatively simple combinations of parameters calculated on purely frame-by-frame basis.

It would require a deeper understanding of the error pattern in each channel and how sub-recognizer parameters affect it.

Decision Integration

Within the constraints of the frame-by-frame integration of the labels generated by sub-recognizers, a fairly broad array of methods was investigated. It seems unlikely that a dramatically better method of estimating the conditional a posteriori probabilities exists. Of the ones tested, occurrence-weighted interpolation produced the best results, with log-linear modeling II[9] and independence assumption as close second and third best, respectively. The estimation of interpolation weights might be further improved, perhaps through an algorithm similar to *deleted interpolation* [38] used to interpolate between HMM parameters. Experience with hand-tuning suggests, however, that only small gains are likely.

The results shown in Figure 6-1 also support the hypothesis that little further improvement in frame integration may be expected within the current system. Three different parameterization schemes and five different frame label integration methods were tested, yet the resulting scores show little variation. The best and worst scores are separated by a mere 5.5 percentage points.

It is hard to approximate what the theoretical limit of frame integration accuracy should be given the sub-recognizer performance. Only arbitrarily extensive training data could show the rate at which multiple sub-recognizer output combinations occur for the same phone. Such confusions, rather than poor estimation of probabilities of rarely occurring combinations, are the fundamental source of error in the system. In the absence of "infinite" data, we compared the limits of performance for three increasingly large sets of data, listed in Table 7.3.

For each set, frame counts vs. underlying phones were gathered *from that set.* Subsequently, frames were assigned optimally, based on these counts. Therefore, the scores shown represent the maximum possible accuracy for each set. Errors here obviously cannot be attributed to inaccurate probability estimation. The decline of

[9] Third and fourth order effects ignored.

Percent Frames Correct		
SX TEST (840 sent.)	SX TRAIN (2310 sent.)	SX+SI TRAIN (3696 sent.)
94.7	92.2	90.1

Table 7.3: Maximum possible frame-by-frame integration accuracy for three different sentence sets.

scores with increasing size of data suggests that the limit of performance lies well below 90%. Of course this limit applies only to frame integration based solely on the four channel outputs for that frame.

An alternative is to abandon the frame-by-frame approach for a more segment-oriented approach. This would have the benefit of simultaneous integration across time as well as across channels. In the present system the "across time" integration is performed by the final HMM which only sees the integrated labels.[10] To accomplish this, the problem of incompatible segmentation in different channels has to be faced.

One approach would attempt a dynamic programming alignment of the four segmentations, followed by a maximum likelihood combination of aligned phones (with provisions for deletion and insertion) rather than frames.

Alternatively, one could force the sub-recognizers to produce aligned outputs by providing them with a segmentation or several possible segmentations of the utterance at the outset. The algorithm producing the segmentation, something like the dendrogram approach perhaps [30], would operate on the wideband signal. This would allow it access to acoustic landmarks not available to sub-recognizers. On the other hand, the segmenter would not have to make a decision on the identity of phones. This preserves the concept of accurate waveform representation within band-limited channels.

Multiband recognition on a segmented signal resembles more closely the human experiments cited in Section 1.1.1. There, only syllables were tested, effectively re-

[10] In an attempt to increase the information available to the final HMM, the second best frame identification produced by the frame integration procedure was also provided as a symbol from a second codebook. Performance did not change. However, this method violated at least one assumption of multi-codebook systems: that the symbols from two codebooks are independent. Here they were, in fact, mutually exclusive.

moving the problem of locating the phones prior to identification.

Forced segmentation recognition would also allow use of “second guesses” from the sub-recognizers when combining labels. Currently only the top choice was used. Partly this was due to the sparseness of data to estimate the probability of even the top-guess combination. It is also difficult, however, to meaningfully define the second best phone on a frame-by-frame basis.[11]

7.3 Epilogue

The structure proposed in this thesis is novel and required rethinking of some tenets of speech recognition. While its performance on the given task was not superior to more established methods, it seems high enough to warrant further investigation. It may very well be that the wideband recognition methods that were adapted to narrowband use are simply inappropriate. Further study of human derivation of cues from filtered speech should also prove beneficial to the development.

An area where some of this research could be further applied is in audio-visual speech recognition where signals from two different modalities are present. That situation naturally lends itself to the multiband approach in which the bands corespond to the different modalities.

[11] Since the Viterbi search aligns frames to data, the second best frame is usually another state within the same phone.

Bibliography

[1] A. Averbuch *et al.* "Experiments With The TANGORA 20,000 Word Speech Recognizer." *Proceedings of ICASSP-87*, pp.701–704, Dallas, Texas, Apr. 1987.

[2] Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. "A maximum likelihood approach to to continuous speech recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2): 179–190, Mar. 1983.

[3] L.R. Bahl, P.F. Brown, P.V. de Souza, and R.L. Mercer. "Estimating Hidden Markov Model Parameters So As To Maximize Speech Recognition Accuracy." *IEEE Transactions on Speech and Audio Processing*, 1(1): 77–83, Jan. 1993.

[4] James K. Baker. "The DRAGON System – An Overview." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23(1): 24–29, Feb. 1975.

[5] James K. Baker *et al.* "Cost-effective speech processing." *Proceedings of ICASSP-84*, pp. 9.7.1–4, 1984.

[6] L.E. Baum. "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes." *Inequalities*, 3: 1–8, 1972.

[7] BBN Laboratories Incorporated. "Development of a Vidvox Feasibility System." Report No. 5906, July 1985.

[8] Robert L. Beadles. personal communication with Prof. Louis Braida made available to the author, Aug. 1989.

[9] Jerome R. Bellegarda and David Nahamoo. "Tied Mixture Continuous Parameter Modeling for Speech Recognition." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(12): 2033–2045, Dec. 1990.

[10] C.A. Binnie, P.L. Jackson and A.A. Montgomery. "Visual intelligibility of consonants: A lipreading screening test with implications for aural rehabilitation." *Journal of Speech and Hearing Disorders*, 41: 530–539, 1976.

[11] M.W. Birch. "Maximum likelihood in three-way contingency tables." *Journal of the Royal Statistical Society*, Ser. B25, pp.220–233, 1963.

[12] Yvonne M.M. Bishop, Stephen E. Fienberg, and Paul W. Holland. *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: The MIT Press, 1975.

[13] Louis D. Braida. "Development of a model for multidimensional identification experiments." *Journal of the Acoustical Society of America*, 84, S142, 1988.

[14] Louis D. Braida. "Crossmodal Integration in the Identification of Consonant Segments." *The Quarterly Journal of Experimental Psychology*, 43A(3): 647–677, 1991.

[15] Louis D. Braida. "Integration Models of Speech Intelligibility." Presented at the "NAS–CHABA Meeting on Speech Communication Metrics and Human Performance." Washington D.C., June 3–4, 1993.

[16] Louis D. Braida. Personal communication. 1991.

[17] B. Clarke and D. Ling. "The effects of using Cued Speech: A follow-up study." *The Volta Review*, 78: 23–34, 1976.

[18] R. Orin Cornett. "Cued Speech." *American Annals of the Deaf*, 112: 3–13, 1967.

[19] R. Orin Cornett, Robert Beadles and Blake Wilson. "Automatic Cued Speech." Proc. Res. Conf. on Speech-Processing Aids for the Deaf, Gallaudet College, 1977.

[20] P.B. Denes. "On the Statistics of Spoken English." *The Journal of the Acoustical Society of America*, 35(6): 892–904, June 1963.

[21] B. Dautrich, L. Rabiner and T. Martin. "On the effects of varying filterbank parameters on isolated word recognition." *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-31(4): 793–807, Aug. 1983.

[22] S.B. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-28(4): 357–366, Aug. 1980.

[23] P. Delattre, A. Lieberman, and F. Cooper. "Acoustic loci and transitional cues for consonants." *Journal of the Acoustical Society of America*, 27(4): 769–773, Jul. 1955.

[24] Vassilios V. Digalakis, Mari Ostendorf, and Jan R. Rohlicek. "Fast Algorithms for Phone Classification and Recognition Using Segment-Based Models." *IEEE Transactions on Signal Processing*, 40(12): 2885–2897, Dec. 1992.

[25] George R. Doddington. "Phonetically sensitive discriminants for improved speech recognition." *Proceedings of ICASSP-89*, Glasgow, Scotland, pp.556–559, May 1989.

[26] C.G. Fisher. "Confusions among visually perceived consonants." *Journal of Speech and Hearing Research*, 11: 796–804, 1968.

[27] William M. Fisher *et al.* "An acoustic-phonetic database." *The Journal of the Acoustical Society of America*, Suppl.1 (81), p.S92, Spring 1987.

[28] Sadaoki Furui. "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum." *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34(1): 52–59, Feb. 1986.

[29] Oded Ghitza. "Auditory Nerve Representation as a Basis for Speech Processing." in *Advances in Speech Signal Processing*, Sadaoki Furui and M. Mohan Sondhi (eds.), New York: Marcel Dekker, Inc., pp.453–484, 1992.

[30] James R. Glass. *Finding Acoustic Regularities in Speech: Applications to Phonetic Recognition.* M.I.T. Ph.D. Thesis, Cambridge MA, May 1988.

[31] V.N. Gupta, M. Lennig and P. Mermelstein. "Integration of acoustic information in a large vocabulary word recognizer." *Proceedings of ICASSP-87*, pp. 697–700, Dallas, Texas, Apr. 1987.

[32] Hsiao-Wuen Hon, personal communication, ICASSP-91, Toronto, May 17 1991.

[33] X.D. Huang. "Phoneme Classification Using Semicontinuous Hidden Markov Models." *IEEE Transactions on Signal Processing*, 40(5): 1062–1067, May 1992.

[34] X.D. Huang and M.A. Jack. "Semi-continuous hidden Markov models for speech signals." *Computer Speech and Language*, 3: 239–251, 1989.

[35] Xuedong Huang, Kai-Fu Lee and Hsiao-Wuen Hon. "On Semi-Continuous Hidden Markov Modeling." *Proceedings of ICASSP-90*, pp.689–692, Albuquerque, New Mexico, 1990.

[36] X.D. Huang, K.F. Lee, H.W. Hon, and M.Y. Hwang. "Improved Acoustic Modeling with the SPHINX Speech Recognition System." *Proceedings of ICASSP-91*, pp.345–348, Toronto, May 1991.

[37] Pamela L. Jackson. "The Theoretical Minimal Unit for Visual Speech Perception: Visemes and Coarticulation." *The Volta Review*, 90(5): 99–115, Sept. 1988.

[38] F. Jelinek and R.L. Mercer. "Interpolated Estimation of Markov Source Parameters from Sparse Data." in *Pattern Recognition in Practice*, E.S. Gelsema and L.N. Kanal (eds.), Amsterdam: North-Holland, pp.381–397, 1980.

[39] Elizabeth Kipila and Barbara Williams-Scott. "Cued Speech and Speechreading." *The Volta Review*, 90(5): 179–189, Sept. 1988.

[40] L.F. Lamel, R.H. Kassel, and S. Seneff. "Speech database development: Design and analysis of the acoustic-phonetic corpus." in *Proceedings of DARPA Speech Recognition Workshop*, L.S. Baumann (ed.), pp.100–109, Feb. 1986.

[41] C.H. Lee *et al.* "Acoustic modeling for large vocabulary speech recognition." *Computer Speech and Language*, 4: 127–165, 1990.

[42] Kai-Fu Lee. *Automatic Speech Recognition.* Norwell, MA: Kluwer Academic Publishers, 1989.

[43] Kai-Fu Lee and Hsiao-Wuen Hon. "Speaker Independent Phone Recognition Using Hidden Markov Models." *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(11): 1641–1648, Nov. 1989.

[44] Kai-Fu Lee *et al.* "Speech Recognition Using Hidden Markov Models: A CMU Perspective." *Speech Communication*, 9: 497–508, 1990.

[45] Kai-Fu Lee and Sanjoy Mahajan. "Corrective and reinforcement learning for speaker-independent continuous speech recognition." *Computer Speech and Language*, 4: 231–245, 1990.

[46] Y. Linde, A. Buzo and R.M. Gray. "An algorithm for vector quantizer design." *IEEE Transactions on Communications*, COM-28(1): 84–95, Jan. 1980.

[47] R.P. Lippman. "Review of Research on Neural Nets for Speech." *Neural Computation*, 1(1): 1–38, March 1989.

[48] A. Ljolje and M.D. Riley. "Automatic Segmentation and Labeling of Speech." *Proceedings of ICASSP-91*, pp.473–476, Toronto, May 1991.

[49] J. Makhoul, S. Roucos, and H. Gish. "Vector quantization in speech coding." *Proceedings of the IEEE*, 73(11): 1551–1588, Nov. 1985.

[50] J.D. Markel and A.H. Gray. *Linear Predication of Speech.* New York: Springer-Verlag, 1976.

[51] Matthew McGrath and Quentin Summerfield. "Intermodal timing relations and audio-visual speech recognition by normal-hearing adults." *Journal of the Acoustical Society of America*, 70(2): 678–685, Feb. 1985.

[52] R.L. Mercer. "Language Modeling for Speech Recognition." 1988 IEEE Workshop On Speech Recognition, Arden House, Harriman, N.Y. May 1988.

[53] George A. Miller and Patricia E. Nicely. "An analysis of perceptual confusions among some English consonants." *Journal of the Acoustical Society of America*, 27(2): 338–352, Mar. 1955.

[54] Paul Milner, Louis D. Braida, Nathaniel I. Durlach and Harry Levitt. "Perception of Filtered Speech by Hearing-Impaired Listeners." Appendix to Articulation Testing Methods for Evaluating Speech Reception by Impaired Listeners, by Louis D. Braida. *ASHA Reports*, No.14, pp.30–41, 1984.

[55] David P. Morgan and Christopher L. Scofield. *Neural Networks and Speech Processing.* Norwell, MA: Kluwer Academic Publishers, 1991.

[56] National Institute of Standards and Technology. "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus." NIST Speech Disc CD1-1.1, 1990.

[57] Russell J. Niederjohn and Meir Lahat. "A Zero-Crossing Consistency Method for Formant Tracking of Voiced Speech in High Noise Levels." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-33(2): 349–355, Apr. 1985.

[58] Gaye H. Nicholls and Daniel Ling. "Cued Speech and the reception of spoken language." *Journal of Speech and Hearing Research*, 25: 262–269, Jun. 1982.

[59] Douglas O'Shaughnessy. *Speech Communication, Human and Machine*, Reading MA: Addison-Wesley, 1987.

[60] Elmer Owens and Barbara Blazek. "Visemes observed by hearing-impaired and normal-hearing adult viewers." *Journal of Speech and Hearing Research*, 28: 381–393, Sept. 1985.

[61] Prem C. Pandey, Hans Kunov and Sharon M. Abel. "Disruptive Effects of Auditory Signal Delay on Speech Perception With Lipreading." *The Journal of Auditory Research*, 26: 27–41, 1986.

[62] Athanasios Papoulis. *Probability, random variables, and stochastic processes.* New York: McGraw-Hill, 1984.

[63] Douglas B. Paul. "Speech Recognition Using Hidden Markov Models." *The Lincoln Laboratory Journal*, 3(1): 41–61, 1990.

[64] Douglas B. Paul. "The Lincoln Tied-Mixture HMM Continuous Speech Recognizer." *Proceedings of ICASSP-91*, pp.329–332, Toronto, May 1991.

[65] Patrick M. Peterson and Joseph A. Frisbie. "An interactive environment for signal processing on a VAX computer." *Proceedings of ICASSP-87*, pp.1891–1894, Dallas, Texas, Apr. 1987.

[66] Joseph Picone. "Continuous Speech Recognition Using Hidden Markov Models." *IEEE ASSP Magazine*, 7(3): 26–41, July 1990.

[67] L.R. Rabiner and R.W. Schafer. *Digital Processing of Speech Signals*, Englewood Cliffs, N.J.: Prentice-Hall, 1978.

[68] L.R. Rabiner "Tutorial on Isolated and Connected Word Recognition." in *Signal Processing II: Theories and Applications*, H.W. Schüssler (ed.), Elsevier Science Publishers B.V. (North-Holland), 1983.

[69] Lawrence R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE*, 77(2): 257–286, Feb. 1989.

[70] L.R. Rabiner, B.-H Juang, S.E. Levinson, and M.M. Sondhi. "Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities." *AT&T Technical Journal*, 64(6): 1211–1233, Jul.-Aug. 1985.

[71] L.R. Rabiner, S.E. Levinson, M.M. Sondhi. "On the application of vector quantization and hidden Markov models to speaker independent isolated word recognition." *Bell System Technical Journal*, 62(4): 1075–1105, Apr. 1983.

[72] L.R. Rabiner, J.G. Wilpon and F.K. Soong. "High Performance Digit Recognition Using Hidden Markov Models." *Proceedings of ICASSP-88*, pp. 119-122, New York, Apr. 1988.

[73] Research Triangle Institute. Autocuer Project, Technical Status Report. Research Triangle Park, Nov. 7, 1984.

[74] David Sankoff and Joseph B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.* Reading, MA: Addison-Wesley, 1983.

[75] R.W. Schafer and L.R. Rabiner. Design of Digital Filter Banks for Speech Analysis. *The Bell System Technical Journal*, 50(10): 3097–3115, 1971.

[76] R.Schwartz *et al.* "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech." *Proceedings of ICASSP-85*, pp. 31.3.1–4, Tampa, FL, Apr. 1985.

[77] Hugh Secker-Walker and Campbell Searle. "Time-domain analysis of auditory-nerve-fiber firing rates." *Journal of the Acoustical Society of America*, 88(3): 1427–1436, Sep. 1990.

[78] Stephanie Seneff. "A joint synchrony/mean-rate model of auditory speech processing." *Journal of Phonetics*, 16: 55–76, 1988.

[79] K. Stevens, S. Keyser and H. Kawasaki. "Toward a phonetic and phonological theory of redundant features." in *Invariance & Variability in Speech Processes*, J. Perkell and D. Klatt, eds., Hillsdale, N.J.: Erlbaum, 1986.

[80] Quentin Summerfield. "Audio-visual Speech Perception, Lipreading and Artificial Stimulation." in *Hearing Science and Hearing Disorders*, M.E. Lutman and M.P. Haggard eds., New York: Academic Press, 1983.

[81] R.M. Uchanski, L.A. Delhorne, A.K. Dix, L.D. Braida, C.M. Reed, and N.I. Durlach. “Automatic Speech Recognition to Aid the Hearing Impaired. Prospects for the Automatic Generation of Cued Speech.” *Journal of Rehabilitation Research and Engineering*, in press.

[82] Georges Vilaclara. “Recognition of Labial-Doubles for a Substitution Hearing-Aid.” in *Signal Processing III: Theories and Applications*, I.T. Young *et al.* (eds.) Elsevier Science Publishers B.V. (North-Holland), 1986.

[83] A.J. Viterbi. “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm.” *IEEE Transactions on Information Theory*, IT-13(2): 260–269, Apr. 1967.

[84] B.E. Walden, R.A. Prosek and A.A. Montgomery. “Effects of training on the visual recognition of consonants.” *Journal of Speech and Hearing Research*, 20: 130–145, 1977.

[85] Virginia D. Wozniak and Pamela L. Jackson. “Visual vowel and diphthong perception from two horizontal viewing angles.” *Journal of Speech and Hearing Research*, 22: 354–365, Jun. 1987.

[86] V. Zue *et al.* “Recent progress on the SUMMIT system.” in *Proceedings of the Third Darpa Workshop on Speech and Natural Language*, pp.380–384, Jun. 1990.

Appendix A

Vector Quantizer Codebook Generation

This is the procedure that was employed throughout the investigation whenever a VQ codebook was called for. It is a very slightly modified version of the algorithm described in [46] which is commonly used in speech applications.

VQ Algorithm

Assume to have N training vectors $\vec{v}_n$ from which to produce a codebook of M vectors.

1. Initialization:

   Find first centroid: $\vec{c}_1 = \frac{1}{N}\sum_{n=1}^{N}\vec{v}_n$

   Set number of centroids: $K = 1$

   Assign all vectors to set 1.

2. For k between 1 and $2K$ generate markers $\vec{r}$:

$$\vec{r}_k = \begin{cases} \vec{c}_{\frac{k}{2}}(1+\epsilon) & \text{for } k \text{ even} \\ \vec{c}_{\frac{k+1}{2}}(1-\epsilon) & \text{for } k \text{ odd} \end{cases}$$

   where ϵ is a constant $\ll 1$.

   Set K to double its old value.

3. Assign vectors to sets.

   For all n between 1 and N compute:

   $$k_{\min}(n) = \arg \min_{1 \leq k \leq K} (||\vec{v}_n - \vec{r}_k||)$$

   and assign vector $\vec{v}_n$ to set $k_{\min}(n)$.

4. Compute average distance of vectors to markers:

   $$D_A = \frac{1}{N} \sum_{k=1}^{K} \sum_{\substack{n \text{ s.t.} \\ \vec{v}_n \in \text{ set } k}} ||\vec{v}_n - \vec{r}_k||$$

5. Find centroids of all sets.

   For all k between 1 and K:

   $$\vec{c}_k = \frac{1}{\#\text{ of vec. in set } k} \sum_{\substack{n \text{ s.t.} \\ \vec{v}_n \in \text{ set } k}} \vec{v}_n$$

6. Compute average distance to centroids:

   $$D_B = \frac{1}{N} \sum_{k=1}^{K} \sum_{\substack{n \text{ s.t.} \\ \vec{v}_n \in \text{ set } k}} ||\vec{v}_n - \vec{c}_k||$$

7. Check convergence, where δ is a preset convergence constant:

   If $\left|\frac{D_A - D_B}{D_A}\right| < \delta$ then for all k between 1 and K set $\vec{r}_k = \vec{c}_k$ and go to step 3; otherwise continue.

8. If $K = M$ stop: codebook consists of the vectors $\vec{c}_k$ for $1 \leq k \leq M$; otherwise go to step 2.

The modification to the algorithm of [46] occurs in step 2; in the original procedure, a fixed perturbation vector $\vec{\epsilon}$ was added and subtracted when generating the two markers from each prototype vector.

Appendix B

Fitting of Data to an Unsaturated Log-linear Model

The following is the iterative procedure used in Section 5.3.3 to compute maximum likelihood estimates of expected cell count values in a contingency table. It follows the method given in [12, Chap. 3]. The tables employed in this work refer exclusively to arrays holding the number of occurrences of all possible sub-recognizer output combinations for a given spoken phone and training data. The algorithm given below applies to four-dimensional tables but generalizes in an obvious way to tables with any number of coordinates.

As before, x_{ijkl} refers the actual count observed and $\hat{m}_{ijkl}$ is the ML estimate of the expected value that we want to calculate. Furthermore, we are assumed to have picked some unsaturated log-linear model of the table's structure, i.e. Equation 5.8 with high order terms set to zero.

Algorithm

1. Find the sufficient statistics. These will comprise marginal sums of the full table. They can be identified by inspection from the unsaturated model. For each u-term the candidate sufficient statistic is the marginal sum with the same subscripts as the u-term, for instance: $u_{23(jk)} \rightarrow x_{+jk+}$. Eliminate all candidate

marginal sums that can be derived from another candidate sum by adding across one of the latter's subscripts. The remaining S sums are the sufficient statistics we need to fit. A particular sum configuration is designated as $\mathcal{C}_s$, thus x_{ij++} might be $x_{\mathcal{C}_1}$.

2. Call the value of the estimates after the nth iteration $\hat{m}_{ijkl}(n)$. To set all $\hat{m}_{ijkl}(0)$ (i.e. to initialize the iteration) any initial values that do not exhibit the effects that had been set to zero in the model may be used. A convenient initial value is 1.

3. Set the cycle number r to zero and enter the iteration.

4. For all sufficient configurations compute the updated estimate of the expected cell count value:

$$\hat{m}_{ijkl}(rS+s) = \hat{m}_{ijkl}(rS+s-1)\frac{x_{\mathcal{C}_s}}{\hat{m}_{\mathcal{C}_s}(rS+s-1)}$$

   where s takes values from 1 to S.

5. Find the maximum change in any cell from previous cycle:

$$\Delta m_{\max} = \max_{\{i,j,k,l\}} |\hat{m}_{ijkl}((r+1)S)) - \hat{m}_{ijkl}(rS)|$$

6. If $\Delta m_{\max}$ is greater than a preset δ then increment r and go to step 4; otherwise the estimates $\hat{m}_{ijkl}((r+1)S)$ are final.

In this work the initial value of 1 was indeed used. The stopping rule was modified, however. Having to store the entire array of the estimated values from the previous cycle required over 21 megabytes of memory beyond that already used for the calculations. It was found that simply fixing the number of cycles (r) could be used instead. In general, five such cycles were run.

Appendix C

Alignment of Original and Output Phone Sequences

The alignment between the TIMIT transcription of a sentence and the sequence of phones produced by the recognizer was accomplished through a dynamic programming string matching algorithm given in [74]. This algorithm is outlined below.

We assume to have two phone sequences: the "true" or input sequence, $\vec{q}^M = [q_1 \ldots q_M]$ and the recognized or output, $\vec{r}^N = [r_1 \ldots r_N]$. The lengths M and N in general will be different. The objective is to produce a pair string (the alignment) matching each of the phones from the input with one from the output. The insertion/deletion symbol ϕ may be included in either string and matched to a phone occurring in the other. Thus the pair string will be of the form $[a_1 b_1, a_2 b_2, \ldots, a_P b_P]$ with a and b comprising the strings q and r, respectively, and the insertion/deletion symbol.[1]

The alignment is determined by specifying an array of *distances* between all input and output symbols $d(x, y)$. The algorithm then computes the alignment that minimizes the total (additive) distance. It proceeds by aligning substrings and then extending them. The total distance between any two substrings $\vec{q}^i$ and $\vec{q}^j$ is denoted

[1] As far as the matching algorithm is concerned, a deletion in one string is equivalent to an insertion in the other. The algorithm disallows matching an insertion in the input string with a deletion in the output string.

D_{ij}.

Initialize: $D_{00} = 0$ and $D_{ij} = \infty$ if i or j is negative.

Recursion:

Find the optimal partial alignment up to the current position in each string:

$$D_{ij} = \min \begin{cases} D_{i-1,j} + d(q_i, \phi) & \text{deletion} \\ D_{i-1,j-1} + d(q_i, r_j) & \text{substitution/match} \\ D_{i,j-1} + d(\phi, r_j) & \text{insertion} \end{cases}$$

We also record the pointers to the best predecessor cell, i.e. respectively to which of the terms above was minimum we have:

$$\text{pointer}_{ij} = \begin{cases} (i-1, j) & \text{deletion} \\ (i-1, j-1) & \text{substitution/match} \\ (i, j-1) & \text{insertion} \end{cases}$$

Solution: The optimal distance after the recursion is D_{MN}. The pair string is recovered by backtracking along the pointers saved during the recursion.

Table C.2 is the matrix of distances between individual phones that was used in the alignment. Distance for a match was zero, for deletions and most substitutions it was three. Distance between phones that are collapsed when forming the 39-element alphabet (see Sec. 6.1) was set to one. Because we also wanted to obtain meaningful confusion matrices from the final output, some of the substitution weights, between what were considered easily confusable phones, were set to two: for instance, all stop consonants. This helped the alignment, especially when a deletion occurred from a vowel-consonant sequence in the input. Insertion penalty was set to one after some experimentation. This distance, combined with that for a deletion, seemed to produce most consistently useful alignments.

Table C.1 shows alignment examples for three sentences of varying recognition accuracy. There are essentially no problems aligning the two higher-scoring utterances. One potential error in the third sentence occurs at input frame 50 where "ah", in the input, is matched with "ow" in the output. It should have probably been aligned with the preceding output "aa" and "ow" should have been counted as an insertion. Such mis-alignments seemed fairly rare. They also had virtually no effect on the resulting overall score.

Sentence 1				Sentence 2				Sentence 3			
TIMIT		Recognized		TIMIT		Recognized		TIMIT		Recognized	
0	sil	sil	0	0	sil	sil	0	0	sil	sil	0
14	b	b	12	-	*ins*	k	13	-	*ins*	dh	12
15	aa	ao	16	15	ah	ah	18	15	iy	iy	15
27	r	er	30	22	vcl	vcl	22	21	vcl	ng	18
34	vcl	vcl	34	35	b	b	25	26	g	k	21
42	b	b	37	-	*ins*	l	36	30	w	w	30
43	er	er	44	36	ao	ao	44	35	aa	aa	34
54	n	n	54	47	r	r	51	47	n	m	47
63	vcl	vcl	62	56	ih	iy	59	-	*ins*	aa	50
71	p	p	66	66	ng	ng	64	50	ah	ow	57
76	ey	ey	75	73	n	n	68	62	z	s	63
87	cl	cl	86	79	aa	aa	78	72	ix	ah	72
93	p	k	92	98	v	*del*	-	-	*ins*	dx	76
95	er	er	95	102	el	l	98	78	n	er	79
107	en	m	107	117	ix	ih	120	79	ae	v	82
117	l	ix	117	130	z	f	128	98	l	aw	85
122	iy	ey	120	135	ix	uw	134	101	ax	ao	99
132	v	n	133	141	s	z	139	107	vcl	vcl	106
136	z	z	137	154	uh	ix	155	111	g	p	111
144	ix	ix	144	159	cl	cl	159	116	ey	y	116
148	n	n	147	170	p	p	169	130	dx	*del*	-
152	ah	ah	152	178	er	er	178	132	er	er	129
-	*ins*	n	156	197	vcl	vcl	196	142	z	z	140
157	vcl	vcl	159	204	b	*del*	-	149	er	ax	149
169	b	b	162	207	s	s	205	156	cl	cl	154
170	ih	ih	171	-	*ins*	t	216	163	t	k	162
179	vcl	vcl	177	221	l	w	221	172	r	r	170
192	b	b	180	225	iy	iy	225	176	aa	ay	173
193	aa	ao	194	232	cl	cl	231	184	cl	*del*	-
210	n	n	209	237	p	b	235	190	p	*del*	-
214	f	f	215	240	ix	iy	240	194	cl	cl	183
225	ay	aa	223	244	ng	ng	244	198	k	k	198
235	er	er	231	250	cl	cl	252	-	*ins*	ow	202
243	sil	sil	242	258	p	k	257	203	el	v	211
				262	ih	ix	262	212	r	r	216
				270	l	l	267	220	eh	*del*	-
				274	sil	sil	275	227	cl	cl	225
								239	t	k	239
								244	ay	aa	244
								-	*ins*	cl	264
								259	l	k	267
								270	s	s	272
								286	sil	sil	286
				48-phone Percent Correct							
72.7				55.9				39.5			
				39-phone Percent Correct							
78.8				64.7				39.5			

Table C.1: Matching of true (TIMIT) and recognized phone sequences of three sentences, as performed by the alignment algorithm. Cepstrum coefficients and occurrence-weighted interpolation used in recognition. Numbers columns show the frame segmentation of the sequences. The text of the sentences is: Sent. 1 - "Barb burned paper and leaves in a big bonfire", Sent. 2 - "A boring novel is a superb sleeping pill", Sent. 3 - "Iguanas and alligators are tropical reptiles".

Input	Output Phone																								
Phone	iy	ih	eh	ae	ix	ax	ah	uw	uh	ao	aa	ey	ay	oy	aw	ow	l	el	r	y	w	er	m	n	en
iy	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	2	3	3	3	3	3
ih	2	0	2	2	1	2	2	2	2	2	2	2	2	2	2	2	3	3	3	2	3	3	3	3	3
eh	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	2	3	3	3
ae	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
ix	2	1	2	2	0	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
ax	2	2	2	2	2	0	1	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
ah	2	2	2	2	2	1	0	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
uw	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
uh	2	2	2	2	2	2	2	2	0	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
ao	2	2	2	2	2	2	2	2	2	0	1	2	2	2	2	2	3	3	3	3	3	3	3	3	3
aa	2	2	2	2	2	2	2	2	2	1	0	2	2	2	2	2	3	3	3	3	3	3	3	3	3
ey	2	2	2	2	2	2	2	2	2	2	2	0	2	2	2	2	3	3	3	2	3	2	3	3	3
ay	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	2	3	3	3	3	3	3	3	3	3
oy	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	3	3	3	3	3	3	3	3	3
aw	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	3	3	3	3	3	3	3	3	3
ow	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	3	3	3	3	3	3	3	3	3
l	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	1	2	2	2	3	3	3	3
el	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	0	2	2	2	3	3	3	3
r	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	3	3	2	3	3	3
y	2	2	3	3	3	3	3	3	3	3	3	2	3	3	3	3	2	2	2	0	2	3	3	3	3
w	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	0	3	3	3	3
er	3	3	2	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	2	3	3	0	3	3	3
m	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	2	2
n	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	0	1
en	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	1	0
ng	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	2	2
ch	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
jh	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
dh	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
dx	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
b	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
d	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
g	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
p	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
t	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
k	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
z	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
zh	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
v	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
f	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
th	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
s	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
sh	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
hh	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
cl	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
vcl	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
epi	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
sil	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
in	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table C.2: Phone distances for the alignment algorithm; "in" means insertion, "del" deletion. Continued on next page.

Output Phone																								Input
ng	ch	jh	dh	dx	b	d	g	p	t	k	z	zh	v	f	th	s	sh	hh	cl	vcl	epi	sil	del	Phone
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	iy
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ih
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	eh
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ae
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ix
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ax
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ah
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	uw
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	uh
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ao
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	aa
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ey
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ay
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	oy
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	aw
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ow
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	l
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	el
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	r
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	y
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	w
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	er
2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	m
2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	n
2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	en
0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	ng
3	0	2	3	3	3	3	3	3	2	3	3	3	3	3	3	3	2	3	3	3	3	3	3	ch
3	2	0	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	jh
3	3	3	0	2	2	2	2	3	3	3	3	3	2	3	3	3	2	3	3	3	3	3	3	dh
3	3	3	2	0	3	3	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	dx
3	3	3	2	3	0	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	b
3	3	3	2	3	2	0	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	d
3	3	3	2	3	2	2	0	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	g
3	3	3	3	3	2	2	2	0	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3	p
3	3	3	3	2	2	2	2	2	0	2	3	3	3	3	3	3	3	3	3	3	3	3	3	t
3	3	3	3	3	2	2	2	2	2	0	3	3	3	3	3	3	3	3	3	3	3	3	3	k
3	3	3	3	3	3	3	3	3	3	3	0	2	3	3	3	2	2	3	3	3	3	3	3	z
3	3	2	3	3	3	3	3	3	3	3	2	0	3	3	3	2	1	3	3	3	3	3	3	zh
3	3	3	2	3	3	3	3	3	3	3	3	3	0	2	2	3	3	3	3	3	3	3	3	v
3	3	3	3	3	3	3	3	3	3	3	3	3	2	0	2	2	3	3	3	3	3	3	3	f
3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	0	2	3	2	3	3	3	3	3	th
3	3	3	3	3	3	3	3	3	3	3	2	2	3	2	2	0	2	3	3	3	3	3	3	s
3	3	3	3	3	3	3	3	3	3	3	2	1	3	3	3	2	0	3	3	3	3	3	3	sh
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	3	3	0	3	3	3	2	3	hh
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	0	1	1	1	3	cl
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	0	1	1	3	vcl
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1	1	0	1	3	epi
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	1	1	1	0	3	sil
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NA	in

Table C.3: Continued distance table.

Appendix D

Confusion Matrices

The following are selected confusion matrices showing detailed performance of various versions of the recognizer. The confusions are given for the final output on the full 48-element alphabet defined in Table 4.1. Entries represent *ten times* the percentage[1] of vertically listed phones identified as a given horizontally listed phone. Since percentages lower than 1 are not listed, row quantities may not add up to 1000.

[1] This unconventional notation was required to fit the table on one page: it saves the decimal point while preserving the precision.

Table D.1: Confusion matrix for cepstrum parameters, occurrence-weighted interpolation, SX+SI estimation sentences.

Input	Output Phone																																																	Input
Phone	iy	ih	eh	ae	ix	ax	ah	uw	uh	ao	aa	ey	ay	oy	aw	ow	l	el	r	y	w	er	m	n	en	ng	ch	jh	dh	dx	b	d	g	p	t	k	z	zh	v	f	th	s	sh	hh	cl	vcl	epi	sil	del	Phone
iy	660	35			68			32				58								28																													45	iy
ih	92	294	114	21	131	39	18	31				65				19				12		27																											87	ih
eh		78	316	123	45	24	50			11	15	56	29		11	26			11			36																											107	eh
ae	17	35	146	413	61		23				42	29	56		23							10																											77	ae
ix	57	100	31	15	424	67	14	40	10			24										27																											119	ix
ax		30	33		141	335	39	11	11	31	16					39	11		10			26		10																									176	ax
ah		16	116	88	25	86	172		11	54	104		63		20	57	23		23			25																											70	ah
uw	136	47	18		136	47		391								18	27					15																											77	uw
uh	16	124	62		85	171	85	39	78	47		16	16			54	31					47																											85	uh
ao						50	10			465	140		12	16		32	50	12			32																												94	ao
aa			11	23		12	53			216	401		102		27	18						14																											66	aa
ey	129	53	48	30	46							543	39	21		11																																	25	ey
ay			28	47		26	33			14	118	35	586		14	17																																	31	ay
oy	21		42	42	21	52	10		10	125	21	104	73	302		31	62		10		21	10																											42	oy
aw				105	24	24	56			48	234		56		290	56																																	24	aw
ow		26	65	31	17	77	43	11		88	26	14	14		11	298	43	31				11																											111	ow
l						10				10						10	467	56	48		92		22	10					14	10																			159	l
el						49				15						29	299	407	15		54			10		10				10														10					59	el
r																			530			193																											167	r
y	111	15						11									27		34	374				19				15	34	19													11	23					218	y
w										13							133		24		611										15																		101	w
er		20	26		31	14													194			568		10																									45	er
m																			14		27		532	254		17																							65	m
n																	14		12				111	567	18	38			13	14																12			118	n
en					26	35																	139	435	200														17								26		43	en
ng					19	10																	48	396	10	329																		14		24			87	ng
ch																											500	196							114							49	87						27	ch
jh																											151	486				24	24	14	80	19	33					33	47						24	jh
dh																	11						33	20					302	15	88	81	15	59	20	20			52	11	18								177	dh
dx																	27		15				18	69					39	304	33	36							51							18			290	dx
b																							14						69		559	74	18	102	26	23													86	b
d																								11				26	73		112	256	57	44	152	55			13										117	d
g																			10	10			10						19		84	113	331	45	48	222			10										68	g
p																													56		68	17	14	465	107	143								15					80	p
t																											24	19	24		16	54	10	63	490	118						14		11					114	t
k																																19	28	70	114	622													85	k
z																																					504	10		26		334	22						23	z
zh	18																			18							53	70									246	228				53	298						18	zh
v					13												15		10		18		38	89					13	20	26	15					23		306	69	20	18			13	31			191	v
f																																		20			15		13	726	40	68		13			11		37	f
th				13	13																								45		25	19		32	57	13	38		45	159	159	127	13		115	13			70	th
s																											13	10									125			38		738	26						21	s
sh																											28	20									39	24				169	685						12	sh
hh												12					20						12	12					24	16				72	28	76			24	20			12	398			16		159	hh
cl																																													776	111	10	15	64	cl
vcl																																													199	614		13	108	vcl
epi																								19													14			14					74	46	537	32	204	epi
sil																																													18			956		sil

Table D.2: Confusion matrix for LPC parameters, occurrence-weighted interpolation, SX+SI estimation sentences.

Input	Output Phone																																																Input
Phone	iy	ih	eh	ae	ix	ax	ah	uw	uh	ao	aa	ey	ay	oy	aw	ow	l	el	r	y	w	er	m	n	en	ng	ch	jh	dh	dx	b	d	g	p	t	k	z	zh	v	f	th	s	sh	hh	cl	vcl	epi	sil	del	Phone
iy	651	35			70			36				63								33																														53	iy
ih	82	300	92	17	164	28	17	41	10			62	12			24				14		27																													85	ih
eh	11	92	290	119	36	23	72		12	20	24	42	20		12	20			12			41																													107	eh
ae	13	31	123	412	73	15	23				40	27	59		36	13			10																																		75	ae
ix	61	83	35	22	409	65	15	41	13			21										37																													122	ix
ax		25	30		149	315	46			31	21					26	14	16				20																													199	ax
ah		11	109	81	25	97	181			59	122		61		18	50	16					23																													77	ah
uw	103	59			127	50		372	12	12		12				15						27																											12		100	uw
uh	23	147	62		147	202	101	39	78		23												47	16																												62	uh
ao			10		12	36	16			433	137		22	24		22	36	14			28	12																													131	ao
aa			14	14		16	46			201	414		103		39	12	11																																	69	aa
ey	140	62	34	46	53							529	28	16									16																													34	ey
ay			12	57	28	28	28			12	137	38	530		14								12																													40	ay
oy	52	21	31		31	31	10			125	10	73	104	292	10	21	73	10	10			31																													62	oy
aw			73	89		40	40			40	169		32		331	32			24																									16	16				56	aw
ow	11	14	43	31	40	80	60			68	43		23		11	281	37	28			14																														136	ow
l						11										15	491	56	25		79		14	10																											192	l
el						44	10	10		15						44	254	429	24		73																														44	el
r																			541			177		14																											159	r
y	99				11												23		27	389	15			19				15	46										11			11		19					221	y
w																	125		43		607											24																		105	w
er		21	17		32	13													165			586		10																												57	er
m																	18		21		17		537	198	14	14			16		10																			91	m
n																			14				110	560	18	38				19																					137	n
en					35	17																	139	409	191	52																									87	en
ng					29														10		10		77	348	10	343				14												10					10			97	ng
ch																											511	223							103							43	49						54	ch
jh					14																							142	542				19		14	57		28						52	42						33	jh
dh					11																	17	20					322	20	74	55	13	57	46	11	13		39		17								219	dh
dx		12															15		15				27	60					39	315	21	33							57					18		18			292	dx
b																													51		568	83	14	128	19																93	b
d																									11				13	73	20	117	249	62	53	141	37			18									150	d
g																							13	13					13		119	83	369	54	32	170			10									87	g
p																													53		90	20	10	447	107	145								14					70	p
t																											33	16	28		16	37		83	468	104							12		14					115	t
k																																21	38	82	118	597				11									78	k
z																												10								499			12		343	23						29	z
zh	18																				35								88								140	158				88	456		18					zh
v					10	10											15				18	18	41	69						31	20	23	15					10		342	82	26	13			23				163	v
f																																		15					18	713	44	86			11				38	f
th					19																	13								57		25	19		57	32	25	25		25	178	172	127		32	51				96	th
s																																						120			39		732	25						20	s
sh																											39										39	12				161	702						20	sh
hh					12													12		16	20				16							36	16				52	36	68			12	24	12			394	12			12	151	hh
cl																																														773	113		17	66	cl
vcl																								12																						201	610		14	107	vcl
epi																				14																								19			93	56	491	37	208	epi
sil																																														17	11		948		sil

Table D.3: Confusion matrix for autocorrelation parameters, occurrence-weighted interpolation, only SX estimation sentences.

Input	Output Phone																																																	Input
Phone	iy	ih	eh	ae	ix	ax	ah	uw	uh	ao	aa	ey	ay	oy	aw	ow	l	el	r	y	w	er	m	n	en	ng	ch	jh	dh	dx	b	d	g	p	t	k	z	zh	v	f	th	s	sh	hh	cl	vcl	epi	sil	del	Phone
iy	596	59			76			30				67								28				10																									53	iy
ih	92	263	80	26	127	39	27	32	22			89	12			23				14		24																											85	ih
eh		104	289	95	47	24	65			20	26	39	23		12	26			11			60																											113	eh
ae	29	29	146	355	61	10	29			17	38	40	63		25							10																											71	ae
ix	54	85	30	11	398	77	10	29				22				11			13			27																											135	ix
ax		31	33		182	290	35	21		28	23		10			24		14	13			26																											176	ax
ah		20	109	86	36	109	159			54	122		66		14	45	16					14														10													91	ah
uw	178	56		12	166	33		349								12		12			21	18																											71	uw
uh		116	62	16	116	163	93	39	62	54	23					47	23					31	16																										85	uh
ao			10		26	42	26			376	143		20	18	20	40	34				34																												131	ao
aa			18	34		30	52			180	390		96		32		12														11																		78	aa
ey	120	87	46	41	57							467	28	23		11						28																											39	ey
ay			31	40	38	43	59			38	97	38	478		24	17			17																														38	ay
oy	21	42	73		21	21		21	10	94	42	94	115	208	21	10	21	10	10		31	42								10		10		10															62	oy
aw			32	169	24	65	113			81	153		56		177	24						16																											32	aw
ow		14	34	20	45	80	54	14		88	17		23		20	284	54	14			14	23		17																									128	ow
l										11						10	444	46	44		87		20	26					11										14										164	l
el		10			15	34				39						20	275	377	15		49	15	15	25													10												59	el
r																	23		475		14	172																											193	r
y	149	27			19												31		23	328	23			11				15	31	23		11			11							15		19					218	y
w																	93		52		626		13																										127	w
er	10	18	25		41	20						16							186			524		11																									53	er
m																	26		13		26		452	282	14	20			13										14										79	m
n																			12			12	116	553	24	39				20									12										114	n
en					35	35															17	17	87	400	200	35																							122	en
ng					10												10					14	106	440	14	232																				10			121	ng
ch				11																							560	152							120							54	76						16	ch
jh																											123	571				52			28		38					19	61						38	jh
dh																							17	28					304		112	66	13	52	20	24			35	13	13								203	dh
dx																	27		12	21		12		45					36	439		30							18					12		24			236	dx
b																	11												65		585	70	25	93	11				16										81	b
d																								11					70	20	101	322	55	57	77	57			20										143	d
g																								10					29		74	177	350	68	26	132			16									10	61	g
p																													34		99	27	19	474	72	136								15					77	p
t																											16	19	29		10	60		53	488	132				10		10							114	t
k																																19	33	94	135	569													89	k
z																												16							12		520		10	13		334	25						19	z
zh					18															18							35	53									228	18				105	509						18	zh
v					10	15											13				13		28	64					54	10	23	20					13		334	77	10	15		18	13	26			166	v
f																																		11	13		13		18	711	38	82							29	f
th					25	13																							45		25	13		51	51	25	13		25	204	166	108		19	57	13	13		108	th
s																											17	10									116			48	13	704	41						27	s
sh																											31										47					173	724							sh
hh																	24				12		12	16					16	12				68	16	52	12		12	12	16			454				16	159	hh
cl																																													791	98		12	66	cl
vcl																																													158	649		17	111	vcl
epi						14																															19								74	56	565		199	epi
sil																																													16			951		sil

Table D.4: Confusion matrix for the hybrid system, occurrence-weighted interpolation, only SX estimation sentences.

Input	Output Phone																																																		Input
Phone	iy	ih	eh	ae	ix	ax	ah	uw	uh	ao	aa	ey	ay	oy	aw	ow	l	el	r	y	w	er	m	n	en	ng	ch	jh	dh	dx	b	d	g	p	t	k	z	zh	v	f	th	s	sh	hh	cl	vcl	epi	sil	del	Phone
iy	630	49			66			27				59								32				13																									62	iy
ih	73	290	121	10	141	32	19	45	17			62				19				15		23																											85	ih
eh	12	90	309	125	36	17	78			17	17	45	20		11	18						44																											110	eh
ae	12	31	132	424	65	12	17				38	35	54		23							13																									13		63	ae
ix	55	100	29	16	425	74	18	35	10			24							14			31																											109	ix
ax		31	25		153	326	50	16	13	31	20					33	11					25																											168	ax
ah		25	118	86	41	93	204			59	98		68		16	52	14		11			16																											52	ah
uw	133	65			142	53		396				12				18	12																																74	uw
uh	31	109	47		109	171	70	39	85	31		16				85	23					31																	16										101	uh
ao					12	50	18			439	146		18	18	10	40	44	16			22																												100	ao
aa			29	23		16	50			203	415		75		32	20						11																											68	aa
ey	120	69	60	48	39							510	25	18								18																											32	ey
ay			21	64	28	24	31			47	125	38	534		12																																		19	ay
oy	52	21	52	31	21	31	31	10		104	21	104	62	240	10	31	42				31	21								10																			73	oy
aw			40	113	16	32	32			73	194		56		323	16																												16					40	aw
ow		20	54	23	31	102	57			91	23	20	14		11	290	62	37			11			11																									74	ow
l						10										12	489	47	36		80		16	19																									180	l
el						78				25						44	240	422	15	10	39	10		10		10											10												49	el
r																	11		514			196																											160	r
y	126							11									15		38	408			11					15	11	11	15	11					11					15							233	y
w																	129		37		645		13								11																		84	w
er		19	20		28	17		10				10							181			566		18																									45	er
m																	10		12		29		504	239	18	22			17										13										85	m
n																	15		12			12	107	558	15	42				22									14										116	n
en					35	35																	96	417	226	43													26										78	en
ng	14					10											14						87	367	19	295						10														14			135	ng
ch																											495	223							82							60	87						33	ch
jh																											127	524			14	38	14		57		42					28	52						52	jh
dh																							28	33					276	18	92	87	15	55	28	28			31		18					13			197	dh
dx			12														27		12	12	15		18	69					30	364	18	18					15		36							33			263	dx
b																							14						51		582	81	23	97	18	18													76	b
d																											11	18	57	26	126	300	44	55	101	51			13										137	d
g		10																		13	10		13					13	19		80	113	293	48	48	222			13										64	g
p																													49		89	31	15	475	75	133								10					78	p
t																											27	20	31		12	54		63	480	121				10		10		12					116	t
k																												11				13	32	81	107	595													95	k
z																												19									522			12		338	19						17	z
zh	18	18																									35	53									281	53				70	421						53	zh
v					13												20		10				41	48					18	38	38								357	59	15	28			13	36	10		179	v
f																																		22					16	722	64	69							37	f
th																													51		32	19		51	76	25	19		32	159	191	127		19	45	19			83	th
s																											18	11									125			34		714	31						28	s
sh																											39	24									55					138	705							sh
hh	12																20		32	12	16			16					36	32				56	36	52			12	40		12		414					139	hh
cl																																													771	118		12	64	cl
vcl																								12																					198	612		12	116	vcl
epi					19																																								74	46	537	37	208	epi
sil																																													24			948		sil