DURATION NORMALIZATION FOR ROBUST RECOGNITION OF SPONTANEOUS SPEECH VIA MISSING FEATURE METHODS

Jon P. Nedel

Thesis Committee:

Richard M. Stern, Chair

Tsuhan Chen

Jordan Cohen

B. V. K. Vijaya Kumar

Submitted to the Department of Electrical and Computer Engineering

in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at

Carnegie Mellon University

Pittsburgh, Pennsylvania 15213

April 2004

Amen!
Blessing and glory and wisdom and thanksgiving and honor and power and might
be to our God forever and ever!
Amen.

— Revelation 7:12 —

Abstract

Accurate recognition of spontaneous speech is one of the most difficult problems in speech recognition today. When speech is produced in a carefully planned manner, automatic speech recognition (ASR) systems are very successful at accurate recognition and transcription. In response to casual speech, ASR systems produce more than twice as many errors compared to recognition of the same speech read carefully.

In this thesis, we have developed a practical algorithm to improve the recognition accuracy of ASR systems when transcribing spontaneous speech. We have found that normalizing the speech features so that every sound unit ("phone") has the same duration allows speech recognition models to characterize and recognize speech more accurately.

ASR systems use hidden Markov models (HMMs) to model the sound units from which speech signals are composed. It is well known that HMMs do not accurately model the average phonetic variation or the variability introduced into these durations by the casual production of speech. By normalizing the duration of every speech sound unit, we are eliminating a source of variability in the modeling of speech that can contribute to increased word recognition errors.

When the boundaries between sound units are known *a priori*, the duration normalization approach is able to achieve substantial improvements in recognition accuracy. Automatic identification of unknown boundary locations, however, has proven to be a difficult problem. When speech is highly spontaneous, there is often little or no acoustic evidence in the speech signal to indicate transitions from one sound unit to the next. Duration normalization depends on accurate boundary locations, and even our most accurate automatic segmentation technique when applied in isolation is not sufficiently accurate for duration normalization to perform effectively.

Because our efforts to improve automatic segmentation of spontaneous speech have not been very fruitful, we have focused on the development of duration normalization approaches that are more robust to boundary detection errors. We have also explored the use of duration normalization based on probabilistic identification of phone boundaries. Our most effective system makes use of three simple variants of duration normalization and an algorithm that can combine multiple recognition hypotheses into a single best hypothesis. With this multi-pass approach, we have achieved significant improvements in recognition accuracy by applying duration normalization to a variety of spontaneous speech databases, including a large-scale broadcast news corpus. These techniques achieve a relative reduction in word error rate of 3.9%–7.7%, depending on the size and complexity of the recognition task.

Acknowledgements

Simply stated, there is no way I could have ever finished this thesis on my own.

First and foremost I want to thank God for the wonderful opportunities and the wonderful challenges He has given me. Words cannot express my love for You. The long Ph.D. process has strengthened my faith and helped me to grow in ways that I never would have imagined.

Many thanks to my advisor and friend, Richard Stern, for your patience, encouragement, and creativity throughout this thesis work. Thank you for sticking by me and pressing me to do my best and never give up. Thank you also for the many meetings after sleepless nights and long days of work. Your perseverance and energy always amaze me.

Thank you to my faithful committee: Drs. Jordan Cohen, Tsuhan Chen, and Vijaya Kumar. Thank you for your willingness to serve on my committee and your flexibility as we scheduled (and rescheduled) my defense.

Special thanks also to my colleagues in the Robust Speech Group: Rita and Bhiksha, your great minds for research are only superceded by the kindness of your hearts. I thank you for the countless fruitful discussions about my research and all of your helpful advice as I put this thesis together. Mike and Xiang, your thoughtful advice and friendship are greatly appreciated. Thanks to Matt, Sam-Joo, Evandro, and Juan for your help throughout the years.

To my friends who prayed for me, and supported me every time things felt impossible, I cannot begin to express my gratitude. I pray that God will bless you a hundred fold for the undeserved love, support, and kindness you have shown to me. Thank you.

And last but certainly not least, I want to thank my family for their immense love and thoughtful encouragement. Mom, Dad, and Carly, thank you for believing in me and always doing your best to lend a helping hand when I needed it. You have all sacrificed so much for my sake, and I am forever grateful.

Table of Contents

List of Figures

List of Tables

1: Introduction:
Normalizing Durations to Improve Spontaneous Speech Recognition

Accurate recognition of spontaneous speech is one of the most difficult problems in speech recognition today. In this thesis, we have proposed and developed a technique to normalize the incoming speech feature sequence so that every sound unit ("phone") has the same duration. By normalizing the speech features in such a manner, speech recognition models are better able to characterize the relevant information found in speech signals, especially when the speech is highly spontaneous.

In this chapter, we present a brief introduction to the problem of modeling and recognizing spontaneous speech. We close this chapter with an overview of the thesis document which presents our duration normalization technique in its entirety.

1.1 Improving the Recognition of Spontaneous Speech: A Challenging Task

When speech is produced in a carefully planned manner (*e.g.* the speech of a broadcast news anchor), automatic speech recognition (ASR) systems are very successful at accurate recognition and transcription. The performance of ASR systems in response to casual speech produces more than twice as many errors compared to the recognition of the same speech read carefully.

In order for speech recognition technology to be viable and useful in everyday applications (*e.g.* meeting transcription, telephone-based systems), we need to develop methods to improve recognition accuracy on spontaneous conversational speech. The objective of this thesis is the development of a practical algorithm to improve the strength and robustness of core speech recognition technology when it is applied to transcribe spontaneous speech.

There are many factors that contribute to the difficulty of automatically recognizing spontaneous speech. One of the main difficulties is caused by the variation in duration of the examples used to train recognition models for a given sound unit ("phone"). In spontaneous speech, the duration varies greatly each time a sound is produced. In contrast, the duration variation in carefully-enunciated speech is not as severe. When the training examples for a given sound class vary greatly in duration, it is difficult for an ASR system to properly model that class. When the underlying sound units are modeled poorly, the overall ASR system accuracy degrades.

Our strategy in this thesis is to reduce the duration variability of the tokens used to train an ASR system in order to improve the accuracy when recognizing spontaneous speech. Our earliest attempts to combat the duration variability problem included the idea of mapping spontaneous sound durations back to their

carefully-read counterparts prior to recognition. In the end, we found that normalizing the duration of all sound units to a common duration provided a simple and effective method for improving ASR accuracy when speech is highly spontaneous.

1.2 Thesis Overview

Chapter 2 begins with a review of speech recognition technologies that are relevant to this research. It also contains a review of related research in explicit phone duration modeling in ASR systems. Chapter 3 contains a brief overview of the SPHINX-III recognition system and the speech corpora used in this research.

The specific details of our duration normalization technique are presented in Chapter 4. Results indicate we can successfully improve recognition accuracy on both spontaneous and carefully enunciated speech if we know the locations of the boundaries that separate the underlying sound units. In Chapter 5, we address the difficult problem of blind derivation of consistent and accurate phone boundaries. We explored and evaluated a variety of automatic segmentation techniques and found that segmentation errors have a have a strong impact on duration normalized recognition accuracy.

In Chapter 6, we detail modifications and extensions of the duration normalization algorithm designed to cope with the imperfections in automatically-derived segmentations. In Chapter 7, we present a “soft” re-formulation of the duration normalization algorithm that can make use of probabilistic segmentation information. We close the thesis in Chapter 8 with ideas for future work and conclusions drawn from this research.

2: An Overview of Speech Recognition and Related Research

This chapter presents basic background information relevant to the thesis. We start with a brief overview of automatic speech recognition systems, including a discussion of how recognition features are derived and how hidden Markov models (HMMs) are used to characterize and model speech. We also cover the use of HMMs in automatic segmentation of speech into sound units, as well in automatic recognition of speech. Next is a discussion of previous attempts at incorporating duration modeling into recognition systems. We then discuss some automatic techniques to combine the outputs of multiple recognition systems and choose the best overall hypothesis. We close with a discussion of missing-feature reconstruction techniques which are used extensively in our normalization procedures.

2.1 Automatic Speech Recognition Systems

Speech recognition systems follow the standard, two-stage pattern classification paradigm (Rabiner & Juang, 1993). Stage 1 is to extract relevant features from the observed signal, and Stage 2 is to make some decision based on the features that are observed. A generic pattern recognition system is illustrated in Figure 2.1.

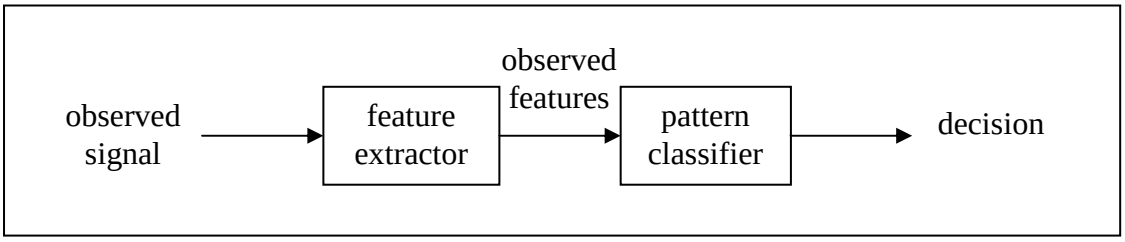

Figure 2.1 Block diagram of a simple pattern classification system. Speech recognition systems are complex pattern classification systems.

In automatic speech recognition, the observed signal is a measurement of air pressure fluctuations recorded by a microphone. The speech is captured as a one-dimensional, time-varying signal. The feature extractor converts the speech signal into a parameterized sequence of feature vectors prior to classification. Recognition systems begin by breaking the speech signal into frames. A frame of speech is a short, windowed segment on the order of 20–30 ms in duration. Each frame of speech is then typically converted to a vector of mel-frequency cepstral coefficients (MFCCs) (Davis & Mermelstein, 1980) or variants of MFCCs (Hermansky, 1990).

For recognition purposes, a speech utterance is modeled as a sequence of sound units. The speech pattern classification engine attempts to automatically identify the correct sequence of sound units found in the

speech signal based on the observed sequence of feature vectors. Typical recognition systems use the phonemes in the language as basic sound units, but other units of varying durations are possible (*e.g.* phoneme sequences, syllables, words, word compounds).

Let $\mathbf{O}$ represent the observed sequence of feature vectors extracted from the speech utterance being recognized. Speech recognition engines search for the optimal sequence of words $\hat{W}$ which maximizes the likelihood of the observation sequence $\mathbf{O}$. The standard Bayesian optimal classification equation for speech recognition is as follows:

$$\hat{W} = \arg\max_{W} \{P(\mathbf{O}|W)P(W)\} \qquad (2.1.1)$$

The term $P(\mathbf{O}|W)$ is called the *acoustic model*; it measures the likelihood that the observed sequence of feature vectors **O** corresponds to a given sequence of words *W*. The term $P(W)$ is called the *language model*; it is an *a priori* measurement of the likelihood that the given sequence of words *W* occurs in the language.

2.2 Speech Features

As mentioned earlier, recognition systems use mel-frequency cepstral coefficients (MFCCs), a parametrical representation derived from the speech signal, to model and recognize speech. The process of converting speech to MFCCs is an efficient approximation of the transformations that the human auditory system makes before sending speech information to the brain. The standard MFCC extraction algorithm is illustrated in Figure 2.2.

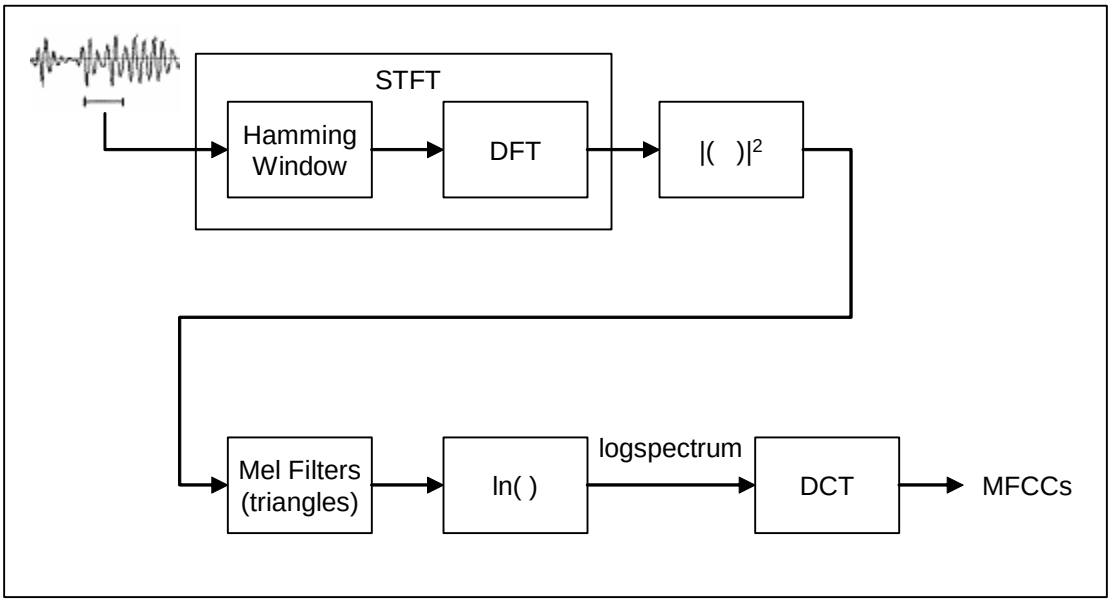

Figure 2.2 Block diagram of the speech feature extraction process. Our work on duration normalization is performed in the log spectral domain.

Each frame of speech is multiplied by a Hamming window and transformed to the frequency domain by the Discrete Fourier Transform (DFT). This process of segmenting a signal in time, applying a window to each segment, and transforming to the frequency domain is known as the Short-Time Fourier Transform (STFT) (Nawab & Quatieri, 1988). The magnitude of the resulting STFT coefficients is computed, and the resulting coefficients are squared, disregarding the phase information that is not necessary for accurate speech recognition.

A bank of triangular shaped mel filters is then applied to the magnitude-square STFT coefficients. The filter's triangles are spaced according to the mel frequency scale, which is approximately linear at lower frequencies and logarithmic at higher frequencies. Adjacent triangles overlap by 50%. The signal energy contained in each triangle is computed, and the resulting values compose a vector of mel-spectral coefficients corresponding to the speech frame. The natural logarithm is then applied to the mel-spectral coefficients, producing a vector of log mel-spectral coefficients.

The sequence of log mel-spectral vectors corresponding to the entire speech signal composes the log mel spectrum of the speech signal. In this thesis, we will typically refer to these values as the *log spectral coefficients* or *log spectrum* of the speech signal. Note that our work on duration normalization is performed in the log spectral domain, prior to the final transformation into MFCC coefficients.

Finally, the Discrete-Cosine Transform (DCT) is applied to each log spectral vector to derive the mel-frequency cepstral coefficients. The output of the DCT is truncated (typically the first 13 coefficients are kept) to form the vector of MFCCs for each frame.

2.3 Hidden Markov Models (HMMs)

A hidden Markov model (HMM) (Baker, 1975) is a probabilistic state machine that can be used to model and recognize speech. Consider the speech signal as a sequence of observable events generated by the mechanical speech production system which transitions from one state to another when producing speech. The term "hidden" refers to the fact the state of the system (*i.e.* the configuration of the speech articulators) is not known to the observer of the speech signal. Speech recognition systems use HMMs to model each sound unit in the language. In this thesis, we have developed a method to help overcome some of the difficulties that occur when HMMs are used to model and recognize spontaneous speech.

In an HMM, each state is associated with a probability distribution that measures the likelihood of events generated by the state. These distributions are known as *output* or *observation* probability distributions. Each state is also associated with a set of *transition* probabilities. Given the current state, transition probabilities model the likelihood that the system will be in a certain state when then the next observation is produced. Typically, Gaussian distributions are used to model the output distribution of each HMM state. The transition probabilities determine the rate at which the model transitions from one state to the next, giving the model some flexibility with respect to sound units which may vary in duration. Figure 2.3 shows a typical left-to-right HMM topology used to model speech sounds. The output distributions and transition probabilities are also illustrated.

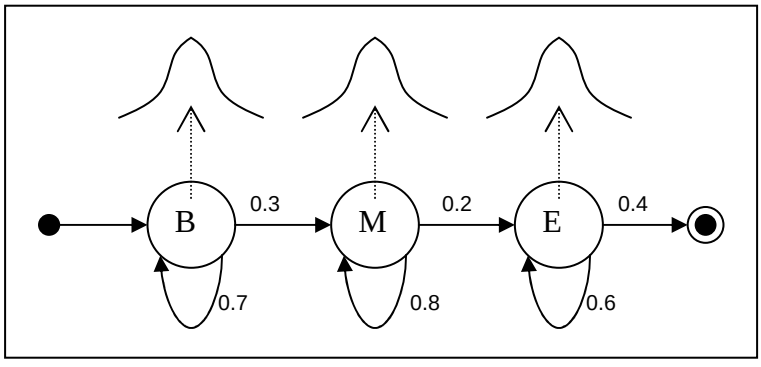

Figure 2.3 Diagram of a typical HMM with explicit output distributions and transition probabilities. Transition probability values are shown on the arrows that transition from one state to the next. Output distributions are shown as Gaussian pdf curves above each state.

State-of-the-art recognition systems today make use of Continuous Density HMMs which model the feature vectors directly. The output distribution of Continuous HMMs is a continuous probability density function (pdf) which contains a corresponding likelihood score for every possible feature vector without quantization. A mixture Gaussian distribution with a finite number of densities is the most common pdf used for Continuous HMM modeling because it has a general shape and parameters that can be automatically re-estimated during training. Large-scale recognition systems trained on large databases train mixture models on the order of 16 or 32 Gaussians per state.

In cases where there is a limited amount of speech training data available, Semi-Continuous Density HMMs are used. Semi-Continuous HMMs share a codebook of mean and variance vectors among all states in the HMM acoustic model. The typical codebook size is 256 vectors that are obtained by *k*-means clustering. Once the codebook is formed, the mixture weights corresponding to each of the 256 means and variances are trained independently for each state in the HMM model.

Given an ensemble of transcribed speech data, the HMM model parameters are automatically learned using the Baum-Welch or forward backward algorithm (Baum, 1972; Rabiner & Juang, 1993). Baum-Welch training is an iterative, expectation-maximization procedure which uses the training data to derive an optimal set of HMM transition probabilities and output distributions. The derived model parameters are optimal in the maximum-likelihood (ML) sense, *i.e.* the resulting model parameters maximize the likelihood that the training data were generated by the HMM.

When speech is spontaneous, there is a high level of variability in the training examples for each sound unit. This variability makes it more difficult for the Baum-Welch algorithm to reliably estimate the corresponding HMM parameters for each sound unit. The inherent variability of spontaneous speech also makes recognition of spontaneous speech via HMMs problematic. This thesis attempts to address these weaknesses and improve the effectiveness of HMM-based speech recognition systems.

2.4 Viterbi Alignment of a Transcript to Speech Data for Segmentation

In this thesis, we must be able to segment the speech signal into sound units prior to normalization. The following technique allows us to automatically derive the location of phoneme boundaries assuming we know the correct transcript of the words spoken.

Given the observed feature vectors derived from a speech signal, a set of HMM acoustic model parameters, and a transcript of the speech, the Viterbi algorithm (Viterbi, 1967) is used to find the most likely time alignment of the transcript to the speech, and thus the corresponding phoneme segmentation

information. This process is commonly referred to as *Viterbi forced alignment*, or simply *forced alignment* or *Viterbi alignment*.

Mathematically, the problem is described as follows. Let **O** be the sequence of feature vectors derived from the speech signal. Let w_C be the word sequence contained in the correct transcript. Let λ be the HMM acoustic modeling parameters. Our goal is to find the state sequence $\hat{\mathbf{s}} = \{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_T\}$ that maximizes the probability that the HMM generated the observed speech data, *i.e.* find $\hat{\mathbf{s}}$ such that:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \left\{ \sum_i \ln\left(P\left(s_i \mid s_{i-1}, w_C, \mathbf{O}, \lambda\right)\right) \right\} \qquad (2.4.1)$$

The Viterbi algorithm makes a fundamental assumption: when computing the probability scores for each state at time t+1, we need only the probability score of the *most likely* state sequence up to time t. The output of the Viterbi algorithm is the most likely sequence of HMM states that generated the observed feature sequence.

To perform Viterbi alignment, we form an HMM model for each word in the sentence by concatenating the HMMs for the sound units that make up the word. The sentence HMM is then formed by concatenating the word HMM models with an optional silence HMM between each word. Once the HMM is built, the Viterbi algorithm aligns the speech features to the sentence HMM and produces a listing of the most likely state for each frame of speech. This state-by-state information can then be used to derive alignment information of the transcript to the speech on a phone-by-phone or word-by-word basis.

2.5 "Decoding": Recognizing and Automatically Transcribing Speech

The heart of automatic speech recognition is the search for the most likely word sequence given the observed features extracted from the speech signal. This is commonly referred to as *decoding* or *recognizing* the speech signal.

When decoding speech, we begin by constructing a search graph which contains every word in the recognition vocabulary. Each word is then replaced by the HMMs that correspond to the sequence of sound units which make up the word. As a result, the search graph is a large HMM, and recognition is performed using the Viterbi algorithm to align the search graph to the speech features derived from the utterance. Because the Viterbi algorithm is used to find the most likely word sequence, the decoding procedure is said to be done via *Viterbi search*. For a complete description of the Viterbi search algorithm used to decode speech, see (Jelinek, 1997).

Note that the search for the most likely word sequence is constrained by the language model being used. Practical recognition systems use context dependent *trigram* language models, which assign probabilities the occurrence of sequences of three words in the language. The search graph derived for trigram language models is complex. If the recognition vocabulary contains N words, the number of states in the search graph is proportional to N^2. The vocabulary size for a practical system is on the order of 10,000 words, which makes a search of the complete trigram search graph intractable. In practice, a *beam search* is used to prune away unlikely paths at every step in the search process. The *beam width* parameter which controls the pruning is chosen so that the recognition is both practical and accurate.

The figure of merit for automatic speech recognition system is known as the *word error rate* (WER). The hypothesized word sequence generated by the decoder is aligned to the reference transcript for the speech data using a non-linear string matching algorithm (Pallet *et al*, 1990). There are three possible types of errors that can be made: An *insertion error* occurs when the ASR system generates a word that does not correspond to any word in the reference transcript. A *deletion error* occurs when the reference transcript contains a word that has no corresponding word in the ASR hypothesis. A *substitution error* occurs when the corresponding word in the ASR transcript is different than that of the reference transcript. The word error rate is the ratio of the total number of errors made (insertions, deletions, and substitutions) to the total number of words in the reference transcript. WER scores are typically reported as percentages. Note that given this formulation, WER scores greater than 100% are possible.

2.6 Explicit State Duration Modeling with HMMs

The inherent probability distribution controlling the duration of each state in a standard HMM framework is exponential in form:

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii}) \qquad (2.6.1)$$

where a_{ii} is the probability of transition from state i to itself, and d is the number of consecutive observations that correspond to state i. For modeling speech signals, this distribution is inappropriate and has been characterized as a weakness of the speech HMM. In the 1980s, researchers experimented with a framework that can incorporate explicit state duration models into an HMM framework (Ferguson, 1980; Russell & Moore, 1985; Levinson, 1986). This framework is known as a Hidden Semi-Markov Model (HSMM) and is illustrated in Figure 2.4.

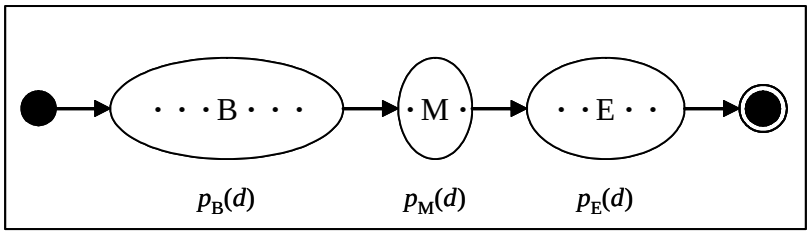

Figure 2.4 Illustration of a Hidden Semi-Markov Model (HSMM) with explicit state duration distributions $p(d)$ corresponding to each state.

In the HSMM, the self transition probabilities have been replaced by the explicit state duration densities $p_i(d)$, and the model is only allowed to transition to the next state after the duration density specifies that the appropriate number of observations have taken place. Note that if $p_i(d)$ is set to the exponential density of Eq. 2.6.1, then the HSMM framework is equivalent to the standard HMM.

The advantage of HSMM is that the quality of the modeling is significantly improved. When implementing HSMM recognition systems, the state duration distributions are truncated to a maximum duration value D for practical reasons. Using a parametric framework for the duration densities of the HSMM, Levinson extended the Baum-Welch algorithm and proved that the training would converge (Levinson, 1986). Recognition with HSMMs is performed by an extension of the Viterbi algorithm which allows for the computation of the probability at a given frame based on the values at D preceding frames (instead of just 1 preceding frame).

However, there are several drawbacks: There is a larger number of parameters (D) associated with each state which must be estimated from the data. Direct implementation of the algorithm increased computation by a factor of D^2. Parametric formulations are more efficient, with computation increased by a factor of D. The storage and computation requirements for the extended Viterbi algorithm for HSMM-based decoding are increased by a factor of D as well.

Researchers observed that although the duration modeling quality of HSMM-based systems was better at the state level, the WER improvements observed were small, especially for connected word recognition tasks. Consequently, this approach has not been widely incorporated in state-of-the-art recognition systems today.

2.7 A Brief Overview of Other Related Research in Duration Modeling

Duration modeling research focuses on the development of accurate statistical models for capturing and predicting the phoneme duration information observed in natural speech. It is generally accepted that

duration information should play an important role for speech when speech is highly spontaneous with large changes in speaking rate.

While we are not trying to model duration explicitly in our research, prior work on duration modeling is relevant to proper segmentation and decomposition of the speech waveform prior to applying our techniques. At the end of this section, we report previous attempts made by duration-modeling researchers to normalize for the effects of varying phone duration.

Duration modeling research began in the 1970s with a focus predicting the proper duration of each phone for natural-sounding speech synthesis applications. Umeda and Klatt focused on rule-based approaches to explain and generate natural segmental duration behavior (Umeda, 1975, 1977; Klatt, 1973, 1976). They were both able to predict segment durations and explain segmental duration variations with reasonable accuracy.

In the late 1980s, duration modeling research focused on models that could be applied to recognition. Port *et al.* examined words produced by different speakers and at different speech rates and attempted to capture the relevant syllable timing information (Port *et al.*, 1988). They used manually derived segmentations of words into primitive units (*e.g.* stop closures, fricatives, vowels) and discriminant analysis to extract relevant information for the differentiation of words in a small vocabulary recognition system. They were successful when words varied dramatically in consonant voicing and stress patterns. They also observed that uniform scaling to eliminate tempo variation as a duration normalization approach would be less effective since changes in overall speech rate do *not* uniformly affect the underlying segmental durations. In 1988, Crystal and House used Hidden Markov Models (HMMs) with carefully tailored topologies to derive mathematical fits to the distributions of the durations of different classes of phones (Crystal & House, 1988). They also postulated a method for embedding their models into a speech recognition framework.

In the early 1990s, the focus was on more elaborate duration models for speech synthesis. Campbell argued that a hierarchical framework is essential to properly capture and model speech timing information (Campbell & Isard, 1991; Campbell 1992). His models attempted to capture duration information at the phrase, foot, and syllable level. The final phonetic segment duration information could then be derived from the resulting interaction of those higher level effects. Campbell observed that while syllable duration is well-predictable, prediction of duration at the phone level is more difficult because there is an inherent relative freedom of phonetic duration variation within a syllable.

More recently, work has again focused on employing duration information to improve speech recognition accuracy. Since it is difficult to incorporate explicit duration information into the HMM itself, most

duration work to date has focused on post-processing. Pitrelli employed a hierarchical recognition model based on phoneme duration (Pitrelli, 1990). He showed a 19% reduction in relative WER on a limited vocabulary, isolated-word recognition system when his models were applied to rescore recognition hypotheses based on duration information. Osaka *et al.* created a word recognition system which adapted to speaking rate (Osaka *et al.*, 1994). Their procedure used phoneme duration as an estimate for speech rate. They normalized phone duration based on the average vowel duration and the average duration of each phone class to yield an increase in accuracy for a system with a 212-word vocabulary.

Jones and Anastasakos used duration information as a post-processing step to improve recognition accuracy (Jones & Woodland, 1993; Anastasakos *et al.*, 1995). They both used duration models to re-score the *N*-best hypothesis list produced by an HMM-based recognizer. Anastasakos noted that the *N*-best paradigm is advantageous because it provides phoneme boundary information and speaking rate information. In both sets of experiments, duration models were developed for automatically-clustered sets of "slow" and "fast" segments. Jones' speech-rate measure was based on average normalized phone duration, and the relative utterance speaking rate was based on the average normalized phone duration in the utterance. Anastasakos' rate measurement was based on observations from a given phone segment as well as the context of a small number of surrounding phone segments. Both researchers attempted to normalize phone duration with respect to their rate estimations by considering phone duration as a function of speaking rate. Jones showed a 10% reduction in relative WER on the TIMIT database from a baseline of 13.6%. Anastasakos showed a 10% reduction in relative WER on the WSJ database from a baseline of 7.7%. These results indicate that recognition accuracy can be improved when duration information is properly modeled.

2.8 "Hypothesis Combination": Automatic Combination of Multiple Hypothesized Speech Transcripts

Combination of multiple recognition hypotheses is a successful technique for compensating for noisy speech. Hypothesis combination can be performed on the output of various recognition systems, or on the output of a single recognition system recognizing multiple feature streams. The success of combining recognition hypotheses depends on the "heterogeneity" of the information sources being combined.

The National Institute of Standards and Technology (NIST) developed a system for hypothesis combination known as Recognizer Output Voting Error Reduction (ROVER) (Fiscus, 1997). The ROVER system makes use of a voting scheme to combine the final recognition hypotheses of multiple

recognition systems. ROVER has been successfully employed in a series of Broadcast News (HUB4) and Conversational Speech (HUB5) evaluations.

While working with the Speech In Noisy Environments (SPINE) evaluation conduced by the Naval Research Labs (NRL) in August 2000, Singh *et al* proposed a parallel hypothesis combination scheme based on word-graphs in order to compensate for the effects of speech utterances with very low signal-to-noise ratios (SNRs) (Singh, *et al.*, 2000). In this thesis, we make use of Singh's word-graph hypothesis combination method to combine recognition hypotheses derived from multiple time warpings of a speech utterance. The details of word graph-based hypothesis combination are presented below.

Initially, the word hypotheses obtained from parallel recognition of multiple feature streams are combined into a word graph. Each word in the hypothesis represents a node in the graph, and the acoustic score of each word is associated with the corresponding graph node. Next, merging is performed on all graph nodes where the same words are hypothesized at the same time. Since acoustic scores are typically given as log-likelihoods, the following formula is used to compute the score of a node after merging:

$$\mathrm{Scr}' = \ln\left(e^{\mathrm{Scr1}} + e^{\mathrm{Scr2}}\right) \qquad (2.8.1)$$

where Scr1 is the acoustic score of the word in the first hypothesis and Scr2 is the acoustic score of the word in the second hypothesis.

Finally, links are added to the graph between nodes where the word end time of the previous word and the word begin time of the following node differ by less than 30ms. Figure 2.5 illustrates two parallel recognition hypotheses in word graph form before combination, and Figure 2.6 illustrates the result of constructing a word graph from the two parallel hypotheses.

Note that in Figure 2.6, additional transitions have been permitted when both hypotheses have word transitions at the same instant in time ("*t*"). The final words in both hypotheses are identical both in label ("</s>") and time, and therefore they have been merged into a single node. The log-likelihood acoustic score ("Scr") of the merged node is calculated by appropriate combination of the original two scores.

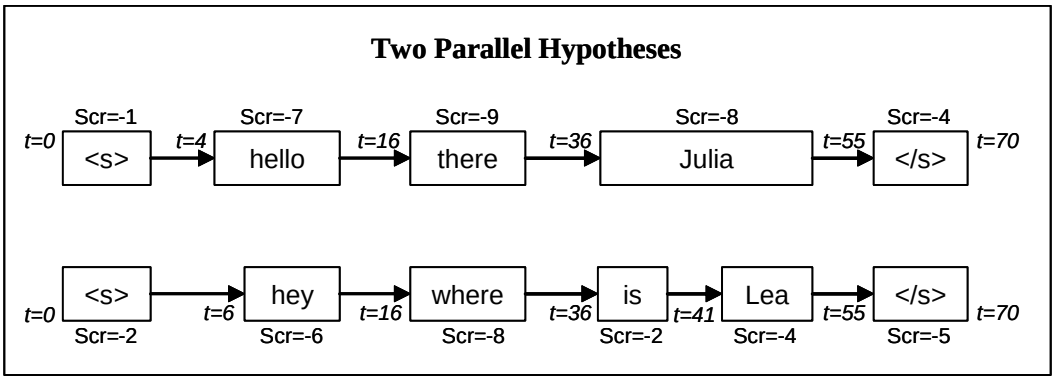

Figure 2.5 Illustration of two parallel hypotheses in word graph form before combination. Acoustic log-likelihoods are labeled "Scr" and placed above or below the corresponding graph nodes. The transition times are labeled "*t*" and are placed before or after the corresponding graph nodes.

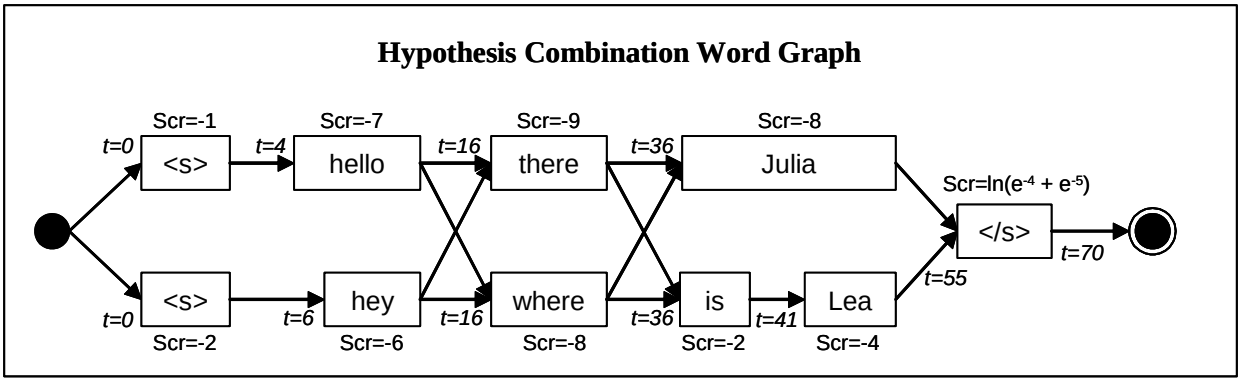

Figure 2.6 The two parallel hypotheses shown in Figure 2.5 have been merged into a single word graph.

After the word graph is formed, the language model is applied to score all paths through the graph. The words along the path with the highest score are chosen as the final, combined recognition hypothesis.

2.9 Missing Feature Compensation for Speech Recognition

Missing feature methods are a series of compensation techniques designed to better recognize speech that is corrupted by noise (Cooke *et al.*, 2001; Raj *et al.*, 2000). Missing feature methods begin by locating components of the observed speech feature vectors that have a low signal-to-noise ratio (SNR). Once the "missing" low SNR regions are identified, there are two methods to compensate:

1. marginalization – recognize the speech using only the reliable or "present" components of higher SNR, ignoring the "missing" regions of lower SNR

2. reconstruction – first use statistical methods or other data driven processes to reconstruct the missing components of the speech feature vectors, and then perform recognition in the usual manner on the reconstructed vectors

Locating and reconstructing the missing speech components are typically performed in the log spectral domain before the speech features are converted to cepstral coefficients. The marginalization-based missing feature compensation techniques are less effective due to the fact that the recognition must also occur in the log spectral domain. The reconstruction-based missing feature compensation techniques are favorable because after the complete log spectral vectors are reconstructed, they can then be converted to the superior MFCC recognition features and recognized with state-of-the-art recognition techniques.

In this thesis, we apply missing-feature reconstruction techniques to reconstruct "missing" portions of fast, spontaneous speech in an effort to recover information that is lost when speech becomes more casual or more rapid. The following sub-section describes the covariance-based reconstruction technique that Raj developed in his Ph.D. research on the reconstruction of incomplete spectrograms (Raj, 2000). These covariance-based reconstruction methods are employed throughout the work of this thesis to compensate for the rapid and unpredictable nature of spontaneous speech.

2.9.1 Estimation of Parameters Needed for Covariance-based Missing-feature reconstruction

A speech spectrogram comprised of the sequence of log spectral vectors extracted from the speech signal can be modeled as the output of a Gaussian wide-sense stationary (WSS) random process (Papoulis, 1991). If we assume that all possible spectrograms are individual observations of a single random process, we can use the statistical parameters of the process to estimate the missing components of spectrograms. In his thesis work, Raj referred to this method of reconstruction as *covariance-based missing-feature reconstruction* (Raj, 2000). The mathematical theory behind this approach is detailed below.

Let $S(t,k)$ be a spectrogram corresponding to a speech utterance. The time index *t* identifies the frame of speech, and the frequency index *k* identifies the component of the log spectral vector, *i.e.* the index of the mel triangle that the component was derived from. For computational convenience, we use spectrograms derived with 20 mel frequency components when performing missing-feature reconstruction. The number of time frames in a given utterance is on the order of hundreds of frames.

Define $\mu(t,k)$ to be the mean of the k^{th} element of the t^{th} log spectral vector. Also define $c(t_1,t_2,k_1,k_2)$ to be the covariance between $S(t_1,k_1)$ and $S(t_2,k_2)$, *i.e.* the covariance between the k_1^{th} component of the t_1^{th} log spectral vector and the k_2^{th} component of the t_2^{th} log spectral vector. Using the expectation operator *E*[ ], the mean and covariance are given by the following equations:

$$\mu(t,k) = E[S(t,k)] \tag{2.9.1}$$

$$c(t_1,t_2,k_1,k_2) = E[(S(t_1,k_1) - \mu(t_1,k_1))(S(t_2,k_2) - \mu(t_2,k_2))] \tag{2.9.2}$$

Because we assume that the process generating the spectrogram is a wide-sense stationary process, we may assume that of the mean value $\mu(t,k)$ of the k^{th} component of a log spectral vector does not depend on where it occurs in the spectrogram (t). We may also assume that the covariance between two components $c(t_1,t_2,k_1,k_2)$ does not depend on their absolute location in the spectrogram (t_1 and t_2), but rather the covariance depends only on the distance τ between the two time indices ($\tau = |t_2 - t_1|$). The wide sense stationary assumption gives us the following two simplified equations for the log spectral mean and covariance (Papoulis, 1991).

$$\mu(t,k) = \mu(t_1,k) = \mu(k) \tag{2.9.3}$$

$$c(t,t+\tau,k_1,k_2) = c(t_1,t_1+\tau,k_1,k_2) = c(\tau,k_1,k_2) \tag{2.9.4}$$

Using this formulation, the proper mean and covariance parameters of speech log spectral vectors can be estimated from a training corpus of clean speech data. Because we assume that the generating process is Gaussian, the mean and covariance parameters completely specify the process and provide all the information we need to reconstruct missing spectrogram features.

The expected value of every component in the spectrogram is given by $\mu(k)$, and the covariance between any component in the spectrogram with any other component in the spectrogram is given by $c(\tau,k_1,k_2)$

$$E[S(t,k)] = \mu(k) \tag{2.9.5}$$

$$E[(S(t_1,k_1) - \mu(t_1,k_1))(S(t_2,k_2) - \mu(t_2,k_2))] = c(\tau,k_1,k_2) \tag{2.9.6}$$

2.9.2 Covariance-based Missing-feature reconstruction

Given these statistical parameters described in Section 2.9.1, we can reconstruct spectrograms containing missing features as follows. Let $\boldsymbol{S}$ be a spectrogram with missing components. Arrange the observed, uncorrupted components of $\boldsymbol{S}$ into a vector $\boldsymbol{S_o}$. Also arrange the missing components of $\boldsymbol{S}$ into another vector $\boldsymbol{S_m}$. We know the mean of every component in the spectrogram and the covariance between any two components in the spectrogram; therefore, we can construct the following four items necessary for reconstruction:

1. $\boldsymbol{\mu}_o^S$ – the mean vector of $\boldsymbol{S_o}$ (the present log spectral components)

2. $\boldsymbol{\mu}_m^S$ – the mean vector of $\boldsymbol{S_m}$ (the missing log spectral components)

3. $\mathbf{C}_{oo}$ – the autocovariance matrix of $\boldsymbol{S_o}$

4. $\mathbf{C}_{mo}$ – the crosscovariance matrix between $\boldsymbol{S_m}$ and $\boldsymbol{S_o}$

Using these parameters, we are able to make an MAP estimate $\hat{\boldsymbol{S}}_{\boldsymbol{m}}$ for the missing components $\boldsymbol{S_m}$ as follows:

$$\hat{\boldsymbol{S}}_{\boldsymbol{m}} = \boldsymbol{\mu}_m^S + \mathbf{C}_{mo}\mathbf{C}_{oo}^{-1}\left(\boldsymbol{S_o} - \boldsymbol{\mu}_o^S\right) \qquad (2.9.7)$$

Eq. 2.9.7 reconstructs all missing elements at one time, but this equation is not computationally efficient. A typical 4 second utterance has 400 frames of speech and 20 frequency components for each frame. Assuming 50% of the features are missing, the matrices $\mathbf{C}_{oo}$ and $\mathbf{C}_{mo}$ would have dimension 4000×4000. In this example, the direct computation of the MAP reconstruction estimate $\hat{\boldsymbol{S}}_{\boldsymbol{m}}$ would require the inversion of a 4000×4000 matrix followed by the multiplication of two 4000×4000 matrices. For practical applications, missing elements are reconstructed incrementally, one at a time. For more details on incremental approaches for missing-feature reconstruction, see Raj's thesis (Raj, 2000).

2.10 Conclusions

In this chapter we presented a brief overview of speech recognition technologies that are relevant to the remainder of the thesis. We started with an overview of automatic speech recognition systems and continued with the transformation of the speech waveform into standard MFCC feature vectors. We described the HMM acoustic models used to model and recognize speech, and provided an overview of the use of HMMs in practical applications. Viterbi alignment is used to align a known transcript to speech data, and Viterbi decoding is used to generate a likely transcript for speech data whose transcript is not known.

We also gave a brief overview of previous attempts to incorporate explicit duration modeling into the recognition framework. Although methods were developed to incorporate duration modeling into the HMM framework, and attempts were made to rescore candidate hypotheses based on duration information, explicit duration modeling is not widely incorporated in state-of-the-art recognition systems today.

We closed with some discussion of hypothesis combination techniques and missing-feature reconstruction, both of which play an instrumental role in the duration normalization research that we develop in this thesis. In the next chapter, we present a brief overview of the SPHINX-III speech recognition system and the speech corpora used in this research. In Chapter 4, we detail the duration normalization algorithm at the heart of this thesis.

3: Speech Recognition System Resources and Speech Corpora

This chapter provides an overview of the specific speech recognition system and speech databases used while conducting our research. The focus of our research is on modifying the speech features prior to training recognition models or recognizing test speech; therefore, the algorithms we develop and the results we present are independent of the specific recognition engine used. The particular aspects of the SPHINX-III recognition system and speech databases are presented to provide the reader with useful context information for interpreting our results and to provide other researchers with enough information to repeat and validate our experiments.

3.1 The SPHINX-III Speech Recognition System

SPHINX-III is the third in a series of state-of-the-art Hidden Markov Model (HMM)-based speech recognition systems pioneered at Carnegie Mellon University (CMU) beginning in the late 1980s. The original SPHINX system was developed by Kai-Fu Lee in 1988 (Lee 1989; Lee *et al.* 1990). SPHINX was one of the first systems to demonstrate speaker-independent, large-vocabulary continuous speech recognition. In 1993, Xuedong Huang *et al.* presented SPHINX-II, one of the first systems to make use of semi-continuous HMM output distributions (Huang *et al.*, 1993).

SPHINX-III was developed and implemented by Ravishankar Mosur and Eric Thayer in the mid 1990s.SPHINX-III provides more flexibility in the modeling and feature frameworks for speech recognition. SPHINX-III allows the user to choose between (fully-)continuous or semi-continuous HMM output distributions. SPHINX-III also allows the user to divide the data into a multiple number of streams and specify how these streams are organized. This feature allows for recognition based on a multiple number of data sources (*e.g.* recognition based on a combination of audio and visual features).

A basic block diagram of the SPHINX-III recognition system is shown in Figure 3.1. For more detailed information on the SPHINX-III system, see (Placeway *et al.*, 1997). For more information on the differences between semi-continuous and fully-continuous HMM output distributions, see the latter part of Section 2.3 in the previous chapter.

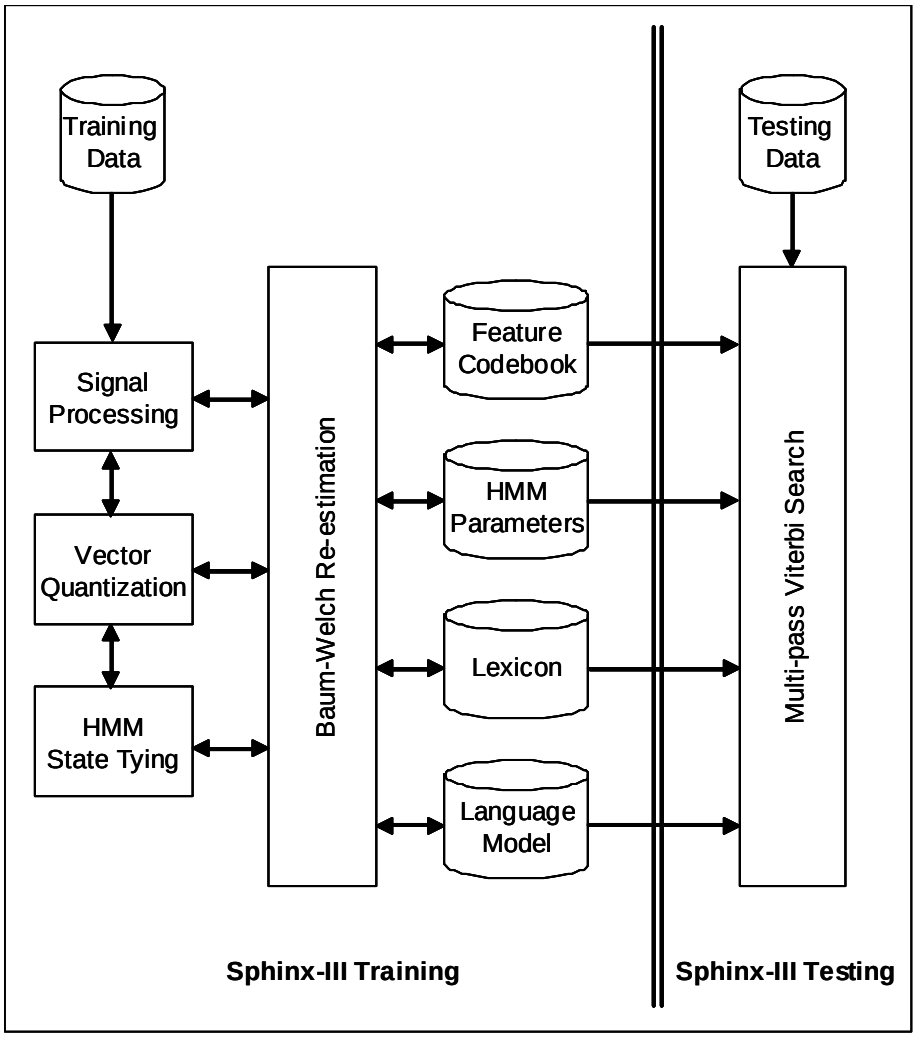

Figure 3.1 Block diagram for the SPHINX-III speech recognition system. Training elements are shown on the left of the figure. Testing elements are shown on the right.

3.2 Speech Database Information

In this section, we describe in brief detail the speech databases used in this thesis: the Telefónica Cellular Telephone Corpus (TID), the NIST Multiple Register Corpus (MR), and the NIST Broadcast News Corpus (BN). TID and MR are smaller corpora with a high level of spontaneity, and BN is a large-scale corpus. Throughout the thesis research, many algorithms were first tested on TID and/or MR. The algorithms showing the most promise were then further tested on the BN data to validate our results.

3.2.1 TID: The Telefónica Cellular Telephone Corpus

We conducted experiments on a Spanish database recorded by Telefónica Investigación y Desarollo in Madrid, Spain. The database consists of cellular telephone callers repeating a small string of digits or a monetary amount. Volunteers were read a prompt and asked to repeat it in a casual manner. The TID

speech is highly spontaneous. Figure 3.2 shows sample utterances from the TID database, along with English translations.

```
quince euros y veinte centimos
    fifteen euros and twenty cents
cuarenta millones noventay una
    forty million ninety one
ochenta cero quinientos setenta siete ochentay tres
    eighty zero five-hundred seventy six eighty three
cien cinco quinientos
    one-hundred five five-hundred
```

Figure 3.2 Example utterances from the TID corpus. English translations are given in italicized text below each example utterance.

The TID speech is small vocabulary: the entire recognition vocabulary is made up of 59 words. Figure 3.3 contains every entry in the TID recognition dictionary. Note that Spanish orthography and pronunciation are directly related, and the dictionary contains no alternate pronunciations.

```
CATORCE     K A T O R Z E
CENTIMO     Z E N T I M O
CENTIMOS    Z E N T I M O S
CERO        Z E R O
CIEN        Z I E N
CIENTAS     Z I E N T A S
CIENTO      Z I E N T O
CIENTOS     Z I E N T O S
CINCO       Z I N K O
CINCUENTA   Z I N K WE N T A
CINCUENTAY  Z I N K WE N T AY
CON         K O N
CUARENTA    K WA R E N T A
CUARENTAY   K WA R E N T AY
CUATRO      K WA T R O
DE          D E
DECIMAS     D E Z I M A S
DIECI       D IE Z I
DIEZ        D IE Z
DOCE        D O Z E
DOS         D O S
EL          E L
EURO        EW R O
EUROS       EW R O S
MEDIA       M E D IA
MEDIO       M E D IO
MIL         M I L
MILLON      M I LL O N
MILLONES    M I LL O N E S
NOVE        N O V E
NOVECIENTOS N O V E Z I E N T O S
NOVENTA     N O V E N T A
NOVENTAY    N O V E N T AY
NUEVE       N WE V E
OCHENTA     O CH E N T A
OCHENTAY    O CH E N T AY
OCHO        O CH O
ONCE        O N Z E
QUINCE      K I N Z E
QUINIENTAS  K I N IE N T A S
QUINIENTOS  K I N IE N T O S
SEIS        S EI S
SESENTA     S E S E N T A
SESENTAY    S E S E N T AY
SETE        S E T E
SETENTA     S E T E N T A
SETENTAY    S E T E N T AY
SIETE       S IE T E
TRECE       T R E Z E
TREINTA     T R EI N T A
TREINTAY    T R EI N T AY
TRES        T R E S
UN          U N
UNA         U N A
UNO         U N O
VEINTE      V EI N T E
VEINTI      V EI N T I
VENTISIETE  V E N T I S IE T E
Y           I
```

Figure 3.3 The recognition dictionary for the TID corpus. The listing contains all 59 words and corresponding pronunciations.

The TID training set consists of 3458 utterances (15543 words), and the testing set consists of 1728 utterances (7634 words). This translates to approximately 4 hours of training data and 2 hours of testing data in the corpus. The average utterance is approximately 4.2 seconds long and contains 4.5 words.

TID speech was collected over European cellular telephone channels, which make use of Global System for Mobile telecommunication (GSM) lossy speech compression. GSM coding uses Regular Pulse Excitation – Long Term Prediction (RPE – LTP) algorithms to digitally compress the speech signals. For our research, the cellular telephone speech has been decompressed and stored as a standard waveform prior to training and recognition. Research has shown that the effects of GSM coding on recognition accuracy with the TID database and the SPHINX-III recognition system are minimal (Huerta, 2000).

3.2.2 MR: The NIST Multiple-Register Corpus

The NIST Multiple Register Speech Corpus (MR) is a parallel corpus for comparison of spontaneous and read speech recorded at SRI. The database contains fifteen spontaneous conversations on assigned topics and re-read versions of the same conversations. For this thesis research, we focus on the examples from the spontaneous register, but at times we experiment with the read counterpart for comparison.

The MR utterances contain highly spontaneous speech with many conversational fillers (*e.g.* ++uh++, ++um++), long pauses, partial words, and repeated words. Also, the grammar is loose and often "improper" according to standard English grammar rules. Figure 3.4 shows an excerpt from one of the conversations on sports and exercise.

```
s1: hi <sil> how're <sil> you doing <sil>
s2: ++mouthnoise++ hi good thanks
s1: what kind of exercise you do <sil>
s2: <sil> oh ++uh++ <sil> my favorite is tennis <sil>
s1: really
s2: <sil> you much of a tennis fan <sil>
s1: yeah <sil> what ever happened to chang <sil>
s2: ++uh++ chang he hasn't been in in the running for <sil>
     for number one <sil> really <sil> seriously he's he's a
     great player good competitor but it just <sil>
s1: really <sil> well i <sil> ++uh++ ++huh++
s2: just doesn't have it to be number one he's <sil>
s1: oh really i'm surprised that agassi's number one i thought
     he was kind of a flake <sil> i <sil> didn't think he had
     the head for ++uh++ <sil> for championship tennis <sil>
s2: well that's that's what everybo- bo- everyone's been
     writing about he he does finally have the head for it
     <sil> he's <sil> he's finally got the ++uh++ <sil> the
     mental game for it <sil>
s1: really going out with barbra streisand really did it for
     him or something <sil>
```

```
s2: i think it <sil> was brooke shields <sil> yeah <sil> that
     did it yeah <sil> that put him over the top <sil>
s1: yeah <sil> oh speaking of tennis what about these gals
     that are playing tennis monica seles is in hiding <sil>
s2: right yeah <sil> she's i think she's withdrawn from from
     c- formal competition <sil> forever yeah <sil>
s1: after she got stabbed <sil>
```

Figure 3.4 An excerpt from a MR conversation between two speakers: s1 and s2. Notice that the speech is characterized by many repeated words, false starts, and repetition. "Noise" and "filler" words are marked with surrounding "++" characters, and long pauses or silence regions are marked as "<sil>".

We divided the MR speech into training and testing sets. Our MR training set consists of 1090 utterances (12209 words), and the testing set consists of 271 utterances (3114 words). There are approximately 80 minutes of training speech and 20 minutes of testing speech in the corpus. The average utterance in the MR corpus contains 11.3 words and is 4.4 seconds long. The conversational nature and limited amount of MR speech available makes this a difficult recognition task for a state-of-the-art recognition system.

3.2.3 BN: The NIST Broadcast News Corpus

In the late 1990s, NIST conducted a series of periodic recognition evaluations on a variety of speech recognition data. HUB4 was one such evaluation series focused on accurate transcription of broadcast news speech (Graff, 1997). Example utterances from the BN corpus are shown in Figure 3.5.

```
we continue our series <sil>
america <sil> in black and white
tonight <sil> how much is <sil> white skin worth
this is a. b. c. news nightline
reporting from <sil> washington
ted <sil> koppel
the business of skin color <sil> inevitably comes up again and
   again <sil>
often as not <sil> white Americans find themselves getting
   defensive on the subject <sil>
it is not <sil> we insist something we dwell on morning noon
   and night
<sil> it is not even the way that most of us define ourselves
```

Figure 3.5 A listing of example utterances from the broadcast news (BN) corpus. Long pauses or silence regions are marked as "<sil>"

Each BN utterance is classified into one of 7 focus (F) conditions according to dialect, mode, fidelity, and background noise (Garofolo, 1997). The focus conditions are detailed in Table 3.1.

Condition	Dialect	Mode	Fidelity	Background
F0: Baseline Broadcast	native	planned	high	clean
F1: Spontaneous Speech	native	spontaneous	high	clean
F2: Reduced Bandwidth	native	(any)	med/low	clean
F3: Background Music	native	(any)	high	music
F4: Degraded Acoustics	native	(any)	high	speech or noise
F5: Non-native Speakers	non-native	planned	high	clean
FX: Other Combinations	–	–	–	–

Table 3.1 Detailed description of broadcast news speech focus conditions as defined by NIST.

We selected a 45 hour subset of the 1996 and 1997 broadcast news corpora to train our acoustic models. Examples were taken from all F conditions. For testing, we made use of the standard 1999 Eval 1 data set, which contains 1 hour of broadcast news speech divided into 347 utterances (11075 words). The average BN utterance contains 19.7 words and has a duration of 6.7 seconds.

3.3.4 Speech Database Summary

To close, we present a table of statistics derived from the speech databases used in our research. A side-by-side comparison of training and testing database size and average utterance length is given in Table 3.2.

	Training Database Size			Testing Database Size			Average Utterance Length	
Database	**hours**	**utterances**	**words**	**hours**	**utterances**	**words**	**seconds**	**words**
TID	4.0	3458	15543	2.0	1728	7634	4.2	4.5
MR	1.3	1090	12209	0.3	271	3114	4.4	11.3
BN	45.0	24319	475372	1.0	347	11075	6.7	19.7

Table 3.2 Size comparison of all speech databases used in this thesis (TID, MR, and BN). Size of the training and testing databases is given in number of hours, number of utterances, and number of words. Also, the average utterance length is given in number of seconds and number of words.

It is interesting to note some similarities and differences between each of the corpora. The average utterance length of TID and MR data are very similar in amount of time (4.2 seconds and 4.4 seconds respectively), but they are vastly different in number of words spoken in that time (4.5 words for TID and 11.3 words for MR). There are several possible factors that contribute to this phenomenon. One is a difference between the Spanish language (TID) and the English language (MR). Another factor may be

the “back-and-forth” nature of the conversational dialog that takes place in the MR corpus compared to the one-sided repetition of digit strings into a cellular phone for TID.

A comparison of BN and MR is a useful English language to English language comparison. Notice that a typical BN utterance contains almost twice as many words as a typical MR utterance. This is largely due to the influence of planned speech in the F0 focus condition, which includes a large number of longer, scripted utterances read by a professional newscaster.

The variety of databases used in this research allows for a robust examination of the quality of the algorithms we develop. It also allows for fast experimentation of a variety of techniques for improved segmentation and recognition quality. In our experience, algorithms that have had the greatest success on the smaller TID and MR databases will also have success on the larger BN database. Conversely, experimental procedures that were not helpful in recognizing TID and MR data were also not useful in recognizing BN data.

3.3 Evaluating Recognition Systems: Accuracy and Statistical Significance

As discussed in Section 2.5, recognition systems are typically evaluated using a metric known as the word error rate (WER). Throughout this thesis, we will use measurements of WER to compare the effectiveness of different algorithms for normalizing the speech prior to recognition.

When comparing different algorithms, it is important to measure not only differences in WER, but also the statistical significance of those differences. In this thesis work, we make use of the Matched-Pairs test proposed by Gillick and Cox (1989). The Matched-Pairs test is a widely accepted method for calculating statistical significance which has also been used by the National Institute of Standards and Technology (NIST) in standard speech recognition evaluations. The significance score produced by the Matched-Pairs test depends on a variety of factors including the error rates of the two systems, the number of utterances in the test set, the vocabulary size, and the range of accuracy within the test set. In particular, the Matched-Pairs test attempts to give weight to instances where one recognition system is able to avoid an error that the other system has made. The output of the Matched-Pairs test is a *p* score which is the probability that the two systems are statistically the same. In general, we say that results are statistically significant if the *p* score is less than 5%.

Although the Matched-Pairs *p* score depends on a variety of factors, we can get a general idea of statistical significance based on absolute differences in WER. Table 3.3 shows examples of *p* score values

and corresponding absolute differences in WER for the TID corpus. Table 3.4 shows similar examples for the MR corpus, and Table 3.5 shows examples for the BN corpus.

ΔWER	***p* score**
0.4%	11.7%
3.0%	6.3×10^{-5}%

Table 3.3 Examples of the correspondence between statistical significance *p*-score and absolute word error rate difference for the TID corpus.

ΔWER	***p* score**
1.5%	7.1%
1.8%	6.4%
2.5%	0.38%
8.6%	2.79×10^{-8}%

Table 3.4 Examples of the correspondence between statistical significance *p*-score and absolute word error rate difference for the MR corpus.

ΔWER	***p* score**
3.9%	0.11%
13.8%	9.18×10^{-11}%

Table 3.5 Examples of the correspondence between statistical significance *p*-score and absolute word error rate difference for the BN corpus.

In the thesis research, the final results presented on MR and BN are statistically significant, while the results presented on the TID data are not below the 5% limit for significance. The TID information was useful in developing this thesis because the trends observed in TID carried over to similar observations on the larger vocabulary MR and BN databases.

3.4 Conclusions

In this chapter we presented a very brief overview of the SPHINX-III automatic speech recognition system. We then described the spontaneous speech corpora used in this research: TID, MR, and BN. Although TID and MR data are small, the results derived on these corpora serve as a consistent indication of the potential for success using large-scale corpora such as BN. We closed this chapter with a descripton of the Matched-Pairs test used to verify the statistical significance of our results, and we included some examples of WER differences and corresponding *p*-scores for each of the corpora used in our research. In the next chapter, we introduce the duration normalization algorithm that we developed for this thesis.

4: The Duration Normalization Algorithm

This chapter begins with a discussion of why it is desirable to normalize the duration of sound units observed in speech prior to modeling and recognition. We then describe in detail the process by which we use missing feature reconstruction techniques to normalize the duration of speech phones. We close with a series of experiments using oracle segmentation information with three databases to investigate the effectiveness and derive an upper bound for accuracy using our duration normalization technique.

4.1 Motivation for Duration Normalization: HMMs and Spontaneous Speech

The hidden Markov model (HMM) is the most widespread and successful modeling framework for large vocabulary, speaker independent speech recognition. We began this research with a simple experiment to see how well standard HMM systems perform on careful speech and how well they perform on spontaneous speech. Using MR, a parallel corpus of spontaneous and read speech, we trained and tested a baseline recognition model for each speech register. The sentences used to train and test each system varied only in the speaking register; everything else remained the same. In the baseline case, a system trained and tested on read speech had a word error rate (WER) of 15.6%, while the parallel system trained and tested on spontaneous speech had a WER of 40.3%. These results indicate that our state-of-the-art ASR system can experience a relative degradation in accuracy of over 150% when the speech being recognized becomes conversational.

It is well known that HMMs do a poor job of modeling the phone durations observed in natural speech. The transition probabilities have little impact on the final hypothesis produced by modern HMM-based recognizers, and some systems have even disregarded them altogether. In 1995, Siegler and Stern reported that the duration information derived from HMM transition probabilities does not correlate well with actual duration measurements, especially when speech rate becomes more rapid or more varied (Siegler & Stern, 1995). In Sections 2.6 and 2.7, we presented an overview of some previous approaches to incorporate explicit duration modeling information into the recognition framework.

There are two possible ways to alleviate the poor duration modeling problem. One is to modify the underlying modeling structure to capture duration information more accurately, which might necessitate an entirely different modeling framework. In this thesis work, we focus on the alternative: our goal is to modify the data so that it is more conducive to the underlying modeling framework of choice, *i.e.* the conventional HMM acoustic models.

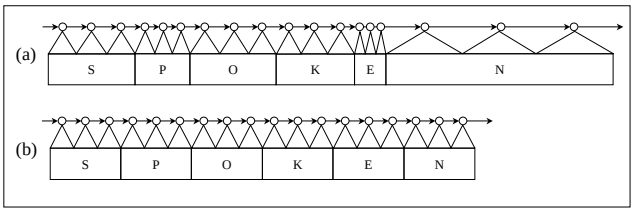

Figure 4.1 Illustration of the word "spoken" before (a), and after (b) duration normalization. Corresponding HMM states are shown above each phone segment and are mapped to the approximate phone region they model.

Figure 4.1 illustrates this duration normalization idea with durations abstracted from actual speech data. Continuous speech contains phones of varying duration. Each time a phone is uttered, it is produced with a different duration that depends on many different factors (*e.g.* phonetic context, speech register, speaking rate, emphasis). However, the underlying HMM that models all of the various phone renderings does a poor job of capturing duration information. Essentially, the HMM duration model is the convolution of the individual exponential duration distributions of each HMM state. This is a poor model of phone duration even if the number of states is chosen optimally for each phone. As seen in Figure 4.1(a), some HMM states model a relatively short amount of speech while others are forced to model many frames of speech data with a single Gaussian mixture. Figure 4.1(b) is a schematic illustration of speech that has been normalized so that every phone has the same duration. This makes the overall duration of a phone deterministic, retaining only the duration variations of the individual states within the phone. We hypothesize that duration normalization would result in reduced modeling variations across phones and improved recognition accuracy, especially for spontaneous speech where there is greater inherent variation of phone duration. This also ensures that each HMM state can characterize well the specific portion of the phone it is tasked to model.

4.1 Algorithm for Duration Normalization via Missing Feature Techniques

In our application, we wish to normalize the duration of each phone occurrence in the speech so that every instance of a phone has the same duration. Specifically, we normalize all instances of all phones to have the *same* duration. As hypothesized earlier, this restructuring is expected to result in an improvement in accuracy with HMM-based modeling. The true duration of a phone can differ from the desired normalized duration: a phone can have a greater duration than what we desire (a "long phone"), or it can have a smaller duration than what we desire (a "short phone").

If a given phone segment has a greater duration than the desired normalized duration, we downsample the observed frame sequence. Normalizing a long phone is illustrated in Figure 4.2(a). Note that missing

feature methods are not needed to accomplish this. However, if a phone has a duration that is less than the desired duration, we need a method for expanding its duration to the desired duration.

Missing feature methods, as discussed in Section 2.9, are traditionally used to reduce the impact on recognition accuracy of unreliable time-frequency locations in the feature space that represents the speech component of the signal. In particular, time-frequency locations that are corrupted by low SNR can be reconstructed based on information contained in other areas of the spectrogram which are assumed to be more reliable. The same reconstruction techniques can also be used to expand and recover the "missing" portions of the phones that have a smaller duration than the desired normalized duration.

Our approach is as follows: For a given short phone, we interleave a sequence of blank frames amid the observed frames so that the new phone duration is correct. We create a missing feature mask that declares our newly-inserted blank frames as "missing" and marks them for reconstruction. The missing frames of the short phones are then filled in using the correlation-based reconstruction method described in Section 2.9. The approach for normalizing short phones is illustrated in Figure 4.2(b). A detailed look at our implementation of this algorithm is presented in Section 4.2.

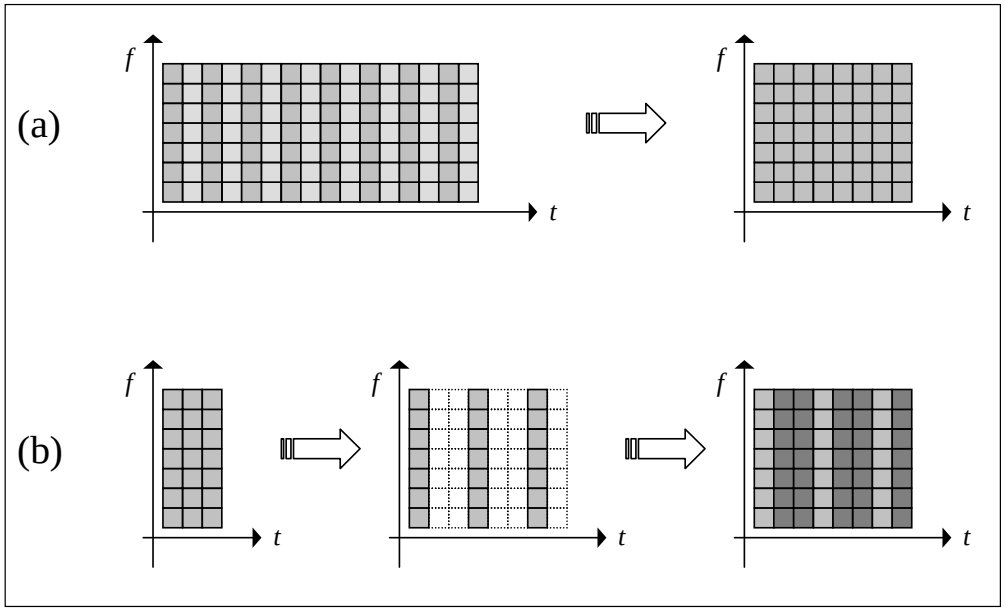

Figure 4.2 Illustration of the duration normalization process. The long phone shown in (a) is downsampled to the correct normalized duration. The short phone shown in (b) is expanded with frames of "missing" feature vectors and then filled in via missing feature reconstruction.

We note that all duration normalization and reconstruction is done in the log spectral domain, in the same manner that the corresponding operation is performed for traditional missing feature reconstruction. The resulting log spectral vectors are converted to Mel-frequency cepstral coefficients for use in training and

testing our standard HMM recognizer. Figure 4.3 shows the log spectrogram for an utterance both before and after duration normalization. (The figure shows is a Spanish utterance: “nove cientos euros y seis centimos”, which in English is “nine hundred euros and six cents”.)

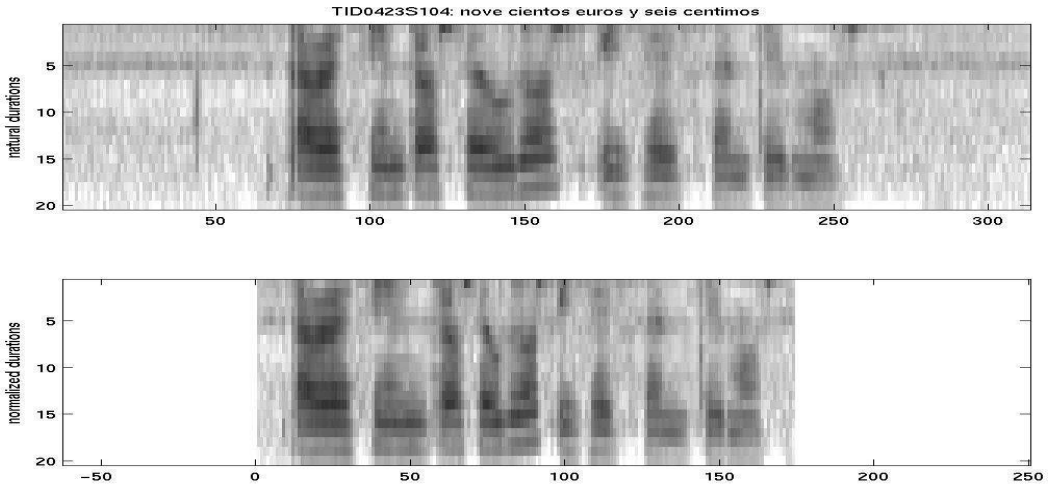

Figure 4.3 Log spectrograms of an example utterance before (top) and after (bottom) duration normalization.

Note that we have also experimented with simpler missing feature reconstruction methods, such as linear interpolation in time (which is the equivalent of simple time warping), to adjust the short phones to the correct duration. These methods resulted in no improvement in recognition accuracy. On the basis of these comparisons we believe that the added information contained in the correlations obtained from carefully-read speech allows us to regain some of the information that is lost when speech is produced very rapidly, as is often the case when speech is produced spontaneously.

4.2 Our Implementation of Missing Feature-Based Duration Normalization in Detail

In this section, we provide a detailed look at our implementation of time duration normalization using missing feature reconstruction. A functional overview of our system is illustrated in Figure 4.4.

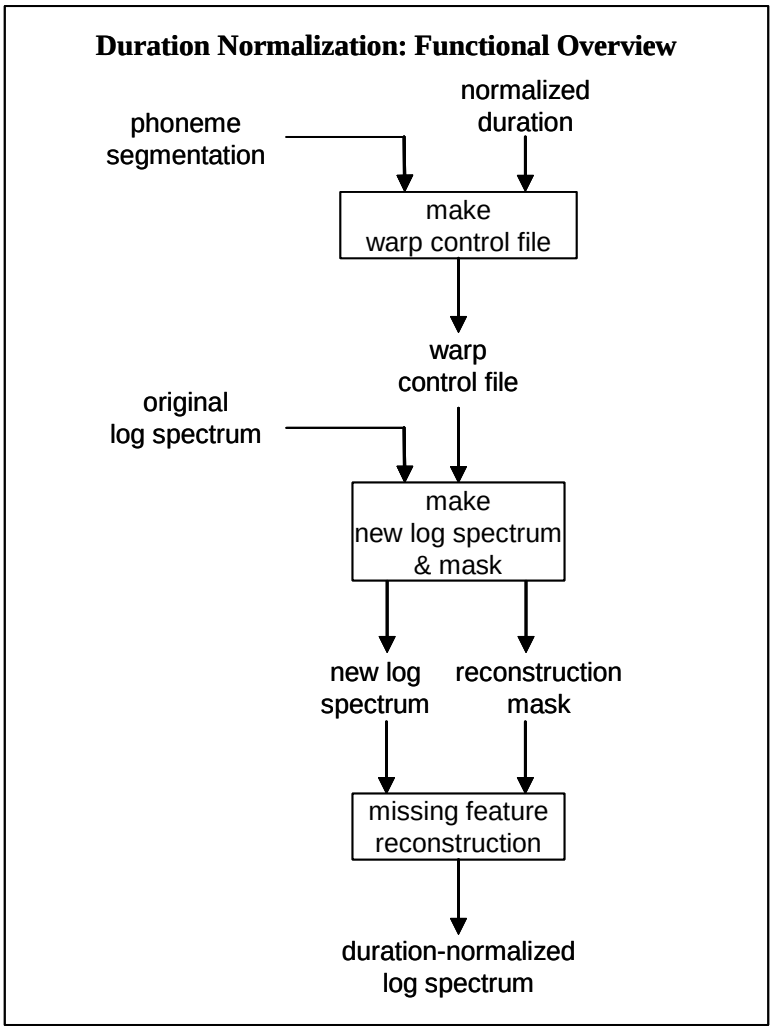

Figure 4.4 Detailed functional overview of duration normalization via missing feature methods.

The system has the following 3 main functional blocks:

- **make warp control file**: Creates a control file detailing which frames from the original log spectrum are kept and which frames are dropped. The locations of added "missing" frames are also included in the control file.
- **make new log spectrum & mask**: Using the warp control file and the original log spectrum, this module creates a new log spectral file containing only the information from the original log spectrum marked as "kept" by the control file. Space is also left in the new log spectrum for the added "missing" frames, and a mask is made to designate the newly added frames as missing.

- **missing feature reconstruction**: Covariance-based missing feature reconstruction is used to fill in the missing frames and generate a complete, duration-normalized log spectrum feature file. These log spectral features are finally converted to standard MFCCs for recognition.

The algorithm that controls the frame warping decisions is described in detail below, and following that is an illustrated example of the remainder of the process.

4.2.1 Warping: Deciding Which Frames Stay and Which Frames Go

To warp from the natural duration of a phoneme to the desired normalized duration, we designed a simple algorithm to add or drop the proper number of frames in an "even" spacing throughout a given speech segment. For example, if the original segment has 6 frames, and we want to compress it to 3 frames, our algorithm will specify that we keep frames 0, 2, and 4. Frames 1, 3, and 5 will be dropped.

For the purposes of this description, we assume our algorithm is performing a contraction in time. In practice, our algorithm treats all problems as contraction problems and fixes the resulting frame pattern at the end when expansion is required. (Note that when expanding a speech segment, we also desire an "even" spacing of frames, but this time we desire an even spacing of *inserted* frames rather than deleted frames.)

Our warping algorithm works as follows:

If there is only one frame to be deleted, the "middle" element of the frame sequence is deleted. If multiple frames must be deleted, we perform the contraction in two passes, a "keep" pass and a "delete" pass.

In the first pass, we choose to keep every k^{th} frame in the segment, where k is the ratio of the original duration of the segment to the normalized duration. All other frames are marked for deletion. Note that k must be an integer number of frames; therefore, there may be too many frames kept after the first pass. When this happens, a second pass is called upon to remove additional frames. In the second pass, we delete every j^{th} element from those that were originally kept, where j is the ratio of the number of frames kept in the first pass to the number of frames we still need to delete. Note that the "delete" pass terminates once we have achieved the desired number of frames.

Figure 4.5 illustrates an example of contracting from 7 frames to 3 frames. The dual example of expanding from 3 frames to 7 frames is also shown. In the figure, "X" represents the location of frames marked for deletion, and "–" represents the location where blank frames are to be inserted and later reconstructed by missing-feature techniques.

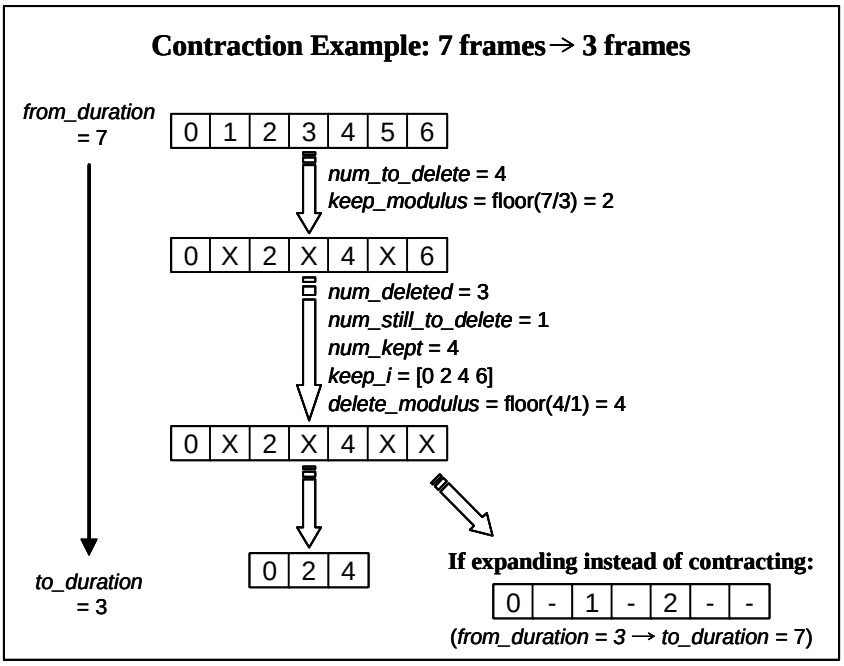

Figure 4.5 Illustration of contraction from 7 frames to 3 frames. (The corresponding pattern for expansion from 3 frames to 7 frames is also shown.)

4.2.2 Reconstruction: An Illustrated Example

Here we describe the remainder of the reconstruction process and illustrate it with an example chosen from the TID corpus. The example is the Spanish utterance: “nove cientos euros y seis centimos”, the same utterance shown previously in Figure 4.2.

Figure 4.6 illustrates the generation of the new log spectral file and reconstruction mask from the original log spectral file. The top panel shows the original log spectral file. The middle panel shows the new log spectral file, and the lower panel shows the corresponding reconstruction mask. This example is typical in that the normalized log spectrum has fewer frames than the original log spectrum. This is largely due to the fact that the long silence regions at the beginning and ending of each utterance are greatly compressed by the normalization process.

The corresponding reconstruction “mask” file is also shown at the bottom of the Figure 4.6. The reconstruction mask flags whether a pixel in the spectrogram should be kept (white) or disregarded and reconstructed (black). In our application, the mask is composed of vertical “stripes” because all of the log spectral values corresponding to a given speech frame are either wholly kept or wholly reconstructed.

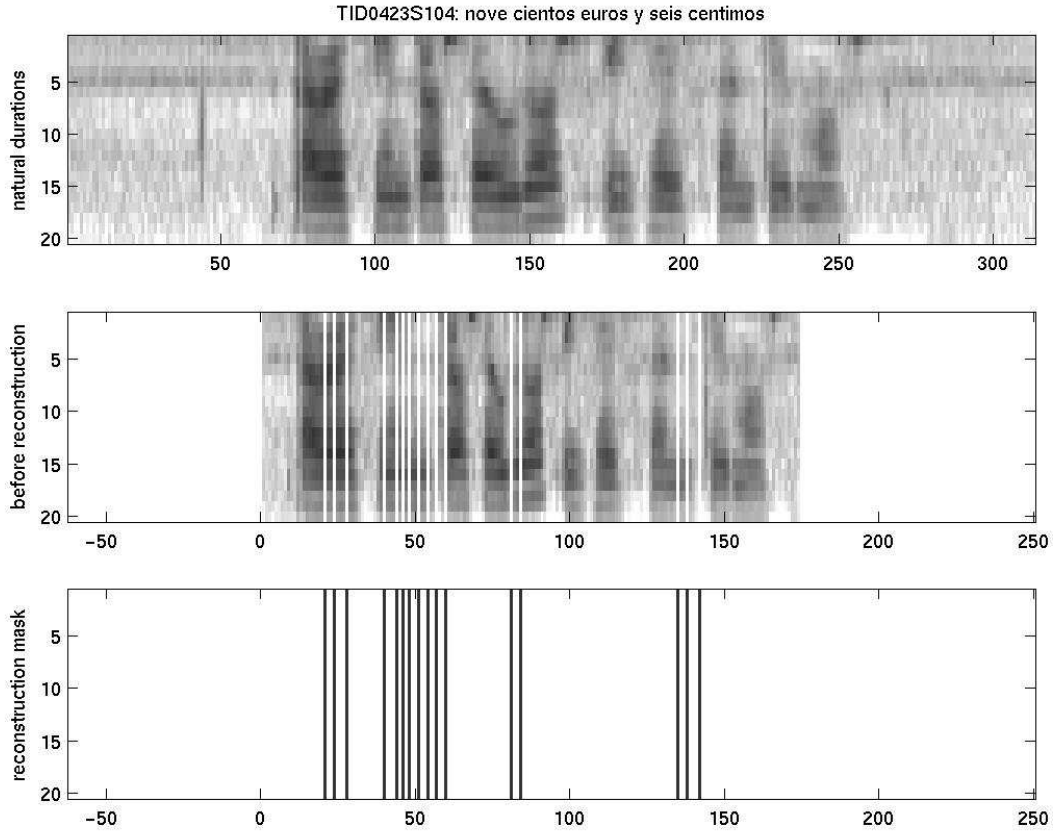

Figure 4.6 Original log spectral file (top) together with the new log spectral file (middle) and reconstruction mask (bottom).

Once the new log spectral file and corresponding reconstruction mask are generated, covariance-based missing feature reconstruction is performed to fill in the "missing" log spectral values, completing the duration normalization process. Figure 4.7 shows our example log spectral file before (top) and after (bottom) the missing vectors are reconstructed. The reconstruction mask is shown in the middle of the figure.

The theory behind covariance-based missing feature reconstruction is described in detail in Section 2.9. Note that in our experiments, the MAP estimate is computed to replace the missing elements in the spectrogram via the procedure termed *covariance joint reconstruction* (Raj, 2000). For computational efficiency, all of the missing values in the log spectrogram are not estimated at the same time; rather, the reconstruction is done on all the missing elements of a single log spectral vector, one frame at a time.

In our duration normalization application, all 20 log spectral elements of each inserted "missing" frame are reconstructed simultaneously using a maximum of 16 "neighbor" log spectral elements from the spectrogram. "Neighbors" are defined as the elements present in the log spectrogram with a relative covariance of at least 0.5 with at least one of the missing elements. Raj showed that this type of reconstruction is computationally efficient and accurate (Raj, 2000).

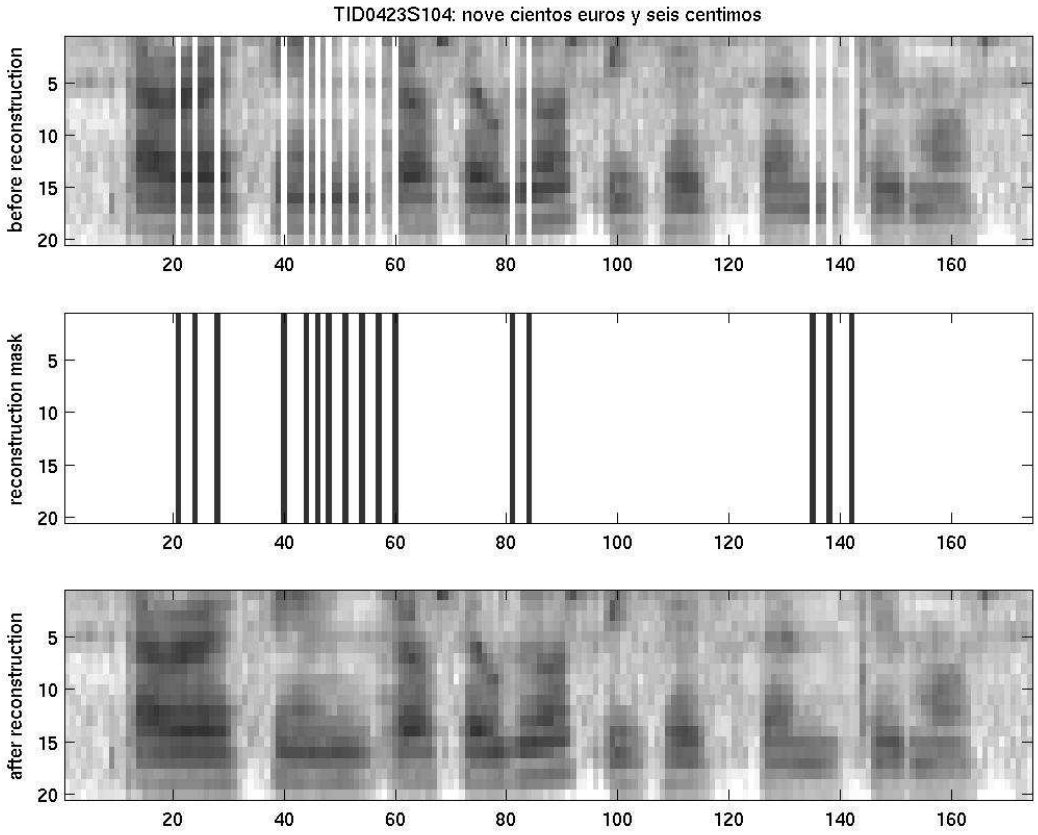

Figure 4.7 Log spectral file before (top) and after (bottom) reconstruction. The reconstruction mask (middle) is also shown.

4.2 Experiments Using Oracle Phone Boundaries

We started by training baseline models on each of the training sets using the standard approach. In order to apply missing feature based duration normalization, we needed to know the location of the phone boundaries in both the training and the testing sets. Using the baseline models and the reference transcripts, we performed a Viterbi alignment of the transcripts to the data and derived what we deemed our "oracle" phone boundaries. Viterbi alignment was performed on both the training and testing sets

used in the experiments. After alignment, however, the only information retained was the location of the boundaries that separate one phone from another.

The CMU SPHINX-III recognition system was used for all experiments. The data were modeled using 3-state left-to-right HMMs with no transitions permitted between non-adjacent states. For the smaller speech corpora, we used semi-continuous HMMs (codebook size 256) to model the data. For the large scale broadcast news data, we used fully-continuous HMMs with a mixture of 16 Gaussians per state.

4.2.1 Oracle Boundaries and the Multiple Register Corpus (MR)

For our first set of oracle boundary experiments, we used the NIST Multiple Register Speech Corpus (MR), a parallel corpus for comparison of spontaneous and read speech recorded at SRI. The database contains fifteen spontaneous conversations on assigned topics and re-read versions of the same conversations. For our experiments, we selected data from the read and spontaneous registers. We trained and tested separate models — one for read speech and the other for spontaneous speech. We used approximately 2 hours of speech to train each acoustic model, and 0.5 hours of speech to test each model. For more information on the MR database, see Section 3.2.2.

We first focused on the data taken from the spontaneous register of the corpus. Given the oracle phone boundaries, we applied the missing feature methods described in Section 4.1 to normalize all phone occurrences in the spontaneous speech data set to a specified frame duration. We then trained standard HMM models on the duration-normalized spontaneous training set and tested their accuracy on the duration-normalized spontaneous test set. For the baseline WER, we also decoded the test set using the standard models and natural duration speech features that were used to derive the oracle phone boundaries.

The normalized duration is a free parameter in this process; we can normalize each phone occurrence to any frame duration we choose. We empirically sought the optimal choice for the normalized duration by repeating the spontaneous speech experiment for several different normalized duration values (ranging from 4 frames to 12 frames). Note that at a normalized duration of 6 frames, the average HMM state in our 3-state models would be responsible for modeling approximately 2 frames of speech data. For a normalized duration of 9 frames, each state would be responsible for approximately 3 frames of speech data, and so forth.

Figure 4.8 plots the resulting accuracy of the duration-normalized models as a function of the chosen normalized frame duration. The baseline accuracy is plotted for reference as well. The baseline accuracy for the spontaneous test set was a word error rate of 40.3%. In the best case, when the speech was

normalized and reconstructed so that every phone had a duration of 8 frames, the resulting WER was 32.2%. This result showed a 20.1% relative improvement over baseline accuracy on spontaneous speech.

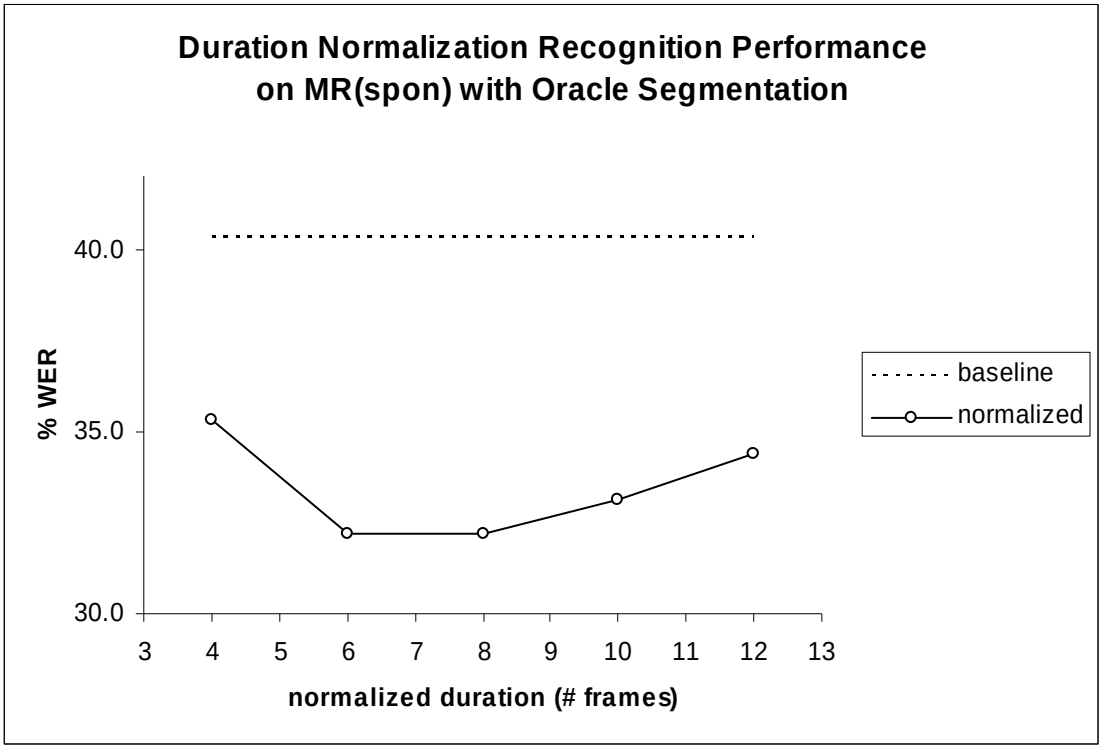

Figure 4.8 Results from phone duration normalization on MR spontaneous speech. WER is plotted as a function of the normalized phone duration. The baseline WER is also shown for reference.

Figure 4.8 also shows that a choice of normalized duration in the range of 6–8 frames is best for this particular data set. When expanding to 10 or 12 frames, it is possible that correlation-based reconstruction cannot adequately estimate the missing frames. Prior experiments have indicated missing-feature reconstruction methods are only effective if the sequences of missing frames being reconstructed are no more than 5 frames long (Raj, 2000). If we expand a very short phone, say 3 frames, up to a duration of 12 frames, the missing feature methods are required to reconstruct 3 "missing" frames in a row three times in a row, with only one frame of information in between.

We then repeated the same experiment on speech taken from the read register of the MR corpus. We used the oracle phone boundaries to normalize the duration of each phone to 8 frames. We again trained standard HMMs on the duration-normalized read training set and evaluated our models on the duration-normalized read testing set. The results are shown in Table 4.1.

	WER	Relative Improvement
Baseline	15.6%	-
Normalized duration (8 frames)	14.0%	10.3%

Table 4.1 Results from phone duration normalization on MR read speech. A 10.3% relative improvement over baseline accuracy is shown when all phones are normalized to a duration of 8 frames.

We observed that our baseline error rate of 15.6% was reduced to 14.0% when missing feature duration normalization was applied to read speech. This reflected a relative improvement of 10.3% over baseline accuracy. These results show that the duration normalization methods are effective with perfect knowledge of segment boundaries for carefully enunciated speech and for spontaneous speech.

4.2.2 Oracle Boundaries and the Telefónica Corpus (TID)

We also conducted oracle experiments on a Spanish database recorded by Telefónica Investigación y Desarollo in Madrid, Spain. The database consists of cellular telephone callers repeating a small string of digits or a monetary amount. The speech is small vocabulary, but highly spontaneous. The training set consists of approximately 4 hours of training speech and 2 hours of testing speech data. For more information on the TID corpus, see Section 3.2.1.

The process was the same as that for MR: We trained and tested standard HMMs on the raw TID speech. We then Viterbi aligned the speech as before to derive oracle segmentation information. We then duration normalized the entire train and test sets, and repeated our training and testing on the Spanish speech. The results are shown in Table 4.2.

	WER	Relative Improvement
Baseline	5.2%	-
Normalized duration (6 frames)	3.4%	34.6%

Table 4.2 Results from phone duration normalization on spontaneous Spanish TID speech.

Note that for the Spanish TID data, we empirically determined that a normalized duration of 6 frames is best. We observed in this case that our baseline error rate of 5.2% was reduced to 3.4%, which reflected a relative improvement of 34.6%. These results confirm the potential effectiveness of the missing feature duration normalization approach. They also indicate that there is a great potential for improved recognition accuracy, especially in the case of smaller vocabulary and limited domains.

4.2.3 Oracle Boundaries and the Broadcast News Corpus (BN)

We also conducted oracle experiments on a the NIST HUB4 Broadcast News evaluation data. This database consists of televised broadcast news collected in the mid to late 1990s. Model training was performed on 45 hours of speech taken from the 1996 and 1997 corpora. Testing was done on the 1999 Eval 1 data set. For more information on the BN corpus, see Section 3.2.3.

The procedure was identical to the procedure used for the MR and TID data sets. For this English broadcast news data, a normalized duration of 8 frames was used. The results are shown in Table 4.3.

	WER	Relative Improvement
Baseline	33.4%	-
Normalized duration (8 frames)	31.6%	5.4%

Table 4.3 Results from phone duration normalization on large-scale broadcast news task.

For the broadcast news task, the baseline error rate of 33.4% was reduced to 31.6% via the duration normalization algorithm. This is a relative reduction in WER of 5.4%. These results further confirm the effectiveness of recognizing speech with normalized phone durations. As with most large-scale tasks, the potential for improvement is not as great as that achieved for smaller tasks. This is most likely due to the large amount of training data and complexity of the models that can be derived from such a data set.

4.2.4 Result Summary: Duration Normalization with Oracle Segmentation Information

We close this chapter with a summary of the results from applying the duration normalization algorithm with oracle segmentation information. Table 4.4 contains a summary of recognition accuracy improvements possible for each of the databases tested.

Corpus	Relative Improvement
MR (spon)	20.1%
MR (read)	10.3%
TID (Spanish)	34.6%
BN	5.4%

Table 4.4 Summary of phone duration normalization results using oracle segmentation on a variety of speech corpora.

From these results, we observe that the potential accuracy improvements varies depending on the size of the task and the nature of the speech. The results from the parallel MR corpus indicate that the potential improvement is greater when the speech is more spontaneous than when the speech is more carefully prepared and read.

The broadcast news speech contains a large amount of speech data with varying levels of spontaneity, from carefully-prepared and professionally-delivered news reports to *ad hoc* interviews in the field with background noise and other issues. With this large amount of training data and more sophisticated recognition models, the potential for improvement with duration normalization, while still significant, is not as great as the potential improvement for other tasks.

4.3 Conclusions

In this chapter, we presented a detailed overview of our duration normalization process which uses missing feature reconstruction techniques to enable the normalization of the duration of all sound units present in the speech prior to modeling and recognition. This normalization is designed to "factor out" the phone duration variability and help ensure robust estimation of the HMM acoustic model parameters despite the high duration variability observed in spontaneous speech data.

Using the correct transcripts, we used Viterbi alignment to generate "oracle" segmentation information for use with our duration normalization algorithm. Experiments on the spontaneous register of the MR corpus indicated that a normalized duration of 8 frames led to the best overall recognition system accuracy, and therefore we fixed our normalized duration to 8 frames for all English language corpora for the remainder of this thesis research. (Similar experiments on the Spanish language TID corpus indicated a normalized duration of 6 was better for spontaneous Spanish speech; therefore, we fix the normalized duration to 6 frames for TID.)

When the segmentation information is known *a priori*, we observed that the duration normalization algorithm yields large improvements in recognition system accuracy, with relative reductions in word error rates in the range of 5.4%–34.6%. The potential for improvement varied depending on the size of the corpus and the size of the vocabulary. Consistent with many speech recognition enhancement algorithms, the observed accuracy improvements were quite large on smaller datasets and more modest on larger datasets presumably due to the large amount of varied speech data used to train the HMM acoustic models.

In the following chapter, we present different techniques for automatic segmentation of speech into sound units. We focus our search on finding the most effective speech segmentation technique that yields significant recognition system accuracy improvements when coupled with our duration normalization algorithm.

5: Blind Phone Segmentation Techniques

In this chapter, we discuss automatic techniques to segment the speech waveform into a sequence of sound units ("phones") when the transcript is not known *a priori*. The duration normalization technique described in the previous chapter depends on the quality of an automatically-derived segmentation of speech into basic phonetic units. We also discuss and apply some metrics from signal detection theory to evaluate the quality of the proposed automatic segmentation techniques.

5.1 Decoder-based Segmentation

A simple way to segment the speech waveform into sound units makes use of the speech recognition engine and HMM acoustic models for the phonetic units. This process is illustrated in Figure 5.1.

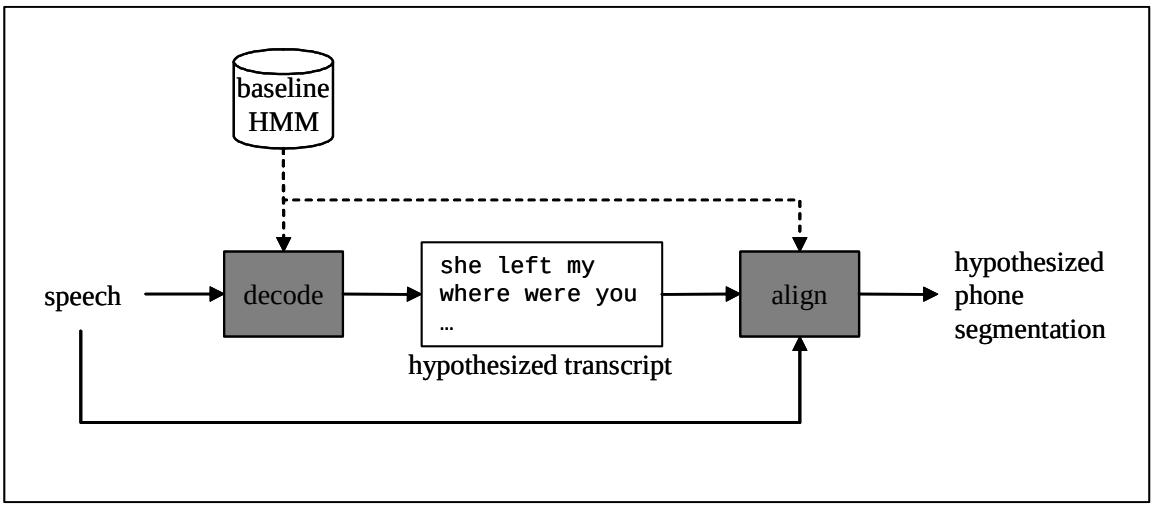

Figure 5.1 Block diagram for the decoder-based segmentation system.

We start by training baseline recognition models on the training speech corpus using standard Baum-Welch training. We then attempt to recognize the speech using the baseline models. The decoder produces a hypothesized transcript of each utterance in the corpus. We then use the Viterbi algorithm and the baseline acoustic models to align the hypothesized transcripts to the speech. Boundary locations are hypothesized every time the aligner records a transition out of the last state of one phone's acoustic model and into the first state of the next phone's acoustic model.

5.2 Experimental Results Using Decoder-based Segmentation

We performed duration normalization experiments on the Telefónica (TID), Multiple Register (MR), and Broadcast News (BN) corpora using decoder-based segmentation. In each case, we trained baseline

models for the particular corpus. We then decoded and Viterbi-aligned both the training and testing sets to generate hypothesized phone segmentations for each complete corpus. We performed duration normalization on the training set of each corpus using the hypothesized segmentation information. We then trained HMM acoustic models on the test corpora. Finally, we performed duration normalization on the testing sets and decoded the speech. The results are summarized in Table 5.1.

	TID	**MR**	**BN**
baseline	5.2%	40.3%	33.4%
duration normalization using decoder-based segmentation	5.4%	39.8%	36.0%
duration normalization using oracle segmentation	3.2%	32.2%	31.6%

Table 5.1 Duration normalization results on three corpora using decoder-based segmentation. Baseline and oracle segmentation results are presented for reference.

It is clear from these results that decoder-based segmentation is insufficient for use with the duration normalization algorithm. In 2 of the 3 databases tested, accuracy actually degrades with respect to baseline when duration normalization is applied using the decoder-based segmentation.

5.3 Signal Detection Theory: ROCs and the *d′* Sensitivity Metric

In order to properly evaluate and compare different segmentation techniques, we make a short digression to discuss some fundamental notions of signal detection theory (Engen, 1971). We consider the speech segmentation problem as a detection problem where we are trying to detect phone boundary locations within a speech signal. This is a two-class pattern recognition problem: the detector must decide whether a frame of speech data corresponds to a true boundary location (**"T"**) or a false, non-boundary location (**"F"**).

When attempting to automatically detect boundaries, there are four possible situations that can arise. These situations are defined and described using standard signal detection theory terminology as follows:

1. "hit" = a true boundary location (**T**) is correctly identified as a boundary (**"T"**)
2. "miss" = a true boundary location (**T**) is incorrectly identified as a non-boundary (**"F"**)
3. "false alarm" = a non-boundary location (**F**) is incorrectly identified as a boundary (**"T"**)
4. "correct rejection" = a non-boundary location (**F**) is correctly identified as a non-boundary (**"F"**)

A good detector will maximize the number of "hits" and "correct rejections" while minimizing the number of "misses" and "false alarms".

Many of our segmentation systems make use of a decision threshold (θ) when hypothesizing speech segmentations. We evaluate our detection systems by purposefully varying the decision threshold and recording the resulting probability of correct detection and probability of false alarm for each value of θ. Given this information, we plot the Receiver Operating Characteristic (ROC), with the probability of false alarm on the *x*-axis and the probability of correct detection on the *y*-axis. In evaluating different segmentation algorithms, we will report results together with ROC graphs and estimated sensitivity parameters where appropriate.

To evaluate the accuracy of a detector, it is important to separate the *sensitivity* (d') of the detector from its *bias* (β). These measures are derived by assuming that the **T** and **F** detection classes are governed by underlying normal distributions with respective means m_T and m_F, and equal standard deviations $\sigma_T = \sigma_F = \sigma$. This is illustrated in Figure 5.2.

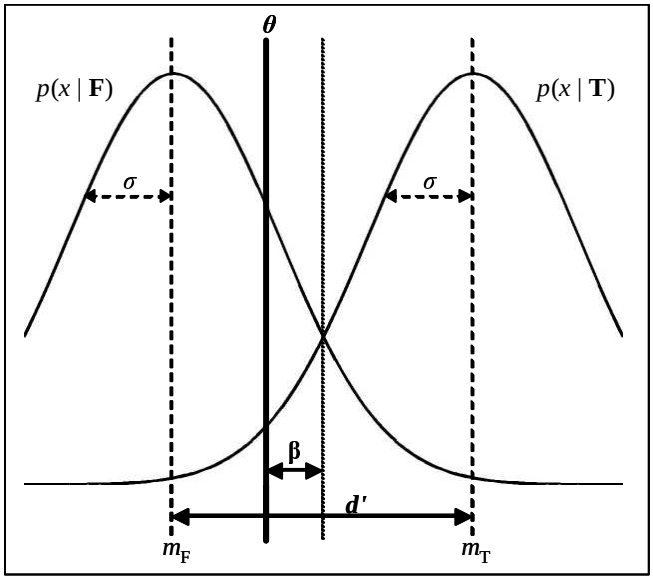

Figure 5.2 Illustration of detector sensitivity (d') and bias (β) for a two-class problem with underlying normal probability distributions. The d' shown in the figure assumes that the standard deviation (σ) of both classes is 1.

The sensitivity measure d', which is independent of the decision threshold θ, is given by the difference between the means of the two classes divided by the standard deviation:

$$d' = \frac{m_T - m_F}{\sigma} \qquad (5.3.1)$$

The bias β is the difference between the decision threshold θ and the midpoint between the two means:

$$\beta = \theta - \frac{m_T + m_F}{2} \tag{5.3.2}$$

In practice, the decision threshold θ is adjusted according to the *a priori* statistics of the two classes and the costs associated with each type of detection error (misses v. false alarms). This adjustment is a purposeful bias of the detector.

To evaluate the accuracy of a classifier on a given test set, we must first estimate the detector's probability of correct detection (P_D) and probability of false alarm (P_F). We estimate P_D by computing the ratio of the number of hits to the total number of boundary locations in the corpus, and we estimate P_F by computing the ratio of the number of false alarms to the total number of non-boundary locations in the corpus. For ease of computation, we then convert the estimated P_D and P_F values to *z* scores using the following coordinate transforms:

$$P_D = \frac{1}{2\pi} \int_{-\infty}^{z_D} e^{-x^2/2} dx \tag{5.3.3}$$

$$P_F = \frac{1}{2\pi} \int_{-\infty}^{z_F} e^{-x^2/2} dx \tag{5.3.4}$$

Finally, the sensitivity parameter d' and detector bias β are calculated using the following formulas:

$$d' = z_D - z_F \tag{5.3.5}$$

$$\beta = \frac{z_D + z_F}{2} \tag{5.3.6}$$

The graph in Figure 5.3 shows different ROC curves that correspond to particular values of the sensitivity parameter d'. These curves are known as *isosensitivity ROC curves* because the value of d' is the same for every point on the curve. A sensitivity value of $d' = 0$ corresponds to chance accuracy. The greater the value of d', the more accurate the system is. Note that the accuracy of an unbiased detector would lie on the "knee" of each curve. The more biased a detector is, the further away the operating point lies from the knee of its corresponding isosensitivity curve.

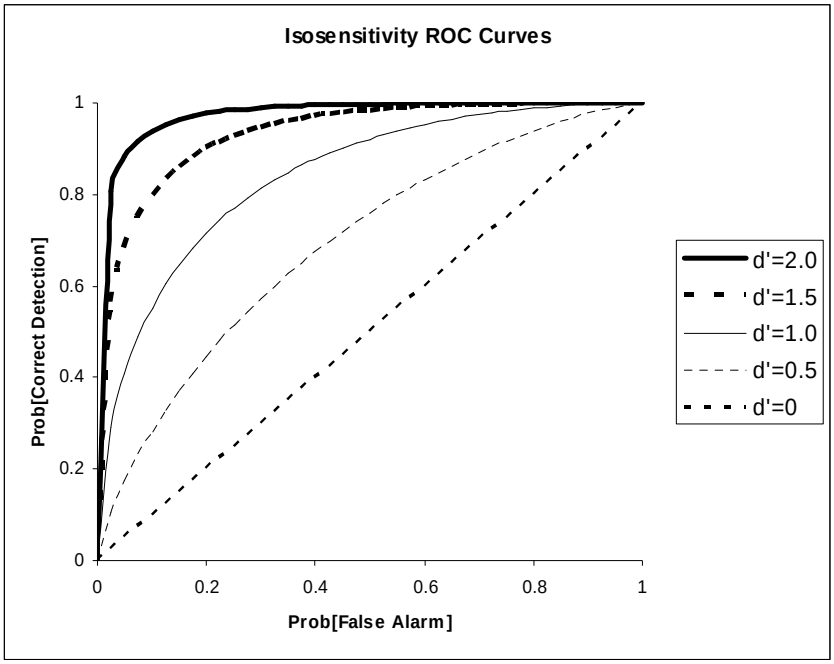

Figure 5.3 Example isosensitivity ROC curves for different values of the sensitivity measure *d′*.

The graphs in Figure 5.4 show the relationship between the probability of correct detection P_D and the sensitivity parameter *d′*. To generate these curves, we assume that the detector is unbiased, *i.e.* that P_F is always equal to $1 - P_D$ Note that *d′* approaches infinity as P_D approaches 1.0.

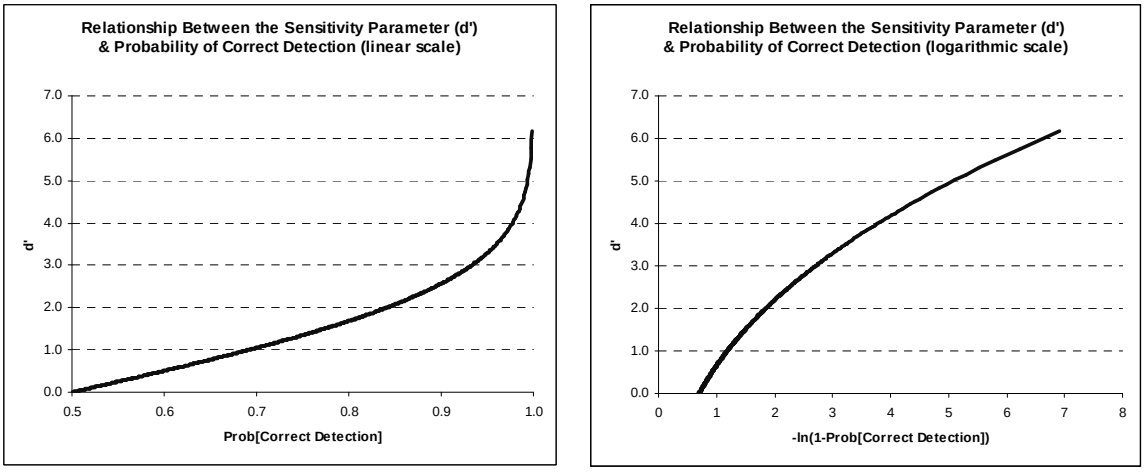

Figure 5.4 Relationship between the sensitivity measure *d′* and the probability of correct detection, assuming that the classifier is perfectly unbiased. The plot on the left uses a linear scale, and the plot on the right uses a semi-log scale.

5.4 Results and Analysis: Decoder-based Segmentation

In this section, we report segmentation results for decoder-based segmentation performed on the test set of each of the three evaluation corpora (TID, MR, BN).

The decoder-based segmentation results for the TID test corpus are shown in Table 5.2. The first chart shows the raw count in number of frames processed and detected. The second chart shows the results as a percentage. The sensitivity index d' is also given.

	"T"	"F"
T	43126	883
F	1308	675876

	"T"	"F"
T	98.0%	2.0%
F	0.2%	99.8%

$d' = 5.0$

Table 5.2 Decoder-based segmentation detection results for the TID corpus.

It is clear from these results that the decoder-based segmentation performs very well on the TID test data, with a 98% probability of correct detection and only a 0.2% probability of false alarm. The sensitivity index d' is 5.0 for this detector.

Decoder-based segmentation on the spontaneous MR test set performs as follows (Table 5.3):

	"T"	"F"
T	9489	1452
F	985	104143

	"T"	"F"
T	86.7%	13.3%
F	0.9%	99.1%

$d' = 3.5$

Table 5.3 Decoder-based segmentation detection results for the MR corpus.

While the classification is not as accurate on MR as it is on TID, the classifier still performs with a high sensitivity index of 3.5. Decoder-based segmentation achieves an 86.7% hit rate with a 0.9% rate of false alarm on the conversational MR data. It is clear from these results that decoder-based segmentation accuracy is related to the WER of the standard baseline models when decoding the test set. For TID, baseline WER is 5.2%, while for MR, the baseline WER is 40.3%. Consequently, the decoder-based segmentation is less accurate on the MR data.

Decoder-based segmentation and the BN corpus yields the following results (Table 5.4):

	"T"	"F"
T	41112	3139
F	2320	305653

	"T"	"F"
T	92.9%	7.1%
F	0.7%	99.3%

$d' = 3.9$

Table 5.4 Decoder-based segmentation detection results for the BN corpus.

Again, the decoder-based segmentation performs quite well on BN data. The sensitivity index is 3.9 with a hit rate of 92.9% and a false alarm rate of 0.7%. We also note that the accuracy of decoder-based segmentation again relates to the baseline word error rate of the recognition system. The BN baseline WER of 33.4% is better than the baseline WER for MR but worse that the baseline for TID. Accordingly, the decoder-based segmentation accuracy is better than that of the MR system but worse than that of the TID system. In all cases, the decoder-based segmentation yields strong sensitivity indices ranging from 3.5 to 5.0.

5.4.1 Decoder-Based Segmentation—Detector Bias

The decoder-based segmentation results presented above indicate that the detection system is biased, *i.e.* the probability of missing a boundary is much greater than the probability of false alarm. For TID, we estimate a detection bias β of 0.412. For MR the detection bias β is 0.626, and for BN the detection bias β is 0.494. We therefore investigated the possibility of adjusting the speech recognizer so that the resulting segmentation results would be less biased.

The SPHINX-III speech recognition system offers two adjustable parameters to optimize accuracy: the *word insertion penalty* and the *phone insertion penalty*. These penalties are incurred during the search every time the recognizer hypothesizes a transition into a new word or phoneme. The word insertion penalty is designed to favor hypotheses with a "reasonable" number of words given the overall duration of the utterance, and most state-of-the-art HMM-based systems make use of such a parameter.

We experimented with a variety of word and phone insertion penalty values on the MR corpus and found that we were able to move the operating point of the detector only slightly towards the desired unbiased operating point. The observed change was too small to have a significant impact on our recognition

results. After exhausting the practical means to reduce the bias of the decoder-based segmentation system, we concluded that the effects of the bias were unavoidable.

5.5 Phonetic Decoder-based Segmentation

At the end of this chapter, we discuss in detail a fundamental problem that results when decoder-based segmentation is used together with duration normalization (see Section 5.7 "The Decoder-based Segmentation Dilemma"). It would be advantageous to develop a high quality segmentation algorithm which does not depend on the sequence of word strings hypothesized by the baseline recognition system.

We experimented with using the acoustic recognition models to search for the most likely sequence of phones present in the speech signal. This is known as phonetic or "all phone" decoding. The phonetic decoding search is not constrained by the dictionary of words and their corresponding "valid" sequences of phones in the language. Unlike word decoding, when SPHINX-III is used in "all phone" mode, the system outputs the hypothesized sequence of phonemes and the corresponding start and end frames for each hypothesized phone. Using the baseline recognition models trained for each corpus, we performed phonetic decoding of the TID corpus.

Phonetic decoder-based segmentation on the TID test set performs as follows (Table 5.5):

	"T"	**"F"**
T	40856	3153
F	7479	669705

	"T"	**"F"**
T	92.8%	7.2%
F	1.1%	98.9%

$d' = 3.8$

Table 5.5 Phonetic decoder-based segmentation detection results for the TID corpus.

In this instance, the segmentation derived by the phonetic recognition system performs worse than the segmentation derived by the word recognition system. Similar results were observed on MR data. This is due to the fact that the phonetic decoding is often error prone, and unconstrained phone error rates of the systems are significantly worse than corresponding word error rate values for constrained word decoding.

We attempted to use all phone decoding to segment speech for duration normalization, but the technique was not successful. The accuracy of all phone decoder-based segmentation duration normalization was always worse than baseline accuracy due to the large number of phone and boundary errors introduced by the lower quality phonetic decoding.

5.6 Signal Processing-based Segmentation Techniques

We also investigated and evaluated a series of segmentation techniques that work directly on the speech signal, independent of the recognition engine. We experimented first with an "edge detection" technique where we assigned boundary locations to places in the signal where the spectrum changed dramatically using a variety of distortion metrics. We then moved to a more elaborate "split-and-merge" algorithm to find regions of spectral stability within the speech signal.

5.6.1 Edge Detection Segmentation

In edge detection segmentation, we analyze the speech signal and look for locations in the speech where the signal is changing rapidly. We first convert the speech signal to the 20-dimensional log Mel spectral domain. This gives a time sequence of 20-dimensional vectors we call $\mathbf{x}[n]$. We then calculate a running distortion metric (Δ) across the time sequence of speech log-spectral vectors and compare to a decision threshold θ. If Δ is greater than θ, we hypothesize a boundary location at that point in the signal; otherwise, we assign that frame of speech to the non-boundary class.

We experiment with the following three distortion metrics for edge detection segmentation. Note that $d(\mathbf{x}, \mathbf{y})$ represents the standard Euclidian distance between vectors $\mathbf{x}$ and $\mathbf{y}$.

1. Backward Difference:

$$\Delta[n] = d\left(\mathbf{x}[n], \mathbf{x}[n-1]\right) \tag{5.6.1}$$

2. Forward + Backward Difference:

$$\delta_{\text{backward}}[n] = d\left(\mathbf{x}[n], \mathbf{x}[n-2]\right) \tag{5.6.2}$$

$$\delta_{\text{forward}}[n] = d\left(\mathbf{x}[n], \mathbf{x}[n+2]\right) \tag{5.6.3}$$

$$\Delta[n] = \delta_{\text{forward}}[n] + \delta_{\text{backward}}[n] \tag{5.6.4}$$

3. Dendrogram-based Distortion Metric

$$a[n] = \delta_{\text{forward}}[n] - \delta_{\text{backward}}[n] \tag{5.6.5}$$

$$\Delta[n] = \begin{cases} \left| a[n] - a[n-1] \right| & , \text{ if } a[n] \text{ and } a[n-1] \text{ differ in sign} \\ 0 & , \text{ otherwise} \end{cases} \tag{5.6.6}$$

The backward distortion (Eq. 5.6.1) is the simplest distortion metric which looks for immediate spectral change within a range of one frame of speech. The forward + backward distortion metric (Eq. 5.6.4) looks for locations in the speech signal which are vastly different than those that come 2 frames earlier and those that come 2 frames later. The idea is that at boundary locations, the speech frame should look different from its neighbors in either direction.

The dendrogram-based distortion metric is similar to the metric used by segmental modeling/recognition systems when they build a "dendrogram" or hierarchical segmentation network for the speech signal (Glass & Zue, 1988). When building a dendrogram, each frame of speech is "associated" either with the frames preceding the given frame or the frames following the given frame depending on the association direction with the smaller distortion. In Eq. 5.6.5, we define $a[n]$ as an "association" metric corresponding to the speech signal. If $a[n]$ is positive, then the forward distortion is greater than the backward distortion, which means that frame n is associated with the preceding frames. If $a[n]$ is negative, then the backward distortion is the greater distortion, implying an association with the following frames. Since boundaries are hypothesized in the base level of a dendrogram network whenever the association direction changes, we define the final dendrogram-based distortion metric in Eq. 5.6.6 as follows: $\Delta[n]$ is the magnitude of the association change when there is an association change (*i.e.* when $a[n]$ and $a[n–1]$ differ in sign), and $\Delta[n]$ is 0 when there is no association change.

The ROC results for the backward distortion metric are presented in Figure 5.5 for TID and MR data. Note that the accuracy for decoder-based segmentation is presented with symbol $\times$ for comparison. The sensitivity index for backward difference edge detection is approximately 1.9 for TID and 2.0 for MR data. It is clear that the decoder-based segmentation outperforms backward difference edge detection.

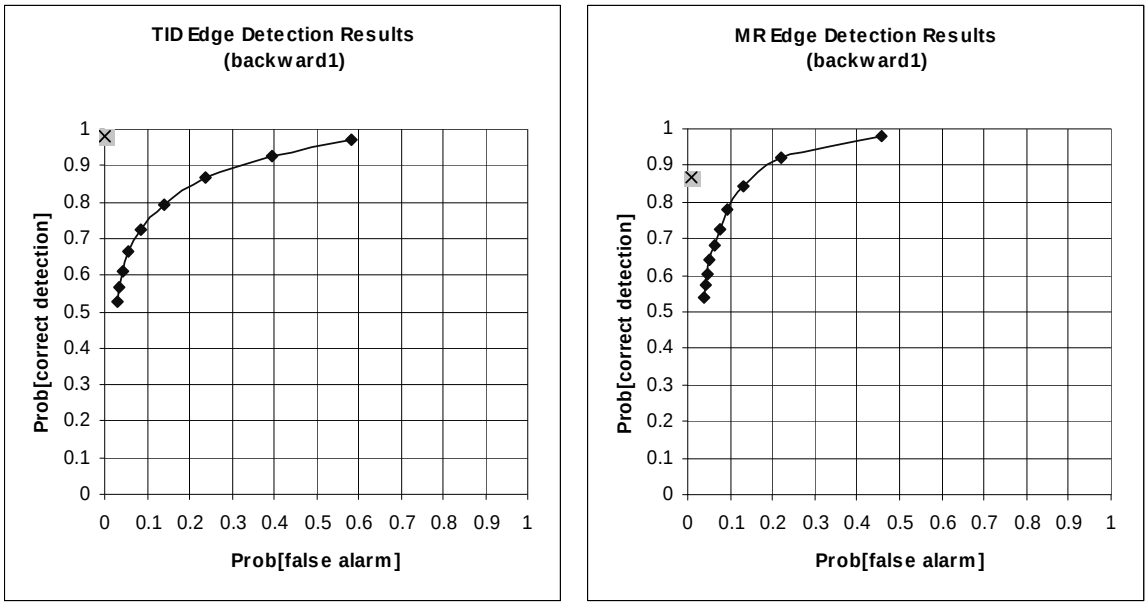

Figure 5.5 ROC results for edge detection using the backward distortion metric on TID (left) and MR (right). Decoder-based segmentation is shown as an "X" for reference.

Forward + backward difference edge detection ROC for TID and MR data are shown in Figure 5.6. Again, the decoder-based segmentation accuracy is presented for reference.

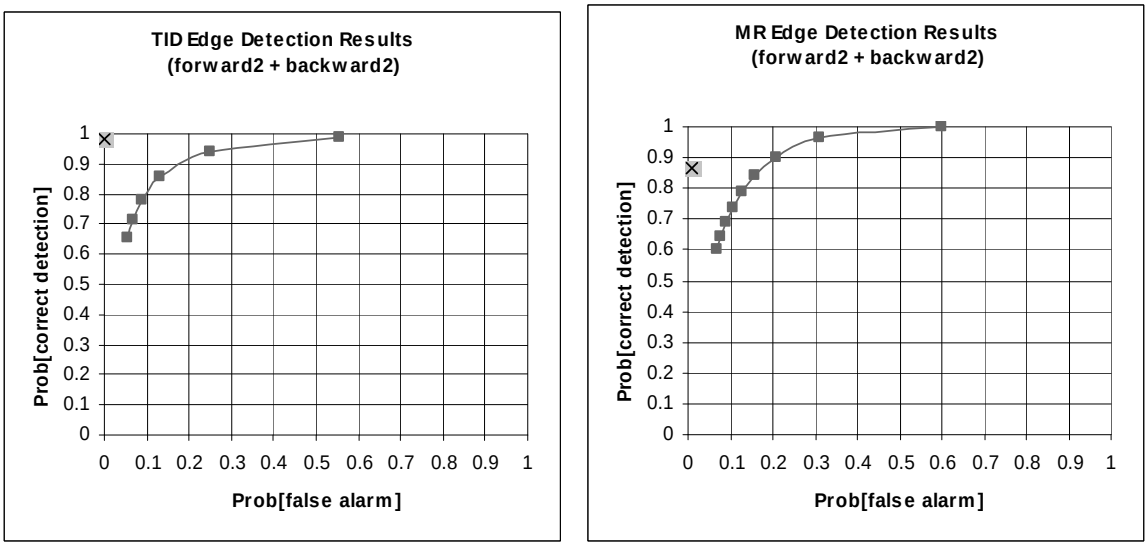

Figure 5.6 ROC results for edge detection using the forward and backward distortion metric on TID (left) and MR (right). Decoder-based segmentation is shown as an "X" for reference.

Sensitivity index d' values of approximately 2.1 and 1.9 are observed on TID and MR data respectively. Again, decoder-based segmentation outperforms forward+backward difference edge detection on both data sets.

The results for the dendrogram-based edge detection are shown in the following ROCs (Figure 5.7):

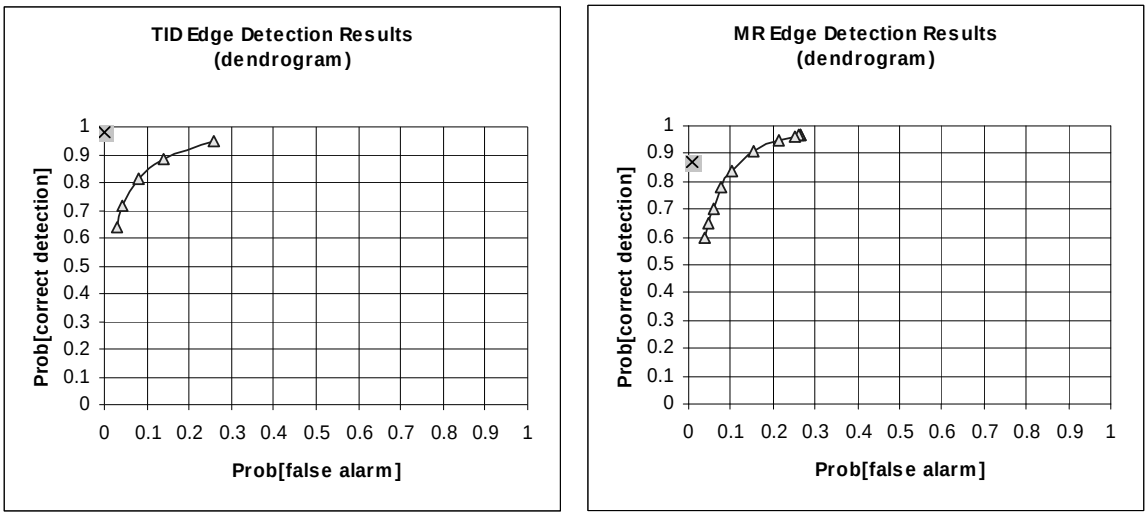

Figure 5.7 ROC results for edge detection using the dendrogram-based distortion metric on TID (left) and MR (right). Decoder-based segmentation is shown as an "X" for reference.

For dendrogram-based edge detection, the d' values are approximately 2.3 for both TID and MR. Although we see a slight improvement over the other two metrics, we still observe that the decoder-based segmentation greatly outperforms our edge detection techniques.

In summary, we present overlapping ROC curves for direct comparison of the different edge-detection distortion metrics (Figures 5.8 and 5.9). We also present a chart of sensitivity index (d') values for each edge-detection technique as well as for the decoder-based segmentation (Table 5.6).

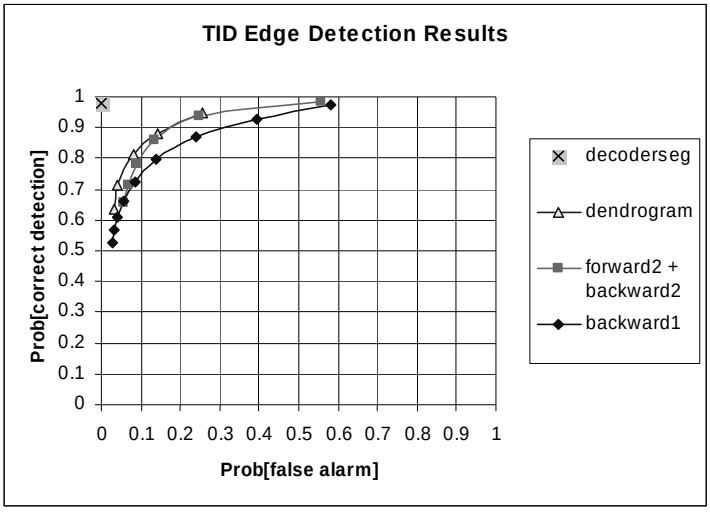

Figure 5.8 Summary ROC results for edge detection using the different distortion metrics on the TID corpus. Decoder-based segmentation is shown as an "X" for reference.

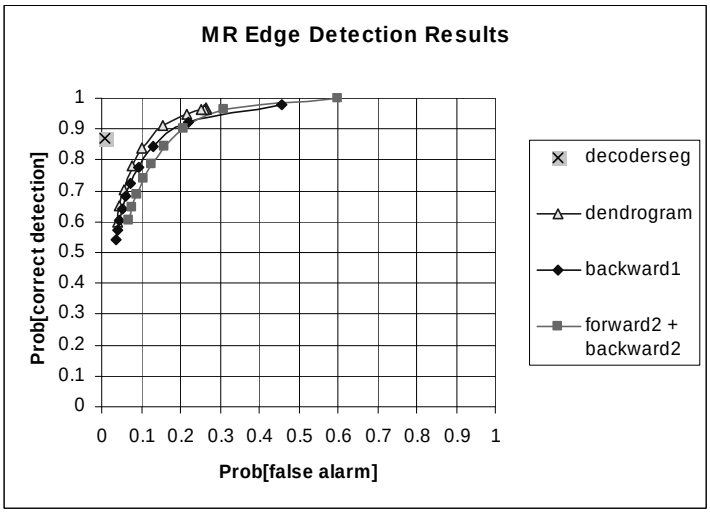

Figure 5.9 Summary ROC results for edge detection using the different distortion metrics on the MR corpus. Decoder-based segmentation is shown as an "X" for reference.

Sensitivity Index Values (*d'*)	**TID**	**MR**
decoder-based segmentation	5.0	3.5
backward difference edge detection	1.9	2.0
forward+backward difference edge detection	2.1	1.9
dendrogram-based edge detection	2.3	2.3

Table 5.6 Summary sensitivity index values for edge detection using the different distortion metrics on the TID and MR corpora. Decoder-based segmentation is shown for reference.

For both TID and MR data, the dendrogram-based edge detection is the best of the edge detection methods. However, in both cases, the decoder-based segmentation approach presented in Section 5.1 is clearly superior. For TID, the edge detection methods would incur a probability of false alarm greater than 0.5 to achieve the probability of correct detection that decoder-based segmentation achieves. For MR, the edge detection methods would have a false alarm probability of 0.1 when the decoder-based segmentation hit probability is achieved.

5.6.2 "Split-and-Merge" Segmentation

Image processing researchers often use a technique known as "split-and-merge" segmentation to automatically locate distinct regions or "objects" within a given image (Horowitz & Pavlidis, 1974). With a few modifications, we can apply split-and-merge to the phonetic segmentation problem in speech.

We start with a spectrogram representation of the speech signal, which for our experiments is a time sequence of 20-dimensional log Mel spectral vectors ($\mathbf{x}[n]$). An example log spectrogram from TID data is shown in Figure 4.2.

We constrain our search for regions within the speech image to blocks that span the entire vertical axis in order to look for time sequences with similar spectral characteristics. We then proceed to break the image up into distinct regions with a small variability in spectral features. We define the mean vector and variability corresponding to a speech region as follows:

$$\text{mean vector:} \qquad \bar{\mathbf{x}}\Big|_{\text{start n}}^{\text{end n}} = \frac{1}{N}\sum_{i=\text{start n}}^{\text{end n}} \mathbf{x}[i] \tag{5.6.7}$$

$$\text{variability (scalar):} \qquad \sigma^2\Big|_{\text{start n}}^{\text{end n}} = \frac{1}{N-1}\sum_{i=\text{start n}}^{\text{end n}} \left[d\left(\mathbf{x}[i], \bar{\mathbf{x}}_{\text{start n}}^{\text{end n}}\right)\right]^2 \tag{5.6.8}$$

In the above equations, N is the number of frames in the region ($N = \text{end n} - \text{start n} + 1$), and $d(\mathbf{x}, \mathbf{y})$ is the Euclidian distance between the vectors **x** and **y**.

For split-and-merge segmentation, we have the freedom to vary two threshold parameters: θ_{split} and θ_{merge} Any region with variability greater than θ_{split} will be split during the split phase of the processing, and any two regions whose mean vectors differ (Euclidian distance) by less than θ_{merge} will be merged during the merge phase of the processing.

Split-and-merge segmentation works as follows:

1. **Initialize**: Assign the entire image to a single starting "region". Calculate the mean vector $\bar{\mathbf{x}}$ and variability σ^2 corresponding to the base region.

2. **Split phase**: Examine all regions in the image. If the variability of a region is greater than θ_{split}, bisect the region into two distinct regions. Continue splitting until every region has a variability less than θ_{split}.

3. **Merge phase**: Examine all regions in the image pairwise from left to right. If the Euclidian distance between a pair of mean vectors is less than θ_{merge}, merge the regions. Continue merging until the difference between all neighboring mean vectors is greater than θ_{merge}.

4. **Iterate**: Repeat the split and merge phases until the final iteration makes no further changes to the hypothesized region list.

We performed split-and-merge segmentation on the TID and MR testing data sets. The results are given in the following ROC plots (Figure 5.10). We experimented with different threshold values for θ_{split} and θ_{merge}. The ROCs below are the best detection accuracies we achieved, which are a result of holding θ_{merge} constant at 1.125 and varying θ_{split} from 10–40. As in previous sections, the decoder-based segmentation accuracy is shown for reference.

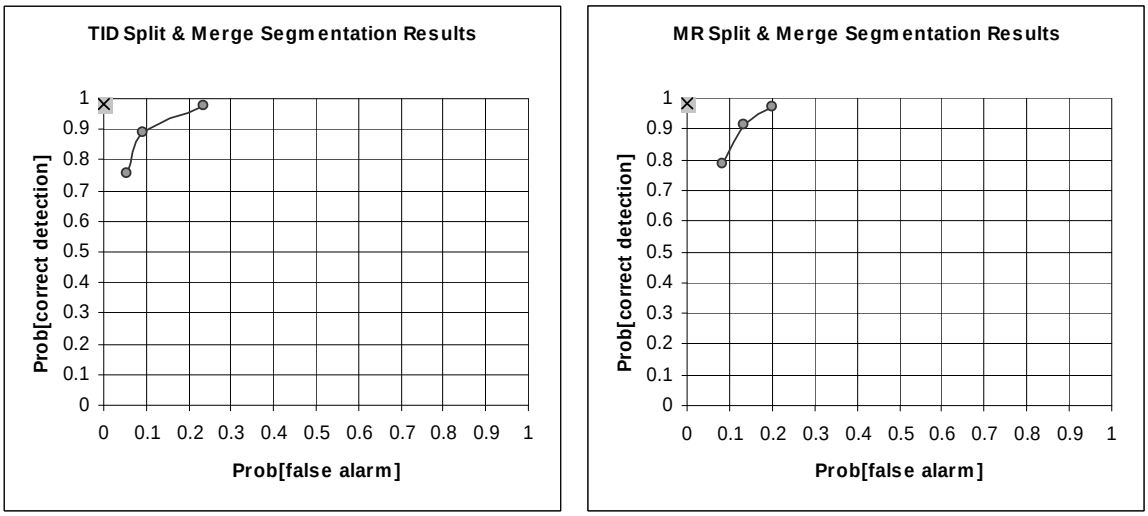

Figure 5.10 ROC results for split-and-merge segmentation on TID (left) and MR (right). Decoder-based segmentation is shown as an "X" for reference.

As with edge detection, split-and-merge segmentation is an effective speech segmentation method, but it is also inferior to decoder-based segmentation. We overlay the ROCs for edge detection and split-and-merge below in Figures 5.11 and 5.12. For both TID and MR data, split-and-merge performs slightly better than our best edge detection method (dendrogram-based distortion).

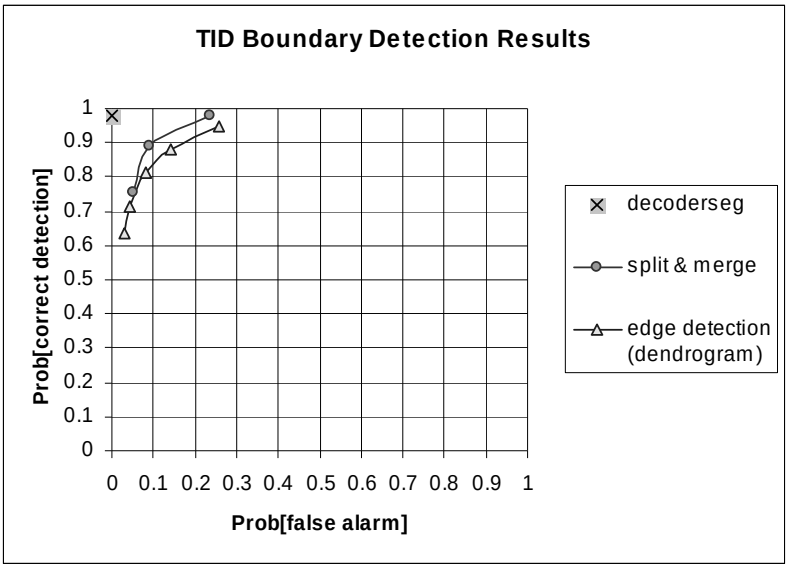

Figure 5. 11 Summary ROC results for split-and-merge segmentation and edge detection segmentation on the TID corpus. Decoder-based segmentation is shown as an "X" for reference.

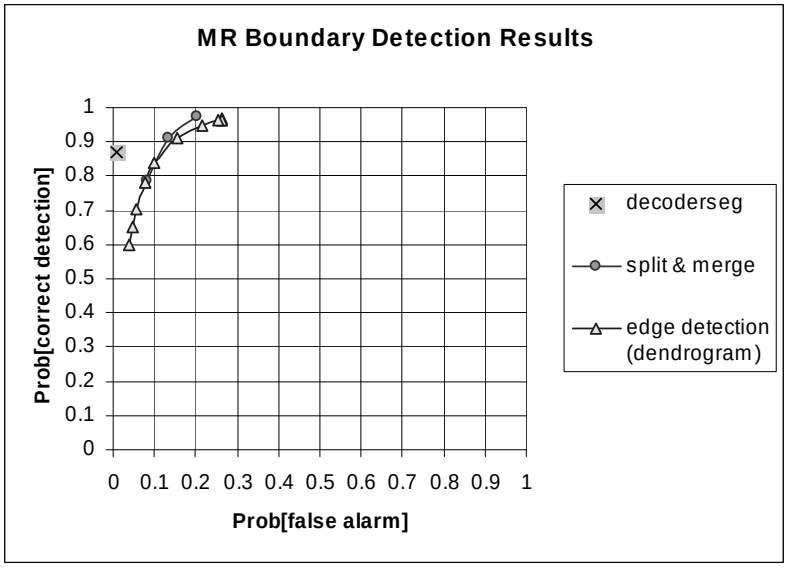

Figure 5.12 Summary ROC results for split-and-merge segmentation and edge detection segmentation on the MR corpus. Decoder-based segmentation is shown as an "X" for reference.

From the ROC graphs, it is clear that although split-and-merge segmentation shows a slight improvement over edge detection segmentation, it is still not able to compete with the original decoder-based segmentation approaches. Table 5.7 quantifies the results in terms of sensitivity index:

Sensitivity Index Values (*d′*)	**TID**	**MR**
decoder-based segmentation	5.0	3.5
phonetic decoder-based segmentation	3.8	~3.0
split & merge segmentation	2.6	2.4
dendrogram-based edge detection	2.3	2.3

Table 5.7 Summary sensitivity index values for split-and-merge segmentation and edge detection segmentation on the TID and MR corpora. Decoder-based segmentation results are shown for reference.

Split-and merge segmentation has a *d′* value of 2.6 for TID data and 2.4 for MR data. While the classifiers are good, they are far from the decoder-based segmentation *d′* of 5.0 for TID data and 3.5 for MR data.

5.7 Analysis: The "Decoder-based Segmentation Dilemma"

The sequence of automatic segmentation experiments presented in this chapter led us to the conclusion that the simplest method of decoding the speech and Viterbi-aligning the hypothesized transcript to the speech data produces segmentations with a high level of quality that our other methods are unable to achieve. However, making use of decoder-based segmentation together with duration normalization is problematic: What typically happens is that words found in the speech signal that are highly spontaneous or under articulated are misrecognized by the baseline recognition system. The incorrect word is then aligned to the speech signal to produce a corresponding errorful boundary hypothesis. When these incorrect boundaries are applied to normalize the speech signal so that every phone has the same duration, the mistake is then *reinforced* by our approach. The duration normalization recognition models then tend to make the same mistake that the baseline decoder made in the first place. We term this phenomenon the "decoder-based segmentation dilemma".

The underlying problem is as follows: good segmentation requires the use of a good statistical model for the fundamental speech units we are looking for in the speech. State-of-the-art HMM acoustic models provide a compact and effective way to model the spectral characteristics of the fundamental speech units together with the time series nature of observed speech features. When the HMM acoustic models are constrained by language models containing information concerning the likelihood of the sequence of words in a given language, adequate speech recognition and segmentation is possible. When these constraints are removed and the HMMs are used to recognize sequences of phonemes rather than

sequences of words, the resulting recognition and segmentation quality is substantially degraded. Segmentation techniques like edge-detection and split-and-merge segmentation which do not make use of an underlying statistical model of speech are even less accurate.

We also investigated combining the different segmentation techniques presented in this chapter to build upon the accuracy of decoder-based segmentation. This was not successful for several reasons: First, in many of the cases when boundaries are missed by decoder-based segmentation, there is little or no evidence in the speech signal that would indicate that a boundary should be hypothesized at that location. Attempts to use distortion metrics or other methods to recover such lost boundaries are too prone to errors and are therefore ineffective. Also, automatic, minimum-error techniques to combine the classifiers place the largest weight on the most accurate classifiers. Because the decoder-based segmentation is far superior to the other classifiers, the supporting information provided by the other classifiers is virtually ignored when combined decisions were made.

Our conclusion is that the "decoder-based segmentation dilemma" will not be overcome by further work to improve the segmentation quality. In the following chapters, we investigate reformulation of the duration normalization algorithm in order to cope with imperfect segmentation information that will be present in real world recognition tasks.

5.8 Conclusions

In this chapter, we presented a variety of techniques for estimating the segmentation of the speech waveform into its constituent sound units. We found that the decoder-based segmentation approach to be by far the best approach for automatically segmenting speech into sound units. Decoder-based segmentation outperforms traditional signal processing-based approaches to detect edges or coherent regions in the speech spectrogram.

While decoder-based segmentation is the best approach from a boundary detection perspective, it is problematic when used in conjunction with duration normalization to improve overall recognition accuracy on spontaneous speech. The recognition errors made by the baseline system are reinforced by the use of the corresponding boundaries, and the duration normalization system is often led to repeat the same errors.

Segmentation approaches that do not rely on a recognition system are attractive because they avoid the problem of reinforced recognition errors; however, the quality of their segmentation information is inadequate for use with the duration normalization system.

We conclude that further improvements of segmentation quality would be very difficult to achieve. In the next chapter, we investigate the possibility of modifying the duration normalization algorithm so it can better cope with segmentation errors.

6: The Modified Duration Normalization Algorithm

In the previous chapter, we showed that despite the high quality of automatic segmentation techniques, the basic duration normalization process does not yield significant improvements in recognition accuracy. This chapter begins with an examination of the effect of phone segmentation errors on the duration normalization process. We then detail simple variants of duration normalization which are designed to help cope with boundary insertions and deletions in automatically-derived phone segmentations. Finally, we close with experiments that show meaningful improvements in speech recognition accuracy via duration normalization and automatically-derived phone boundaries.

6.1 Motivation: Impact of Segmentation Errors

While duration normalization has the potential for large improvements in recognition accuracy, the problem of blindly estimating accurate phone segmentations has continued to thwart our efforts to achieve real recognition improvements via duration normalization. Major problems occur when phone boundaries are inserted or deleted.

Figure 6.1 illustrates an example abstracted from our test speech data. In this example, there are two phone boundaries relatively close together in an utterance. As is often the case in spontaneous speech, there is little evidence for these boundaries in the data (probably due to phone elision), and the automatic segmentation algorithm misses these boundaries entirely. If both boundaries had been detected, the "short" darkened segment between the boundaries would have been expanded by the duration normalization algorithm, as illustrated in the lower left of Figure 6.1. Because the boundaries are not detected, the little evidence for the "short" phone present in the original speech is almost completely discarded when the length of the improperly-detected "long" segment is reduced for duration normalization. This type of boundary detection error leads to a word deletion or substitution error in the final recognition hypothesis. Similarly, when the boundary detection algorithm makes boundary insertion errors, the resulting recognition hypothesis often contains a word insertion or substitution error.

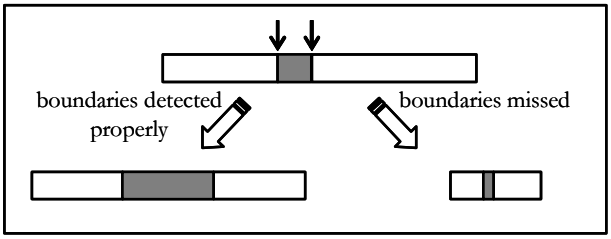

Figure 6.1 Illustration of resulting normalized segments when boundary detection is in issue.

Figure 6.2 shows mel-frequency log spectrograms for an utterance from the TID Spanish database. In this example, the boundaries for the short /z/ and /i/ phones in the middle of the word "sete**ci**entas" (/s e t e **z i** e n t a s/) are missed by our best automatic segmentation technique. The segmentation shown above the spectrogram in the uppermost panel is the correct segmentation. The segmentation shown below the spectrogram in the uppermost panel is derived automatically using decoder-based segmentation, and it contains boundary deletions. When the speech is normalized using the oracle boundaries, we can see an expansion of the short /z/ phone which makes it more "visible" to the recognizer. When the speech is reconstructed and recognized using the automatically-derived, errorful phone boundaries, the evidence for the short /z/ phone is lost, and the recognizer misrecognizes "setecientas" as "setenta" (/s e t e n t a/).

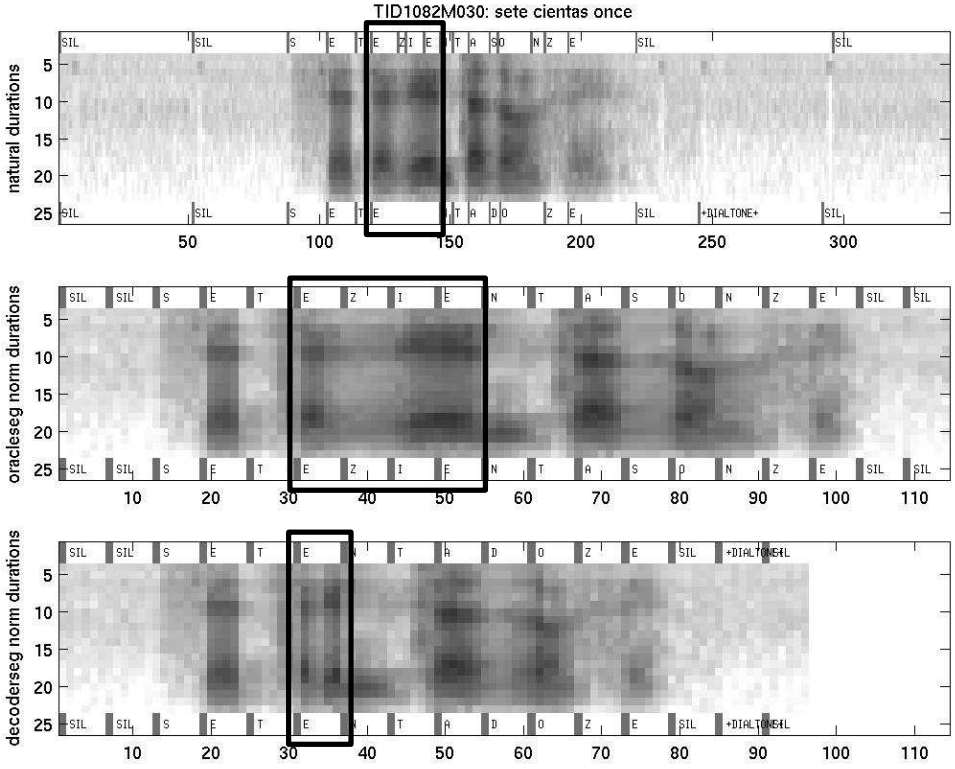

Figure 6.2 Log spectrograms illustrating the result of normalizing with correct and incorrect segmentation information. The segmentation above the spectrogram in the uppermost panel is correct. The segmentation below the same spectrogram contains several deleted boundaries. The result of normalizing using the correct segmentation is shown in the middle panel. The result of normalizing using the errorful segmentation is shown in the bottom panel.

These observations lead us to investigate variants of the duration normalization algorithm that will incur less devastating consequences when boundaries are missed or inserted by automatic boundary detection techniques.

6.2 Partial Contraction Duration Normalization

As illustrated in the previous section, hypothesized phone segmentations with deleted boundaries gravely impact the recognition system by throwing away useful information when incorrectly labeled long segments are contracted. We therefore investigated a *partial* contraction of the long segments to help ensure that useful information is not discarded. The partial contraction is controlled by a reduction parameter r which can be any real number between 0 and 1.

Partial contraction is performed as follows: Let l_0 be the original length of a given long segment. Let L_{norm} be the desired normalized duration prescribed for each segment. Since the segment under consideration is a long segment, we know that $l_0 > L_{\text{norm}}$. Define l_{diff} to be the difference between the original length of the segment l_0 and the desired normalized duration L_{norm} .

$$l_{\text{diff}} = l_0 - L_{\text{norm}} \tag{6.2.1}$$

The normalized length of the long segment is given by the following equation:

$$l' = L_{\text{norm}} + r \cdot l_{\text{diff}} \tag{6.2.2}$$

If the reduction parameter r is set to 0, we have complete contraction as performed in standard duration normalization. If r is set to 1, the long segments are not contracted and are remain at their original durations. For values of r between 0 and 1, a partial contraction is performed as a percentage of the difference between the original length of the hypothesized segment and the desired normalized length of the segment. The partial contraction process is illustrated in the following Figure 6.3.

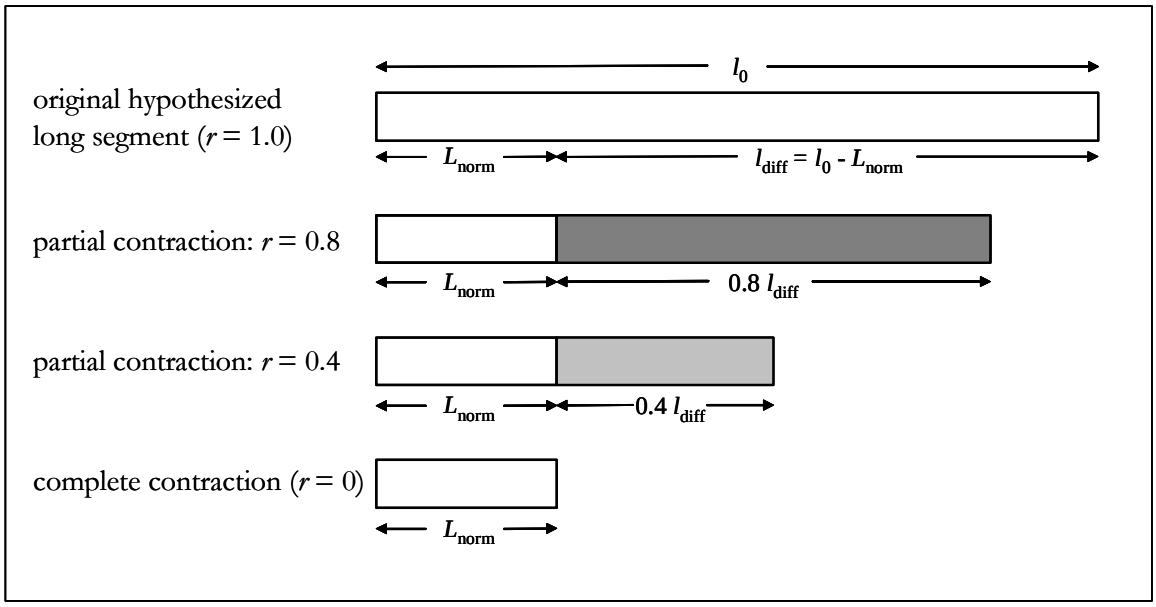

Figure 6.3 Illustration of partial contraction duration normalization using different values of the reduction parameter *r*.

Figure 6.4 shows the same TID utterance from Figure 6.2 normalized using partial contraction duration normalization and a variety of reduction parameter (*r*) values.

Note that in partial contraction duration normalization, expansion operation is not changed. Typically, we are using normalized durations of 6 or 8 frames. Because the HMM acoustic model for each phone contains 3 states, the hypothesized short segments will range in duration from 3–7 frames. Since the resulting normalized duration must be specified as an integer number of frames, we hypothesize that interpolating between the original length of a short segment and a normalized duration of 8 frames would not make a significant impact on the accuracy of the system.

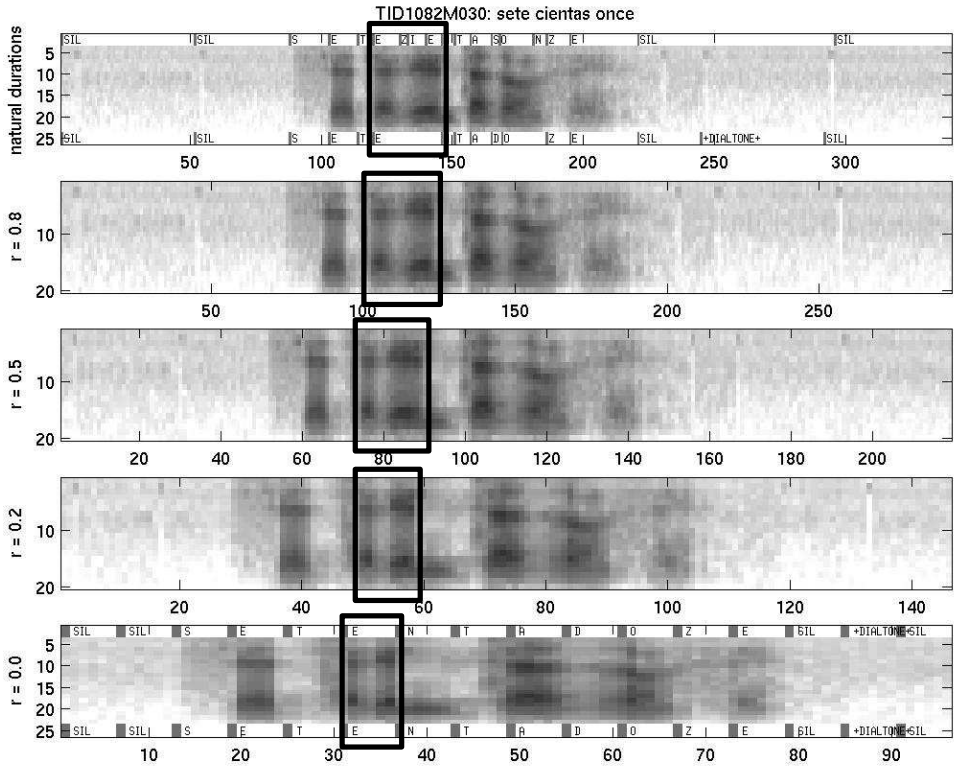

Figure 6.4 Log spectrograms illustrating the result of partial contraction duration normalization using a variety of reduction parameters. Note that the time scale is not the same from panel to panel.

6.3 Partial Contraction Duration Normalization: Experimental Results

We performed partial contraction duration normalization experiments on the Telefónica (TID) and Multiple Register (MR) corpuses. In both cases, we segmented the databases blindly using the decoder-based approach described in Section 5.1. We varied the reduction parameter *r* between 0 and 1 and normalized the training and testing data sets using partial contraction duration normalization. We then trained models to correspond to each *r* value and tested under matched conditions.

The results for TID are given in Figure 6.5.

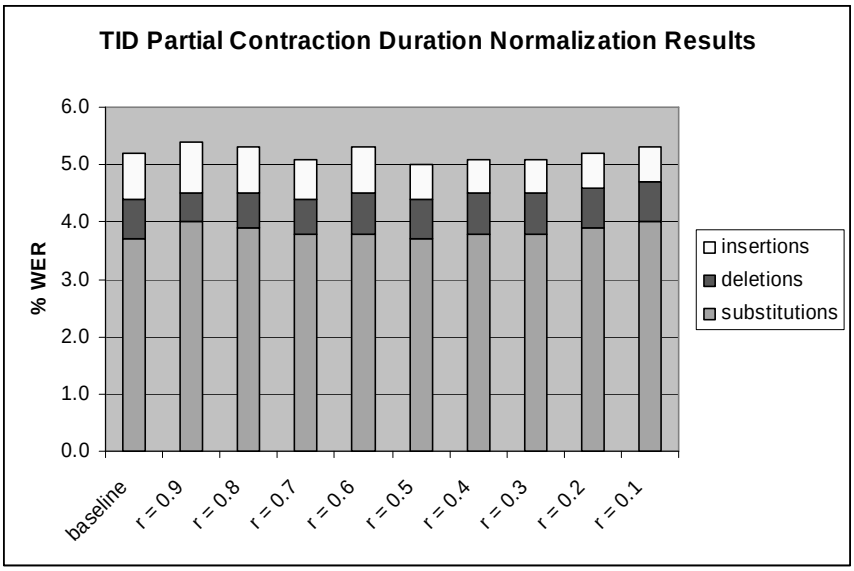

Figure 6.5 Recognition results using partial contraction duration normalization on the TID corpus. Results are presented as a function of the reduction parameter *r*.

On the TID data, partial contraction achieves a 3.8% relative improvement over baseline accuracy. The WER is 5.0% when the reduction parameter *r* is 0.5. We observe accuracy slightly better than baseline for *r* values of 0.3, 0.4, 0.5, and 0.7.

Partial contraction duration normalization on the MR corpus yields the following results (Figure 6.6):

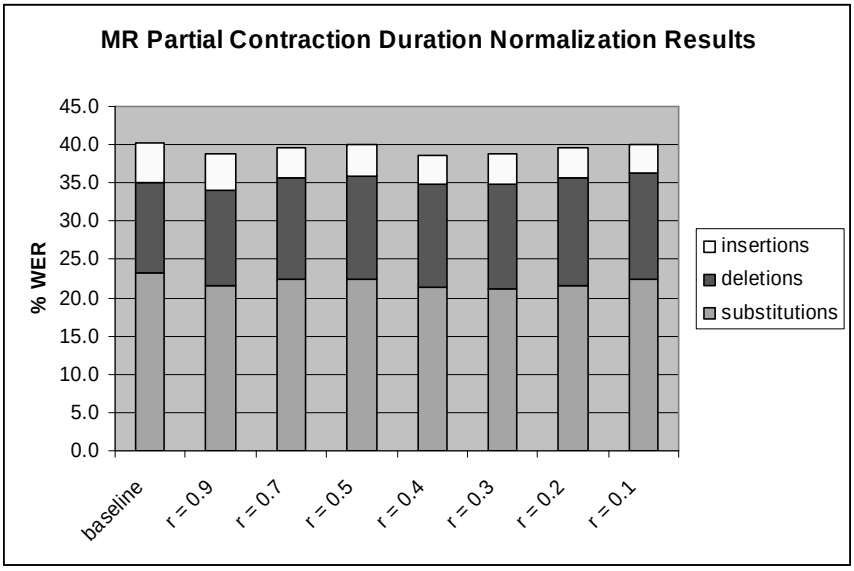

Figure 6.6 Recognition results using partial contraction duration normalization on the MR corpus. Results are presented as a function of the reduction parameter *r*.

Partial contraction duration normalization performs slightly better than baseline for all values of *r* tested. The best accuracy is seen when *r* = 0.4: the WER of 38.5% reflects a 4.5% relative improvement over baseline accuracy on MR data.

6.4 Variants of Duration Normalization: Standard, Expand-Only, Contract-Only

For a given phone segment, the duration normalization algorithm will do one of two things: If the phone is longer than the desired duration, the sequence of log spectral vectors corresponding to the phone are downsampled in time to achieve the normalized duration. If the phone is shorter than the desired duration, the log spectral vectors are expanded in time, and the "missing" vectors are replaced by missing feature methods.

As described in the Section 6.1, boundary detection errors often lead to recognition errors, especially in cases when short phones are not detected. To alleviate this problem, we experimented with the following variants of duration normalization:

- **standard**: expand short phones, contract long phones
- **expand-only**: expand short phones, leave long phones at their natural durations
- **contract-only**: contract long phones, leave short phones at their natural durations

Figure 6.7 illustrates the results of normalizing a given segment of speech using each of the three variants of duration normalization:

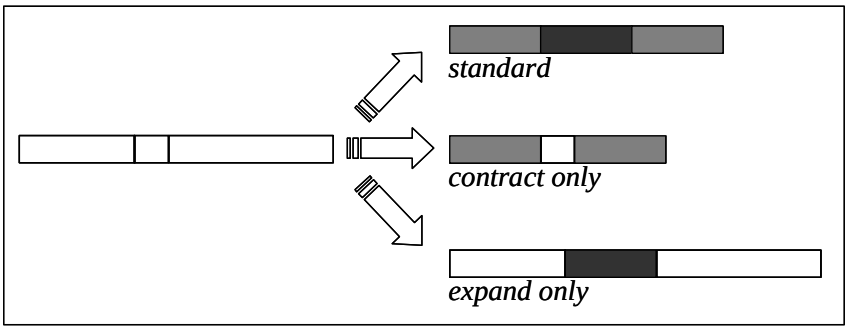

Figure 6.7 Illustration of the different variants of duration normalization: standard, contract-only, and expand-only.

The expand-only variant helps to compensate for examples like the illustration shown in Figure 6.1 and the example shown in Figure 6.2. If the boundaries of a "short" phone were missed, the surrounding segment would be incorrectly considered a long phone and contracted by the standard duration normalization approach. In expand-only duration normalization, the incorrect long phone would *not* be contracted in time, giving us a better chance to properly recognize the missed short phone during decoding. Similarly, contract-only duration normalization helps to compensate for spurious boundaries inserted by automatic boundary estimation algorithms.

Each variant of duration normalization gives rise to a different set of acoustic models during training and a different recognition hypothesis during decoding. Decoding with expand-only duration normalization should produce fewer word deletion errors but more word insertion errors. Conversely, decoding with contract-only duration normalization should result in more word deletion errors and fewer word insertion errors. These systematic variations should make the hypotheses good candidates for merging via the parallel hypothesis combination method reported by Singh in (Singh *et al.*, 2001).

In Singh's method, the hypotheses are combined into a graph with nodes representing each word. Crossovers are introduced between the hypotheses at time instants when both hypotheses have a transition from one word to the next. (Note that if the same word is seen in both hypotheses at the same time, the two words are merged into a single node in the graph.) The graph is then searched for the best scoring hypothesis with respect to the language model. For more details on hypothesis combination, see the complete description in Section 2.8.

6.5 Experiments Using Automatically-Derived Phone Boundaries and Hypothesis Combination

We started by training baseline models on each of the training sets using standard Baum-Welch training. In our "oracle" experiments, we used decoder-based segmentation and the reference transcripts to derive "oracle" phone boundaries. In our "blind" experiments, we used decoder-based segmentation to derive the locations of our estimated phone boundaries.

Using these phone boundaries, we then normalized our training and testing sets using each of the three variants of duration normalization (standard, expand-only, contract-only). For each corpus, we trained three separate acoustic models on the training set, one model for each variant of duration normalization.

We then decoded the testing sets using each variant of duration normalization, which produced three recognition hypotheses for a given utterance. Finally, we employed hypothesis combination to select the final recognition hypothesis and scored our results. Table 6.1 reports results for TID data. Table 6.2 contains results for MR data. BN results are reported in Table 6.3.

TID results	WER	Relative Improvement
baseline	5.2%	—
“oracle” experiment	3.2%	38.5%
“blind” experiment	4.8%	7.7%

Table 6.1 Results for duration normalization and hypothesis combination on the TID Spanish connected digits data. This technique achieves a 7.7% relative reduction in WER on TID.

MR results	WER	Relative Improvement
baseline	40.3%	—
“oracle” experiment	31.7%	21.3%
“blind” experiment	37.8%	6.2%

Table 6.2 Duration normalization and hypothesis combination results for the spontaneous register of the MR corpus. A relative reduction in WER of 6.2% is seen on MR data.

BN results	WER	Relative Improvement
baseline	33.4%	—
“oracle” experiment	28.8%	13.8%
“blind” experiment	32.1%	3.9%

Table 6.3 Broadcast News 1999 Eval 1 recognition results with duration normalization and hypothesis combination. A 3.9% relative reduction in WER is achieved on BN data.

Our experimental results show a reduction in WER over baseline for each of the databases tested. Consistent with experiments using various speech compensation algorithms for robust recognition, the accuracy improvement achieved using smaller databases is greater than the accuracy improvement achieved using larger databases such as broadcast news. In large tasks, the extensive amount of training data and detailed modeling framework lead to a system that is presumably more robust. It is interesting to note that the duration normalization and hypothesis combination algorithm yields an accuracy improvement even in the large-scale test.

We note that when using standard duration normalization alone with oracle segmentations, the best possible relative reduction in WER is 34.6% for TID, 20.1% for MR, and 5.4% for BN. Standard duration normalization alone with estimated segmentations does not yield significant improvements over baseline accuracy on any of the databases tested. When duration normalization is combined with hypothesis combination, significant improvements are achieved in all of our tests.

Our results show that duration normalization is a practical technique for improving speech recognition accuracy for HMM-based systems when the recognition hypotheses produced by its variants are combined with hypothesis combination.

6.5.1 Detailed accuracy analysis for variants of duration normalization

Tables 6.4, 6.5, and 6.6 show the breakdown of errors made by each variant of duration normalization using estimated segmentation information on our test corpora. The word recognition errors are broken down into substitution (sub.), deletion (del.), and insertion (ins.) errors. The baseline error breakdown and post-hypothesis combination error breakdowns are also given for reference.

TID WER breakdown	Sub. errors	Del. errors	Ins. errors
Baseline	3.7%	0.7%	0.8%
Standard dur. norm.	4.1%	0.7%	0.6%
Expand-only dur. norm.	3.8%	0.6%	0.8%
Contract-only dur. norm.	4.0%	0.7%	0.5%
Dur.norm. + hyp. comb.	3.6%	0.6%	0.6%

Table 6.4 Types of recognition errors made by each variant of duration normalization with estimated segmentation information on TID data. Word recognition errors are broken down into substitution (sub.), deletion (del.), and insertion (ins.) errors.

MR WER breakdown	Sub. errors	Del. errors	Ins. errors
Baseline	23.2%	11.9%	5.2%
Standard dur. norm.	22.2%	13.7%	3.9%
Expand-only dur. norm.	23.0%	12.8%	4.5%
Contract-only dur. norm.	22.1%	13.9%	3.6%
Dur.norm. + hyp. comb.	20.7%	13.6%	3.5%

Table 6.5 Types of recognition errors made by each variant of duration normalization with estimated segmentation information on MR data. Word recognition errors are broken down into substitution (sub.), deletion (del.), and insertion (ins.) errors.

BN WER breakdown	Sub. errors	Del. errors	Ins. errors
Baseline	22.8%	6.8%	3.8%
Standard dur. norm.	24.5%	7.0%	4.5%
Expand-only dur. norm.	25.2%	6.2%	5.2%
Contract-only dur. norm.	22.9%	7.7%	3.3%
Dur.norm. + hyp. comb.	21.8%	7.0%	3.3%

Table 6.6 Types of recognition errors made by each variant of duration normalization with estimated segmentation information on BN data. Word recognition errors are broken down into substitution (sub.), deletion (del.), and insertion (ins.) errors.

As expected, expand-only duration normalization produces fewer word deletion errors and more word insertion errors than standard and contract-only duration normalization. Also, contract-only duration normalization produces fewer word insertion errors and more word insertion errors than standard and expand-only duration normalization. In all cases tested, hypothesis combination is able to take advantage of these variations to produce recognition hypotheses with a lower word substitution rate than any of the single duration normalization variants alone.

Tables 6.7, 6.8, and 6.9 show a complete result summary for each variant of duration normalization applied to each of our data sets. Again, baseline and post-hypothesis combination results are also given.

TID result summary	**WER**	**Relative Improvement**
Baseline	5.2%	—
Standard dur. norm.	5.4%	-3.8%
Expand-only dur. norm.	5.2%	0%
Contract-only dur. norm.	5.2%	0%
Dur. norm. + hyp. comb.	4.8%	7.7%

Table 6.7 Summary of errors made using duration normalization and estimated segmentation information on the TID corpus. Hypothesis combination of the individual recognition hypotheses produces a 7.7% relative reduction in WER when compared with the baseline.

MR result summary	**WER**	**Relative Improvement**
Baseline	40.3%	—
Standard dur. norm.	39.8%	1.2%
Expand-only dur. norm.	40.3%	0%
Contract-only dur. norm.	39.6%	1.7%
Dur. norm. + hyp. comb.	37.8%	6.2%

Table 6.8 Summary of errors made using duration normalization and estimated segmentation information on the MR corpus. Hypothesis combination of the individual recognition hypotheses produces a 6.2% relative reduction in WER when compared with the baseline.

BN result summary	**WER**	**Relative Improvement**
Baseline	33.4%	—
Standard dur. norm.	36.0%	-7.8%
Expand-only dur. norm.	36.6%	-9.6%
Contract-only dur. norm.	33.9%	-1.5%
Dur. norm. + hyp. comb.	32.1%	3.9%

Table 6.9 Summary of errors made using duration normalization and estimated segmentation information on the BN corpus. Hypothesis combination of the individual recognition hypotheses produces a 3.9% relative reduction in WER when compared with the baseline.

Using blindly-estimated segmentation information, the contract-only variant of duration normalization outperforms the other two variants in all cases tested. On TID data, no improvements are seen using any variant of duration normalization alone, and the standard variant actually causes a degradation in accuracy. On the MR data, slight improvements are made by the standard and contract-only variants of duration normalization alone. On BN data, all of the variants by themselves cause a significant *degradation* in accuracy compared to baseline. In spite of this, hypothesis combination is a successful method to combine these individual hypotheses and choose a good overall hypothesis that consistently performs better than baseline.

6.6 Discussion: Duration Normalization Variants and Hypothesis Combination

When duration normalization is combined with hypothesis combination, there is a greater improvement in recognition accuracy than with duration normalization alone. With oracle segmentations, we see a greater potential for improvement than that of standard duration normalization alone. With estimated segmentations and standard duration normalization, we are unable to achieve actual improvements in recognition accuracy. With estimated segmentations, duration normalization, and hypothesis combination, we achieve significant improvements in recognition accuracy on all databases tested, including a more rigorous experiment on a large vocabulary Broadcast News recognition task.

The important thing to note is that each variant of duration normalization makes different *types* of errors at different times, even though by itself it does may not reduce the overall word error rate. Hypothesis combination of the recognition output produced by the duration normalization variants outperforms the individual hypotheses produced by each variant alone, and it also outperforms the baseline accuracy on each data set tested. As stated earlier, when duration normalization and hypothesis combination are used in conjunction on the TID corpus, a 7.7% relative reduction in WER over baseline is achieved. The MR corpus gives a 6.2% relative reduction in WER and the BN corpus gives a 3.9% relative reduction in WER. Using matched pairs analysis, the BN result is statistically significant with 99% confidence.

6.7 Conclusions

In this chapter we examined the impact of segmentation errors on the effectiveness of duration normalization for improved speech recognition accuracy. We found that boundary insertions and deletions have a strong impact on the effectiveness of our algorithm. In the previous chapter, we observed

that it is extremely difficult to further improve the segmentation quality produced by our automatic segmentation algorithms. We therefore searched for ways to make our approach more robust to boundary detection errors.

We observed that the most damaging boundary recognition errors were multiple consecutive boundary deletions because they lead the duration normalization algorithm to incorrectly discard a large number of frames, throwing away what little evidence there may have been in the signal for certain spontaneous speech sound units. We experimented with partial-contraction duration normalization to help combat this problem by reducing the amount by which long phones are reduced in time. This approach yielded some improvements in accuracy on TID and MR.

We then experimented with multiple variants of duration normalization: expand-only, contract-only, and standard. Together with hypothesis combination, we found significant improvements in recognition accuracy, even on the large scale BN test. The different variants made different types of recognition errors, and hypothesis combination was successful at choosing the correct words from the candidate hypotheses. At the cost of multiple recognition passes for each utterance, significant improvements in recognition accuracy can be achieved through duration normalization and hypothesis combination.

In the next chapter, we explore the idea of using probabilistic segmentation information in normalizing sound unit durations.

7: The Soft Segmentation Duration Normalization Algorithm

The dependence of the duration normalization algorithm on exact segmentation information has made it difficult to achieve large reductions in WER when boundaries are inserted or deleted. Previously, we assumed that we only had access to the final output of our segmentation algorithms, *i.e.* the strict binary classification of every frame of speech into one of two categories: *boundary* or *non-boundary*. In this chapter, we present a "soft" formulation of the duration normalization approach that can make use of the underlying likelihood scores associated with each potential boundary location.

7.1 Using Probabilistic Segmentation to Normalize Phone Durations

Assume that a given utterance of speech is comprised of a series of *N* segments: $S_0 \ldots S_{N-1}$. Let $l_0 \ldots l_{N-1}$ represent the natural length of each segment in number of frames. Also define $\varphi_{i,i+1}$ to be the probability that a boundary is present between segments S_i and S_{i+1}. This is illustrated in Figure 7.1.

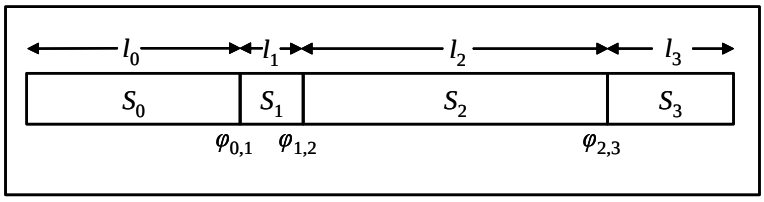

Figure 7.1 Illustration of probability scores assigned to boundaries between segments of different lengths.

Our duration normalization approach assigns a new duration $l'_0 \ldots l'_{N-1}$ to each speech segment. The standard duration normalization algorithm defined in Chapter 4 assumes that the likelihood of each boundary $\varphi_{i,i+1}$ is 1.0 for every *i* and assigns a new duration of L_{norm} to each segment, as shown in Equation 7.1.1.

$$l'_0 = l'_1 = \cdots = l'_{N-1} = L_{\text{norm}} \qquad (7.1.1)$$

In soft segmentation duration-normalization, we assume that $\varphi_{i,i+1}$ can be any real number in the closed interval $[0.0, 1.0]$ for a given boundary location. Given this additional information, we derived a formula to compute for the new duration of each segment in an utterance. To illustrate the approach, we present a simple example containing 1 boundary. We then comment on the general case of N boundaries in an utterance, and we close with a note on the computational complexity of our algorithm.

7.1.1 The Single Boundary Case

The simplest possible case is shown in Figure 7.2. Assume that the speech is composed of two hypothesized segments, S_0 and S_1, with respective segment lengths l_0 and l_1. The probability that a boundary exists between the two segments is $\varphi_{0,1}$. (And the probability that the boundary does not exist is therefore $1-\varphi_{0,1}$.)

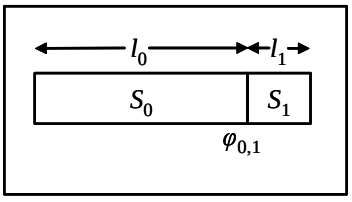

Figure 7.2 Illustration of the single-boundary case.

There are only two possibilities for this example: either the boundary is present or it is not. If the boundary is present, standard duration normalization would assign a new segment length of L_{norm} to each of the two segments. This is illustrated in Figure 7.3 and quantified in Equations 7.1.2 and 7.1.3.

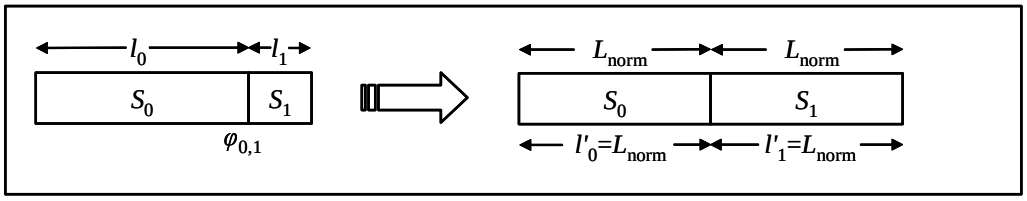

Figure 7.3 Illustration of normalizing the single boundary case when the boundary is assumed to be present.

$$l'_0 \mid \text{boundary} = L_{\text{norm}} \tag{7.1.2}$$

$$l'_1 \mid \text{boundary} = L_{\text{norm}} \tag{7.1.3}$$

If the boundary is not present, standard duration normalization would consider the entire utterance as one segment and assign a new segment length of L_{norm} to the combined segment $S_0 + S_1$. This is illustrated in Figure 7.4.

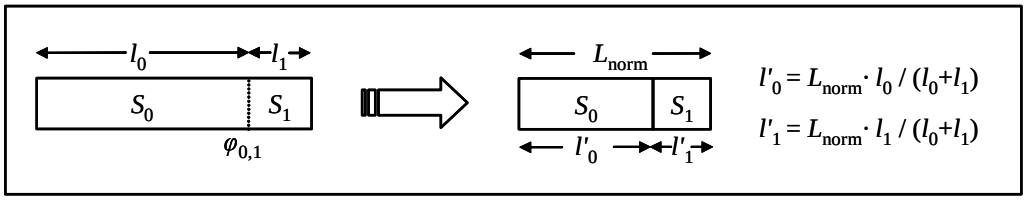

Figure 7.4 Illustration of normalizing the single boundary case when the boundary is assumed to be absent.

In the non-boundary case, the normalized segment lengths l_0' and l_1' should therefore be a fraction of L_{norm} proportional to the original lengths of the two segments. This is quantified in Equations 7.1.4 and 7.1.5.

$$l_0' \mid \text{non} - \text{boundary} = L_{\text{norm}} \cdot \frac{l_0}{l_0 + l_1} \tag{7.1.4}$$

$$l_0' \mid \text{non} - \text{boundary} = L_{\text{norm}} \cdot \frac{l_1}{l_0 + l_1} \tag{7.1.5}$$

Combining Equations 7.1.2 and 7.1.4, and incorporating the boundary probability information, we derive the appropriate value for l_0' as follows:

$$l_0' = \text{P}(\text{boundary}) \cdot l_0' \mid \text{boundary} + \text{P}(\text{non} - \text{boundary}) \cdot l_0' \mid \text{non} - \text{boundary} \tag{7.1.6}$$

$$l_0' = \varphi_{0,1} \cdot L_{\text{norm}} + (1 - \varphi_{0,1}) \cdot L_{\text{norm}} \cdot \frac{l_0}{l_0 + l_1} \tag{7.1.7}$$

$$l_0' = L_{\text{norm}} \left[\varphi_{0,1} + (1 - \varphi_{0,1}) \cdot \frac{l_0}{l_0 + l_1} \right] \tag{7.1.8}$$

A similar combination of Equations 7.1.3 and 7.1.5, followed by simplification, yields the following equation for l_1':

$$l_1' = L_{\text{norm}} \left[\varphi_{0,1} + (1 - \varphi_{0,1}) \cdot \frac{l_1}{l_0 + l_1} \right] \tag{7.1.9}$$

Equation 7.1.7 shows clearly that the normalized length assigned to S_0 is an interpolation between the boundary and non-boundary lengths controlled by the probability that the boundary exists between the given segments.

7.1.2 The General Case

In general, the speech utterance is composed of *N* hypothesized segments, $S_0 \ldots S_{N-1}$, with respective segment lengths $l_0 \ldots l_{N-1}$, where *N* is less than or equal to the number of frames in the utterance. We also have *N*–1 probabilities $\varphi_{0,1} \ldots \varphi_{N-2,N-1}$ corresponding to each hypothesized boundary location.

These *N* hypothesized segments give rise to 2^{N-1} possible boundary configurations that must be considered. For each configuration, we compute P(C), the probability that the configuration occurs, as a product of the probability that each boundary is present ($\varphi_{j,j+1}$) or absent ($1-\varphi_{j,j+1}$), depending on the specific configuration under consideration. We then compute $l'_i \mid \mathrm{C}$, the individual normalized segment lengths given the current boundary configuration. In a particular boundary configuration C, the segment S_i will be part of a combined segment containing itself and zero or more neighbor segments (depending on the assumed boundary configuration). The value of $l'_i \mid \mathrm{C}$ is a fraction of the global normalized duration (L_{norm}) that is proportional to the original segment lengths of the segments that make up the corresponding combined segment.

The final value for the normalized duration l'_i of each segment is computed with the following sum over all possible boundary configurations (Eq. 7.1.10). Each summand is the probability that a configuration occurs times the individual normalized segment length given the assumed boundary configuration:

$$l'_i = \sum_{\text{all possible C}} \mathrm{P(C)} \cdot l'_i \mid \mathrm{C} \qquad (7.1.10)$$

7.1.3 Computational Complexity

While the presented formulation of probabilistic or soft segmentation-based duration normalization is theoretically sound, there are some practical considerations which affect the way it must be implemented in practice. We first note that for a given number *N* of hypothesized segments, the algorithm considers all possible boundary configurations when computing the new length of each segment. The algorithm therefore has a running time $O(2^N)$.

Ideally, we would like to be able to estimate a continuous likelihood function describing the probability that each given frame is a boundary and then use those estimated probabilities together with the softseg formulation to determine the proper duration normalization warping. Typical utterances presented to the recognizer have a length of 500–1000 frames, which makes consideration of all possible boundary

configurations computationally intractable. In practice, we have found it is possible to compute normalized durations for utterances with 30 or fewer hypothesized segment boundaries.

In Section 7.3, we present an experiment where, for practical reasons, decoder-based segmentation is first used to locate "anchor" segments within the speech signal. The softseg approaches are then used to warp the speech in each anchor segment using a manageable number of estimated boundary probabilities occurring within each anchor segment.

7.2 Simulation Using Oracle Segmentation Degraded by Decoder Segmentation

Given the proposed soft segmentation duration normalization algorithm, we performed the following experiment on the Telefónica (TID) database as a "proof of concept" to investigate and evaluate the effectiveness and soundness of the softseg algorithm.

We start with the TID database together with the oracle segmentation information derived from Viterbi-alignment of the reference transcripts to the speech using baseline HMM acoustic models. We also generate decoder-based segmentation information by recognizing the speech using the baseline TID recognition system and Viterbi-aligning the recognition hypotheses to the speech.

Using the decoder-based segmentation, we purposefully degrade the oracle segmentation information in the following controlled manner:

1. **Correct Boundaries**: For each boundary in the oracle segmentation that is correctly located by the decoder-based segmentation, we assign a probability of 1.0 to the boundary location.

2. **Deleted Boundaries**: For each boundary in the oracle segmentation that is not located by the decoder-based segmentation, we assign a probability of φ_{del} to the boundary location.

3. **Inserted Boundaries**: For each boundary hypothesized in the decoder-based segmentation with no corresponding boundary in the oracle segmentation, we assign a probability of φ_{ins} to the boundary location.

We chose values for φ_{del} and φ_{ins} from the following set of possible values: {0.0, 0.3, 0.5, 0.7, 1.0}. When φ_{ins} is 0.0, there is no penalty incurred for inserted boundaries, and when φ_{ins} is 1.0, incorrectly inserted boundaries have the same weight as correct boundaries in the segmentation. Conversely, when φ_{del} is 1.0, there is no penalty for deleting a boundary, and when φ_{del} is 0.0, deleted boundaries are considered completely absent by the soft segmentation duration normalization algorithm. If our soft

segmentation formulation is correct, we expect to observe accuracy similar to duration normalization using oracle segmentation information when φ_{del} is set to 1.0 and φ_{ins} is set to 0.0. When φ_{del} is set to 0.0 and φ_{ins} is set to 1.0, we expect to observe accuracy similar to duration normalization using decoder-based segmentation information. As the weight given to boundary errors is varied between these lower and upper bounds, we expect to see recognition word error rates varying between the bound accuracy values.

For TID, baseline recognition accuracy is a WER of 5.2%. Duration normalization using oracle segmentations yields a WER of 3.4%, and duration normalization using decoder-based segmentation has a WER of 5.4%. In our simulation, we trained a separate recognition model for each possible combination of φ_{del} and φ_{ins} values and tested under matched conditions. The resulting WER values are shown in the Figure 7.5 with a corresponding table in Table 7.1.

The simulation shows that the soft segmentation duration normalization is a sound formulation of the duration normalization problem. The system behaves as expected, with WERs ranging between the oracle segmentation performance (3.4%) and the decoder-based segmentation performance (5.4%) on the TID database. The less severe the probabilities assigned to errorful boundary locations, the better the resulting recognition accuracies.

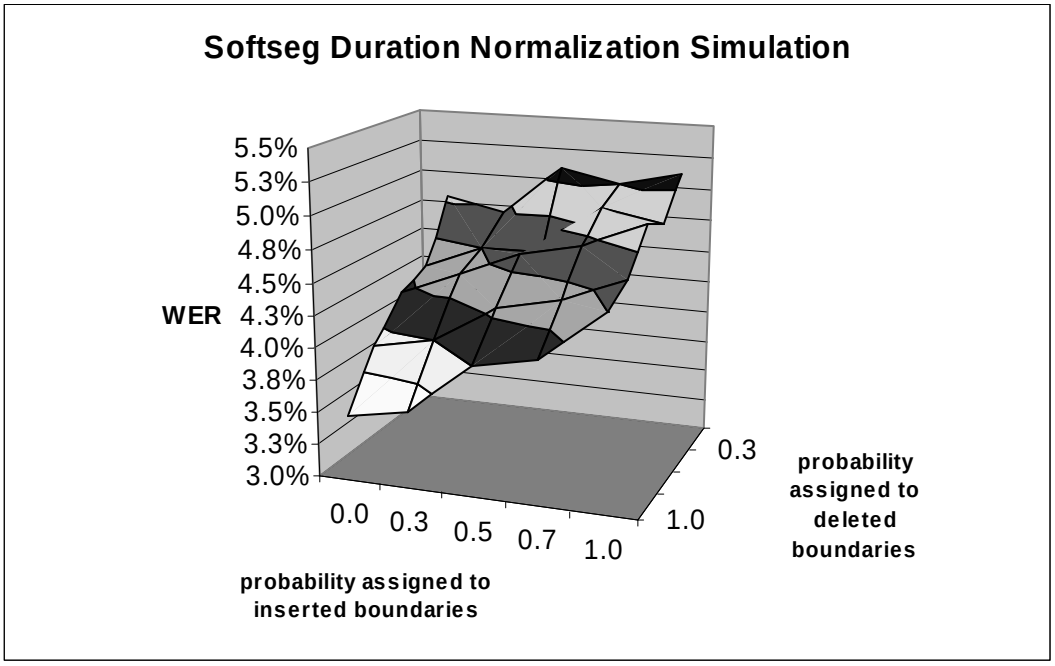

Figure 7.5 WER surface as a function of the probabilities assigned to inserted and deleted boundaries in the decoder-based segmentation of the TID corpus.

				φ_{ins}		
		0.0	0.3	0.5	0.7	1.0
	0.0	4.8%	4.7%	5.1%	5.0%	5.1%
	0.3	4.3%	4.5%	4.5%	4.9%	4.8%
φ_{del}	0.5	4.2%	4.4%	4.6%	4.7%	4.9%
	0.7	3.9%	4.0%	4.3%	4.4%	4.6%
	1.0	3.5%	3.6%	4.0%	4.1%	4.5%

Table 7.1 WER scores as a function of probabilities assigned to the inserted and deleted boundaries in the decoder-based segmentation of the TID corpus.

7.3 Experiment Using Decoder and Edge Detection Segmentations

The formal soft segmentation duration normalization algorithm has a running time of $O(2^N)$, where N is the number of hypothesized boundaries in a given speech utterance. While ideally we would like to estimate and use the probability that a boundary exists at each frame in the speech signal, this is not computationally feasible. As a practical alternative, we performed the following experiment on the TID data set.

Using baseline models, we perform decoder-based segmentation on the TID corpus. This is done to locate "anchor" segments within the speech signal. We also estimate $\varphi[n]$, *i.e.* the probability of a boundary occurring at each frame in the speech signal, using a simple running Euclidian distortion metric on the log spectral vectors. Using the training set of the TID corpus, we divide the data into two classes: "boundary" and "non-boundary". We then estimate the mean and variance of the distortion metric for each class. Assuming that the classes are normally distributed, we then estimate $\varphi[n]$ using these simple distributions.

We then apply the soft segmentation duration normalization algorithm to warp each anchor segment to the proper normalized duration as follows: We first find all peaks in $\varphi[n]$ that occur within the given segment. The peak locations are the estimated boundary locations and probability values input to the soft segmentation duration normalization algorithm. (Note that if more than 20 peaks are detected within a given segment, we take only the top 20 peak locations to ensure that the number of segments being normalized is computationally feasible.) In this manner, we normalize all of the anchor segments in the utterance.

In our experiment, we apply the combined decoder-based segmentation and soft segmentation duration normalization to the training and testing sets of the TID corpus. We train models using standard Baum-Welch training and test under matched normalization conditions. The results are shown in Table 7.2, along with baseline and standard duration normalization results for comparison.

TID Softseg Results	WER	Sub. errors	Del. errors	Ins. errors
Baseline	5.2%	3.7%	0.7%	0.8%
Standard dur. norm.	5.4%	4.1%	0.7%	0.6%
Softseg dur. norm.	5.1%	3.8%	0.7%	0.6%

Table 7.2 Recognition accuracy using duration normalization with decoder-based segmentation and "soft" (probabilistic) segmentation information.

We see that soft segmentation duration normalization performs better than standard duration normalization using the same decoder-based segmentation on the TID database. This accuracy, however, is only slightly better than the baseline recognition accuracy.

Finally, we make use of this softseg hypothesis along with the hypotheses from the other variants of duration normalization (standard, expand-only, contract-only) and use hypothesis combination to see if we can do better than we have done previously. The hypothesis combination results are shown at the bottom of Table 7.3, together with detailed results for each variant of duration normalization tested.

TID Softseg + Hyp. Comb. Results	WER	Sub. errors	Del. errors	Ins. errors
Baseline	5.2%	3.7%	0.7%	0.8%
1. Standard dur. norm.	5.4%	4.1%	0.7%	0.6%
2. Expand-only dur. norm.	5.2%	3.8%	0.6%	0.8%
3. Contract-only dur. norm.	5.2%	4.0%	0.7%	0.5%
4. Softseg dur. norm.	5.1%	3.8%	0.7%	0.6%
Dur.norm. + hyp. comb. (1,2,3)	4.8%	3.6%	0.6%	0.6%
Dur. norm. + hyp. comb (1,2,4)	4.7%	3.5%	0.6%	0.6%

Table 7.3 Comparison of recognition accuracy using duration normalization and hypothesis combination. In the last row, recognition accuracy is shown when the soft segmentation decoder transcripts are passed to hypothesis combination instead of the contract-only decoder transcripts.

We find our best accuracy when hypothesis combination is used to combine standard, expand-only, and the soft segmentation duration normalization variants for the TID test set. We achieve a final WER of 4.7%, which reflects a 9.6% relative reduction in WER over baseline.

7.4 Discussion

In general, the soft segmentation formulation presented at the beginning of this chapter is a sound probabilistic formulation of the duration normalization algorithm. Our simulations confirm that as the probabilistic segmentation information approaches the oracle segmentation, the soft segmentation

algorithm produces recognition results that approach the accuracy of standard duration normalization using oracle segmentation information.

It is clear that the soft segmentation approach requires a sound technique for estimating the probability of a boundary location occurs at several locations throughout the speech signal. As shown in Chapter 5, this is an extremely difficult problem. Decoder-based segmentation techniques are reliable, but imperfect. Distortion metrics can be applied to look for evidence of boundary locations that are missed by decoder-based segmentation.

By design, this combined decoder and soft segmentation-based approach presented in Section 7.3 should compensate for missed boundaries in the decoder-based segmentation stream, assuming that there will be some amount of distortion in the speech signal around missed boundary locations. In spontaneous speech, there are many instances where we see little or no distortion at the missed boundary locations due to the rapid, under-articulated nature of conversational speech. This may account for the fact that only slight accuracy improvements are achieved using this approach.

7.5 Conclusions

In this chapter we presented a reformulation of the duration normalization algorithm to make use of confidence scores associated with each boundary location. We showed that the "soft" formulation of the duration normalization algorithm is consistent but computationally expensive, with a running time on the order of two to the number of segments to be normalized. Soft duration normalization is limited by this large computational complexity. The approach also depends on quality of the algorithm used to generate segmentation probability scores. In order for soft duration normalization to be effective, the segmentation algorithm must be able to properly estimate boundary probabilities in areas where the decoder finds little evidence for a boundary in the speech signal. As stated previously, segmentation based on little or no evidence is a difficult problem.

In a practical experiment, we used soft duration normalization to normalize the speech between decoder-derived segment boundaries. This allowed us to make use of boundary probabilities estimated for every frame and apply soft duration normalization in a computationally manageable fashion. We achieved a small improvement over duration normalization + hypothesis combination alone on the TID database. Unfortunately, the small improvement in recognition accuracy is outweighed by the extra computation required to generate these results.

In the final chapter, we present our thoughts on future work and summarize the major findings of this thesis.

8: Summary and Conclusions

In this chapter, we present a summary of the research and the relevant observations that we have drawn from our investigation of duration normalization and the modeling and recognition of spontaneous speech. We continue with some comments on future research directions and unresolved questions. We close the chapter with our final summary and conclusions.

8.1 Major Findings

8.1.1 Duration Variability of Speech Sound Units is a Problem when Modeling Spontaneous Speech

It is known that HMMs do not effectively model the actual phone durations observed in speech data. A large variability in the durations of tokens for a given phone class make it difficult for HMMs to characterize this class adequately. We have shown that the increased variability of phone durations in spontaneous speech is a considerable factor that leads to degraded recognition accuracy of spontaneous speech in HMM-based systems (when compared with recognition of carefully-read speech). Using an identical HMM-based recognition system on a parallel corpus of read and spontaneous speech, we observed baseline recognition word error rates of 15.6% for carefully-read speech and 40.3% for spontaneous speech. Techniques that attempt to bridge the gap in recognition accuracy between read and spontaneous speech are therefore important avenues of research. The duration normalization technique that we developed is one of many possible techniques to improve modeling and recognition of spontaneous speech.

8.1.2 Duration Normalization Can Help Bridge the Gap

Given *a priori* knowledge of phone boundary locations, normalizing the duration of each phone example in the speech database prior to training is an effective method to overcome the duration modeling weakness of the HMM acoustic speech models. Statistical models of speech such as HMMs attempt to derive a general model to best explain the given training data. These models generalize well to test data that are similar to training data, but fail to generalize as the dissimilarities between the two data sets increase. Duration normalization reduces the duration mismatch between training and test data, which means that HMMs trained and tested on speech data with normalized durations are expected to perform better than HMMs trained and tested on speech data with natural durations.

In a controlled experiment using the parallel Multiple Register database, we have shown that the potential for improvement via phone duration normalization is greater for spontaneous conversational speech than it is for carefully read speech. The potential relative reduction in WER for carefully-read speech was 10.3%, while the potential relative reduction in WER for the spontaneous version of the same speech was 20.1%. Again, this is because the dissimilarities between training and test data are expected to be higher in the case of spontaneous speech. Duration normalization reduces one aspect of these dissimilarities.

8.1.3 Phone Segmentation has a Strong Impact on Duration Normalization Results

Phone segmentation is a difficult problem. The more spontaneous speech data are, the more difficult it becomes to segment automatically these speech data into sound units. Spectrographic signatures of spontaneous speech show small transition regions between phones and numerous regions where it is quite unclear where one phone ends and the next begins. Also, we observe many instances where there is little or no evidence in the speech signal for a sound unit that appears in the recognition dictionary of standard pronunciations.

In the oracle case where perfect transcripts are used to derive "correct" boundary locations, the duration-normalized recognition system benefits from the placement of boundaries and subsequent expansion and reconstruction despite the lack of acoustic evidence for a given phone. We observed potential relative reductions in WER in the range of 5.4%–34.6% when correct boundary information is known *a priori*.

Although phone duration normalization has the potential to increase recognition accuracy by large amounts, the approach is limited in practice due to the difficulties in automatically segmenting the speech into phone units. Automatic detection of boundaries for which there is little or no evidence is a difficult problem. Our duration normalization approach is adversely affected by boundary detection errors, especially when multiple consecutive boundaries are missed. However, one of the key contributions of this thesis is the development of methods that work reasonably despite this factor.

8.1.4 Compensation Techniques Can Cope with "Imperfect" Segmentation

Methods to compensate for boundary detection errors have seen limited success when compared with the large potential recognition improvements observed in "oracle" experiments. We have observed that partial-contraction and soft-normalization techniques are effective in reducing the impact of multiple consecutive boundary detection errors. Partial-contraction achieves relative reductions in WER of 3.8%–4.5%. Soft-normalization reduces the WER on the TID corpus by 0.1% absolute.

The most effective compensation techniques we have developed result from the recognition of multiple "views" of each utterance using different time-normalization schemes (expand-only, contract-only, expand and contract). These multiple perspectives into the speech signal result from the expansion and/or contraction of different regions of the speech signal identified by automatic segmentation techniques. Hypothesis combination techniques are successful in combining the recognition hypotheses from the individual recognition systems based on different time-normalization schemes. We observe relative reductions in WER of 3.9% on BN, 6.2% on MR, and 7.7% on TID data using this technique.

The duration normalization + hypothesis combination approach achieves recognition improvements on 3 separate speech databases, including tests in two languages and tests on a large-scale broadcast news

system. Although the achieved improvements are not as substantial as we would like them to be, the results should generalize to a wide variety of speech recognition systems.

8.2 Some Future Directions

8.2.1 Improving Segmentation Quality

Although from our experience, improving the segmentation quality for spontaneous speech has proven to be quite difficult, a few ideas for future research in this area are possible. If the speech segmentation problem is ever "solved" some day, then the duration normalization approach presented in this thesis becomes more valuable for robust and accurate ASR systems.

The use of speech features that can provide accurate views on multiple time scales is one possible venue for research. Wavelet-based features may allow for better detection of very rapid spontaneous speech events for which there is little evidence in the speech signal. Based on our experience, missing the boundaries that delimit short speech events has a grave impact on duration-normalized recognition accuracy. Features that provide a better chance at locating these regions may have a positive impact on overall recognition accuracy. Note that multiple time scale-based features differ from the dendrogram segmentation networks proposed by Glass and Zue who attempt to generate a hierarchical segmentation network based on a single set of fixed time scale features (Glass and Zue, 1988).

Different basic speech units may also allow for better boundary detection and normalization. The use of fundamental units that have longer durations (*e.g.* syllable-based or word-based units) may allow for more accurate boundary detection. In spontaneous speech, much of the evidence for individual phoneme units is blurred into the neighboring units. There is an increased chance that the evidence for longer fundamental units will remain in the signal even when the speech becomes highly spontaneous.

Further investigation of the human speech production process may result in better segmentation and normalization techniques. The process by which canonical "word representations" in the brain are converted to sounds is a complicated one, and there are many factors that contribute to how a particular instance of a phone is rendered by a speaker. The more we understand about speech production, the better we may be able to adequately capture the necessary salient speech events for robust boundary detection and speech recognition.

8.2.2 Improving Robustness of Duration Normalization to Segmentation Errors

Although there is still room for improvement in speech segmentation, even the most accurate segmentation system will still make errors, especially when the level of spontaneity in the speech

increases. We believe that it would be fruitful to pursue methods of normalization that can cope with the fact that segmentations are errorful.

Historically, speech recognition accuracy is enhanced by algorithms that allow for probabilistic decisions rather than hard decisions. We therefore recommend further investigation of techniques such as the soft duration normalization presented in Chapter 7 of this thesis. Our soft algorithm has a running time on the order of 2^N, which greatly limits the practical experiments that we can perform using this technique. It should be possible, however, to develop methods for normalizing the speech signal using segmentation probabilities that are computationally tractable. Once an efficient means for soft normalization has been developed, it should be possible to generate further real improvements in recognition accuracy via duration normalization techniques.

One possible algorithm for computationally-tractable soft segmentation is as follows: Begin by estimating the probability that a boundary exists at each frame in the speech signal. Then using the estimated boundary probability for each frame, toss a "biased coin" to decide if whether a boundary location should be assigned to the frame or not. Finally, normalize using the standard duration normalization procedures and the resulting segmentation. Note that this type of soft segmentation approach can be implemented using a simple random number generator and has a running time on the order of N rather than 2^N. Preliminary experiments have shown that this technique has potential.

8.3 Summary and Conclusions

In this thesis, we present a technique for improving automatic recognition of spontaneous speech by normalizing the duration of the sound units that make up the speech signal. We conclude that normalizing the speech in such a manner makes it more conducive to the HMM acoustic modeling framework upon which most state-of-the-art ASR systems are built. By reducing the duration variability in the speech tokens presented to the recognition system, we help to ensure that the acoustic models are more accurately trained and better equipped to distinguish between speech sound units.

The duration normalization approach depends on accurate segmentation information. Despite the difficulties in automatic segmentation of spontaneous speech, we were able to develop a duration normalization-based system that provides significant recognition improvements on a variety of spontaneous speech databases, including broadcast news. While the accuracy improvements are not as large as we would like them to be, we have presented a process which helps to bridge the gap in recognition accuracy created when ASR systems are presented with natural, conversational speech.

References

Anastasakos, A., Schwartz, R., Suh, H. (1995). "Duration Modeling in Large Vocabulary Speech Recognition", *1995 IEEE International Conference on Acoustics Speech and Signal Processing Conference Proceedings*, pp. 628–631.

Baker, J. (1975). *Stochastic Modeling as a Means of Automatic Speech Recognition*, Ph.D. Dissertation, Carnegie Mellon University.

Baum, L. (1972). "An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes", *Inequalities*, Vol. 3, pp. 1–8.

Campbell, W.N., Isard, S.D. (1991). "Segment Durations in a Syllable Frame", *Journal of Phonetics*, Vol. 19, 1991, pp. 37–47.

Campbell, W.N. (1992). "Syllable-based Segmental Duration" in G. Bailly, C. Benoit, and T. R. Sawallis eds. *Talking Machines: Theories, Models, and Designs*, Elsevier Science Publishers B. V., 1992, pp. 211–224.

Cooke, M., Green, P., Josifovski, L., Vizinho, A. (2001). "Robust automatic speech recognition with missing and unreliable acoustic data", *Speech Communication*, Vol. 34(3), pp. 267–285.

Crystal, T.H., House, A. S. (1988). "Segmental Durations in Connected Speech Signals: Preliminary Results", *Journal of the Acoustical Society of America*, Vol. 83, No. 4, April 1988, pp. 1553–1573.

Davis, S., Mermelstein, P. (1980). "Comparison of parametric representation for monosyllable word recognition in continuously spoken sentences", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28(4), pp. 357–366.

Engen, T. (1971). "Psychophysics: I. Discrimination and Detection" in *Woodworth & Schlosberg's Experimental Psychology*, 3rd Edition, Kling, J.W. and Riggs, L.A. primary contributors, Holt, Rinehard, and Winston, Inc., New York.

Fiscus, J.G. (1997). "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)", *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Workshop Proceedings*, pp. 347–354.

Ferguson, J.D. (1980). "Hidden Markov Analysis: An Introduction" in *Hidden Markov Models for Speech*, Institute for Defense Analyses, Princeton, New Jersey.

Garofolo, J., Fiscus, J., Fisher, W. "Design and Preparation of the 1996 Hub-4 Broadcast News Benchmark Test Corpora", *Proceedings DARPA Speech Recognition Workshop*, pp. 15–21.

Gillick, L., Cox, S.J. (1989). "Some statistical issues in the comparison of speech recognition algorithms", *1989 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, pp. 532–535.

Glass, J.R., Zue, V.W. (1988). "Multi-Level Acoustic Segmentation of Continuous Speech", *1988 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, pp. 215–218.

Graff, D. (1997). "The 1996 Broadcast News Speech and Language-Model Corpus", *Proceedings DARPA Speech Recognition Workshop*, pp. 11–14.

Hermansky, H. (1990). “Perceptual linear predictive (PLP) analysis of speech”, *Journal of the Acoustic Society of America*, Vol. 87, pp. 1738–1752.

Horowitz, S.L., Pavlidis, T. (1974). “Picture Segmentation by a Directed Split-and-Merge Procedure”, *Proceedings 2nd International Joint Conference on Pattern Recognition*, pp. 424–433.

Huang, X., Alleva, F., Hon, H., Hwang, M., Lee, K., Rosenfeld, R. (1993). “The SPHINX-II Speech Recognition System: An Overview”, *Computer Speech and Language*, vol. 2, pp. 137–148.

Huerta, J.M. (2000). “Robust Speech Recognition in GSM Codec Environments”, Ph.D. Thesis, Carnegie Mellon University, April 2000.

Jelinek, F. (1997). “The Viterbi Search” in *Statistical Methods for Speech Recognition*, pp. 79–91, The MIT Press, Massachusetts.

Jones, M., Woodland, P.C., (1993). “Using Relative Duration in Large Vocabulary Speech Recognition”, *1993 Eurospeech Conference Proceedings*, pp. 311–314.

Klatt, D.H. (1973). “Interaction between Two Factors that Influence Vowel Duration”, *Journal of the Acoustical Society of America*, Vol. 54, No. 4, 1973, pp. 1102–1104.

Klatt, D.H. (1976). “Linguistic Uses of Segmental Duration in English: Acoustic and Perceptual Evidence”, *Journal of the Acoustical Society of America*, Vol. 59, No. 5, May 1976, pp. 1208–1221.

Lee, K. (1989). *Automatic Speech Recognition: The Development of the Sphinx System*, Kluwer Academic Publishers, Boston.

Lee, K., Hon, H., Reddy, R. (1990). “An Overview of the SPHINX Speech Recognition System”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38 (1), pp. 35–45.

Levinson, S.E. (1986). “Continuously variable duration hidden Markov models for automatic speech recognition”, *Computer Speech and Language*, Vol. 1(1), pp. 29–45.

Nawab, S.H., Quatieri, T.F. (1988). “Short-Time Fourier Transform” in *Advanced topics in signal processing*, pp. 289–337, Prentice Hall, New Jersey.

Osaka, Y., Makino, S., Sone, T. (1994). “Spoken Word Recognition Using Phoneme Duration Information Estimated from Speaking Rate of Input Speech”, *1994 International Conference on Spoken Language Processing Conference Proceedings*, pp. 191–194.

Pallet, D.S., Fisher W.M., Fiscus, J.G. (1990). “Tools for the Analysis of Benchmark Speech Recognition Tests”, *1990 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 1, pp. 97–100.

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*, 3rd Edition, McGraw Hill, New York.

Pitrelli, J.F. (1990). “Hierarchical Modeling of Phoneme Duration: Application to Speech Recognition”, Ph.D. Thesis, Massachusetts Institute of Technology, May 1990.

Placeway, P., Chen, S., Eskenazi, M., Jain, U., Parikh, V., Raj, B., Ravishankar, M., Rosenfeld, R., Seymore, K., Siegler, M., Stern, R., Thayer, E. (1997). “The 1996 Hub-4 Sphinx-3 System”, *Proceedings DARPA Speech Recognition Workshop*, pp. 85–89.

Port, R.F., Reilly, W.T., Maki, D.P. (1988). “Use of Syllable-Scale Timing to Discriminate Words”, *Journal of the Acoustical Society of America*, Vol. 83, No. 1, January 1988, pp. 265–273.

Rabiner, L.R., Juang, B-H. (1993). *Fundamentals of Speech Recognition*, Prentice-Hall, New Jersey.

Raj, B., Singh, R., Stern, R.M. (1998). "Inference of Missing Spectrographic Features for Robust Speech Recognition", *1998 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*.

Raj, B. (2000). "Reconstruction of Incomplete Spectrograms for Robust Speech Recognition", Ph.D. Thesis, Carnegie Mellon University, April 2000.

Russell, M.J., Moore, R.K. (1985). "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," *1985 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 1, pp. 5–8.

Siegler, M.A., Stern, R.M. (1995). "On the Effects of Speech Rate in Large Vocabulary Speech Recognition Systems", *1995 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 1, pp. 612–615.

Singh, R., Seltzer, M., Raj, B., Stern, R.M. (2001). "Speech in Noisy Environments: Robust Automatic Segmentation, Feature Extraction, and Hypothesis Combination" *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*.

Umeda, N. (1975). "Vowel Duration in American English", *Journal of the Acoustical Society of America*, Vol. 58, No.2, August 1975, pp. 434–445.

Umeda, N. (1977) "Consonant Duration in American English", *Journal of the Acoustical Society of America*, Vol. 61, No. 3, March 1977, pp. 513–546

Viterbi, A.J. (1967). "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Transactions on Information Theory*, Vol. IT-13, pp. 260–269.