PERFORMANCE ANALYSIS OF ADVANCED FRONT ENDS ON THE AURORA LARGE VOCABULARY EVALUATION

By

Naveen Parihar

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

December 2003

PERFORMANCE ANALYSIS OF ADVANCED FRONT ENDS ON THE

AURORA LARGE VOCABULARY EVALUATION

By

Naveen Parihar

Approved:

Joseph Picone Professor of Electrical and Computer Engineering Director of Thesis	Georgios Y. Lazarou Assistant Professor of Electrical and Computer Engineering Committee Member
Nicolas H. Younan Professor of Electrical and Computer Engineering Graduate Program Director	Corlis Johnson Associate Professor of Mathematics and Statistics Minor Graduate Program Director
Jeff Jonkman Assistant Professor of Mathematics and Statistics Minor Professor	
A. Wayne Bennett Dean of the College of Engineering	

Name: Naveen Parihar

Date of Degree: December 13, 2003

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Joseph Picone

Title of Study: PERFORMANCE ANALYSIS OF ADVANCED FRONT ENDS ON THE AURORA LARGE VOCABULARY EVALUATION

Pages in Study: 116

Candidate for Degree of Master of Science

Over the past few years, speech recognition technology performance on tasks ranging from isolated digit recognition to conversational speech has dramatically improved. Performance on limited recognition tasks in noise-free environments is comparable to that achieved by human transcribers. This advancement in automatic speech recognition technology along with an increase in the compute power of mobile devices, standardization of communication protocols, and the explosion in the popularity of the mobile devices, has created an interest in flexible voice interfaces for mobile devices. However, speech recognition performance degrades dramatically in mobile environments which are inherently noisy. In the recent past, a great amount of effort has been spent on the development of front ends based on advanced noise robust approaches.

The primary objective of this thesis was to analyze the performance of two advanced front ends, referred to as the QIO and MFA front ends, on a speech recognition task based on the Wall Street Journal database. Though the advanced front ends are shown to achieve a significant improvement over an industry-standard baseline front end, this

improvement is not operationally significant. Further, we show that the results of this evaluation were not significantly impacted by suboptimal recognition system parameter settings. Without any front end-specific tuning, the MFA front end outperforms the QIO front end by 9.6% relative. With tuning, the relative performance gap increases to 15.8%. Finally, we also show that mismatched microphone and additive noise evaluation conditions resulted in a significant degradation in performance for both front ends.

DEDICATION

Dedicated to my parents and brothers,
for supporting me in all my endeavours, and for being the never ending constant source of inspiration and motivation.

ACKNOWLEDGMENTS

This thesis would never have been possible without the constant support and encouragement from my major advisor, Joe Picone. When I arrived at Mississippi State University as a graduate student, I had no prior experience or background in speech recognition. Not only did Joe gave me an excellent opportunity to learn and grow as a researcher, his constant strive for excellence had a profound impact on my life.

I also owe a tremendous debt of gratitude to the senior members of ISIP. They never grew tired of my endless questions and were always available for suggestions, discussions and advice. I'm thankful to Ram Sundaram for introducing me to the art of running a recognition experiment, ultimate frisbee and early morning coffee. This acknowledgement would not be complete without thanking Jon Hamaker for giving numerous impromptu lectures on speech recognition, answering my questions on the ISIP speech recognition software, and teaching me tricks for playing ultimate frisbee.

I also extend my thanks to my roommates. Both Anand Kumar and Chandrapalsinh calmly listened to my everyday technology blabber and shared my frustrations when my experiments failed or when it was just a bad day.

I am thankful to the European Telecommunications Standards Institute for funding a major portion of the work presented in this thesis. I wish to acknowledge David Pearce of Motorola Labs, Motorola Ltd., United Kingdom, and Guenter Hirsch of Niederrhein

University of Applied Sciences for their invaluable collaborations and direction on some portions of this thesis. I would like to thank the many participants in the Aurora evaluations: the CDMA Technologies Group at Qualcomm, the Speech Group at International Computer Science Institute (ICSI), the Antropic Signal Processing Group at Oregon Health and Science University (OGI), the Human Interface Lab at Motorola Labs, France Telecom R&D, and Alcatel SEL AG (Germany). They provided me with their front end software which was used in the experimentation presented in this work. I am also thankful to the National Institute of Standards and Technology (NIST) for sharing software for significance tests.

Finally, I am thankful to many other wonderful members of ISIP who made my graduate years extremely memorable.

TABLE OF CONTENTS

LIST OF TABLES

TABLE Page

TABLE Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Over the past few years, speech recognition technology performance on tasks ranging from isolated digit recognition [1] to conversational speech [2,3,4] has dramatically improved. Performance on limited recognition tasks in noise-free environments is comparable to that achieved by human transcribers [5]. This advancement in automatic speech recognition (ASR) technology along with an increase in the compute power of mobile devices, standardization of communication protocols, and the explosion in the popularity of the mobile devices, has created an interest in flexible voice interfaces on mobile devices. Because mobile devices have limited space for text input (e.g., no keyboard) and output space (e.g., a cellular telephone display), voice interfaces are ideal.

One class of approaches for this application involves the use of a client/server architecture as shown in Figure 1. A variety of client/server architectures have been explored in recent years [6]. However, to implement complex applications such as speech recognition and spoken information retrieval, there is a need for a pervasive standard. Hence, standards activity has accelerated in recent years. The standardization of a common architecture is extremely critical to ensure compatibility among various hardware and software platforms. The Aurora Distributed Speech Recognition (DSR) group, a working group under the auspices of the European Telecommunications Standard

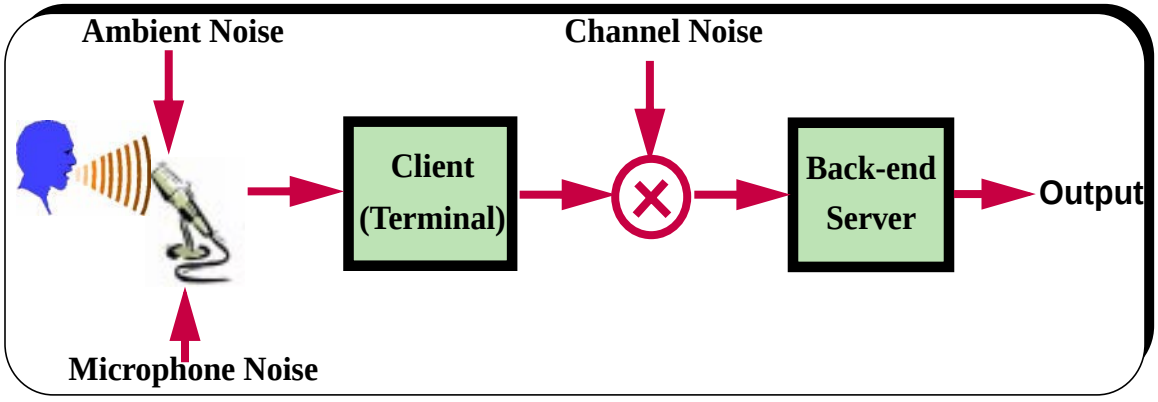

Figure 1. A typical client/server architecture for a mobile computing application. Ambient noise as well as convolutional noise (microphone and channel) are serious problems in this type of application.

Institute (ETSI), has been promoting standards activity for third generation cellular telephony applications [7,8]. Evaluations conducted by the Aurora DSR group were designed to promote standardization of an advanced front end (AFE) for mobile terminal devices as a part of the overall goal of standardization of a DSR architecture.

A speech recognition system can be decomposed into four main components as shown in Figure 2: an acoustic front end, acoustic models, language models, and search. The process of parameterization of a speech signal into a sequence of feature vectors is performed by the acoustic front end, and is referred to as feature extraction. The acoustic front end is a software module that incorporates

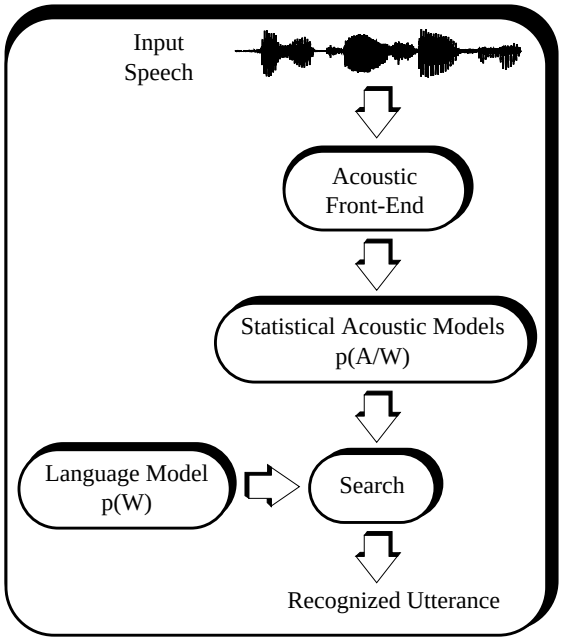

Figure 2. The four main components in a typical speech recognition system.

a set of signal modelling techniques [9] to convert a digital speech signal to a sequence of vectors. Modern speech recognition systems typically produce a feature vector every 10 msec. The design of this component is described in greater detail in chapter 2.

In real applications, front ends must incorporate advanced features, such as quantization and compression, in order for systems to operate at acceptable levels of performance and efficiency. The Aurora working group has been developing a reference client/server architecture shown in Figure 3 for speech recognition applications. The goal of the ETSI Aurora large vocabulary (ALV) evaluation was to standardize front end processing within this architecture for large vocabulary speech recognition applications. A large vocabulary speech recognition system is generally considered to be a system that uses some form of sub-word acoustic modeling [10] and is capable of recognizing tens of thousands of words spoken continuously [11,12,13].

The ALV evaluation was the second in a series of evaluations designed to promote the development of the AFE. The objective of the first Aurora evaluation was to calibrate

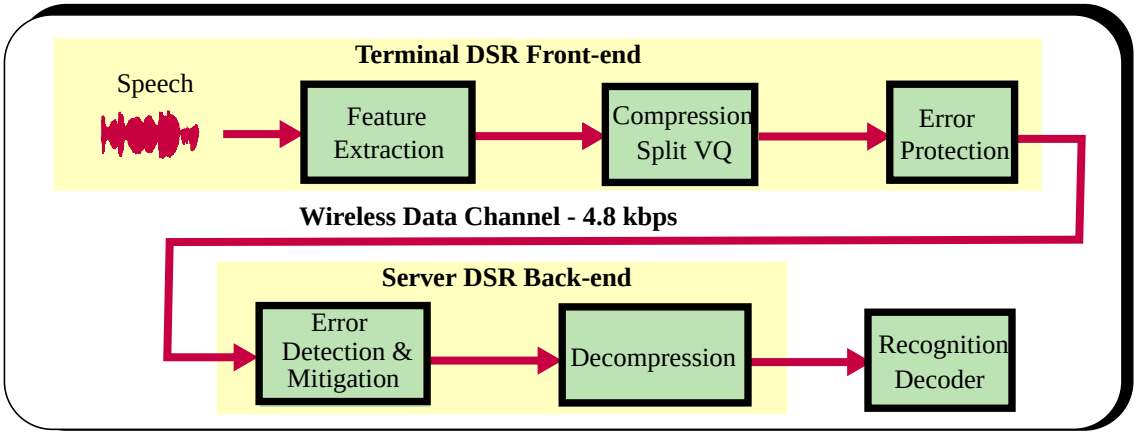

Figure 3. The Aurora standard for a DSR architecture includes provisions for compression and error protection along with feature extraction.

the performance of Mel Frequency Cepstral Coefficients (MFCC) based front ends [14,15] on small vocabulary tasks that use word models. The results of this evaluation showed that the performance of both the ETSI WI007 and HTK front ends degraded heavily under simulated noisy environment. The details of these evaluations are discussed in section 1.2. First, let us review three popular client/server architectures for automatic speech recognition in a mobile environment.

1.1. Architectures for Mobile Speech Recognition Applications

There are three popular architectures for speech recognition applications [6]. These three architectures are classified on the basis of the distribution of the computational resources between the client and the server — terminal-only, server-only, and terminal/server. A terminal-only architecture implements speech recognition on a user's terminal device, often referred to as the client. This is depicted in Figure 4. Because the complete process of recognition is performed on the terminal device with no transmission of recognition-related data involved, this architecture is robust to various artifacts of the communication channel such as transmission errors, channel errors, interference noise and compression. However, applications for this architecture are typically limited to small recognition tasks such as isolated words or phrases because of the limited computational power and memory availability on a portable terminal device. Voice dialing on cellular phones is a popular example of an application on such an architecture.

A server-only architecture involves transmission of speech over a noisy communication channel to a back-end speech recognition server, as shown in Figure 5. The complete process of speech recognition including feature extraction and recognition is performed on the server. Because of the availability of ample computation power and memory resources on the server, complex voice interfaces, such as spoken information retrieval [16], can be implemented. Such applications are popular in large-scale telephony applications, but are not popular in mobile applications because of the great demand for communications bandwidth between the server and terminal device. Typically the speech signal is compressed (coded) before the transmission over the wireless channel to conserve bandwidth. Compression and other characteristics of noisy communication channels (e.g., interference noise and packet loss) result in a significant degradation in speech recognition performance [1,6,17,18,19]. Various channel correction algorithms, such as error detection, packet reconstruction and channel adaptation, are used to reduce the influence of channel artifacts. However, the degradation in recognition performance is not completely alleviated.

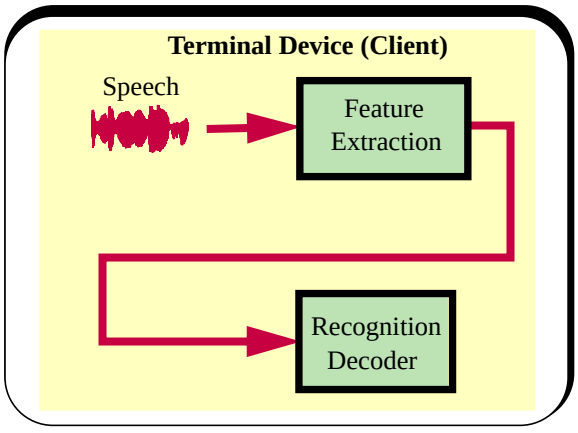

Figure 4. Terminal-only architecture.

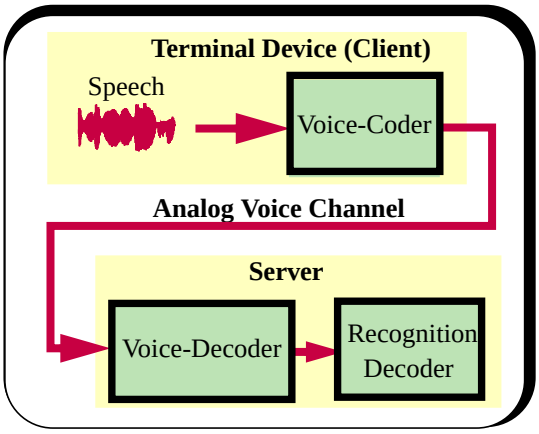

Figure 5. Server-only architecture.

The third architecture, which is the subject of investigation in the Aurora evaluations and the primary focus of this work, distributes processing to both the terminal and server sides. We refer to this architecture as Distributed Speech Recognition (DSR). This architecture combines the advantages of both the terminal-only and server-only architectures by distributing the computational resources between the two devices. The features are extracted on the terminal which are transmitted over the channel. Because these features, and not the speech samples, are digitally transmitted over the noisy channel, the influence of the artifacts of the noisy channel is minimal on recognition performance.

In a typical DSR architecture, such as the Aurora standard shown in Figure 3, features are extracted from the speech signal on the terminal device. This process is often coupled with noise enhancement schemes [20] that require a minimal amount of processing power. Unlike the server-only architecture, the noise enhanced features are then compressed and transmitted digitally over the error-protected channel, resulting in a significant reduction of channel-induced errors. Sophisticated model compensation and natural language modules can be employed on the server to improve speech recognition performance.

1.2. An Overview of the Aurora Evaluations

A major challenge for DSR architectures is the standardization of the front end [21]. Such a standard is required to be robust to the demanding conditions encountered in practical applications such as cellular telephony. It also needs to be robust

to variations in languages. The DSR group of ETSI has been actively involved in an effort to standardize an advanced front end for cellular telephony [7,8]. To achieve this objective, the DSR working group has conducted a series of evaluations of noisy speech constructed using simulated as well as actual noisy environments. The objective of the first Aurora evaluation was to calibrate the most popular speech recognition front ends for small vocabulary speech recognition applications. Two front ends were evaluated on the Aurora-2 database [22,14], which is simply a noisy version of the small vocabulary TIDigits task [23].

The original 16 kHz studio quality TIDigits database was downsampled to 8 kHz and filtered through G.712 characteristic [24] to simulate the Global System for Mobile Communications (GSM) terminal characteristic. Eight different noise types (suburban train, babble, car, exhibition hall, restaurant, street, airport and train station) were added in a controlled fashion to cover a range of signal to noise ratios (SNRs). The range included a no noise condition, referred to as the “clean” condition, and the following SNRs: 20, 15, 10, 5, 0, and -5 dB.

Two training sets, referred to as clean and multi-condition, were defined. The clean training set does not contain any additive noise. The multi-condition training set is representative of four noise types (suburban train, babble, car, and exhibition hall) covering all seven SNR ratios. Three test sets, denoted Test Sets A, B and C, were also defined. Test Set A is representative of all four noise types seen in the multi-condition training set. Test Set B is representative of four noise types not represented in the multi-

Table 1. In the first Aurora evaluation, the performance of the ETSI WI007 front end was shown to be comparable to the HTK front end on all test conditions.

Front end	Training Set	Test Set A	Test Set B	Test Set C
WI007	**Clean**	39.9%	45.0%	36.0%
	Multi-condition	12.2%	14.2%	17.4%
HTK	**Clean**	38.9%	44.4%	33.3%
	Multi-condition	12.7%	14.5%	16.9%

condition training set. Test Set C is filtered through M-IRS filtering [107] to introduce convolutional noise. It contains two noise types (suburban street and train).

For the first evaluation on Aurora-2 database, word-based models were employed. The results of this evaluation for two training conditions is summarized in Table 1. It is evident from the results presented in Table 1 that there was no significant difference in the performance between the two front ends. However, even on the multi-condition training set, which contains ample samples of noisy speech encountered during decoding (Test Set A), the performance is approximately 13.0% WER. State of the art on studio-quality TIDigits is approximately 0.2% [25]. Hence, the performance of these popular front ends degrades by an order of magnitude. For many practical applications, such as cellular telephony, this degradation due to noise is unacceptable. This observation motivated the development of an advanced noise robust front end and a second Aurora evaluation.

The second Aurora evaluation was conducted with a goal to improve performance in noisy environments. The performance of the advanced front end (AFE) was required to be no worse than the ETSI WI007 front end [15,22] and significantly better in demanding

environments. These performance goals are shown in Figure 6. Other general requirements for this evaluation were:

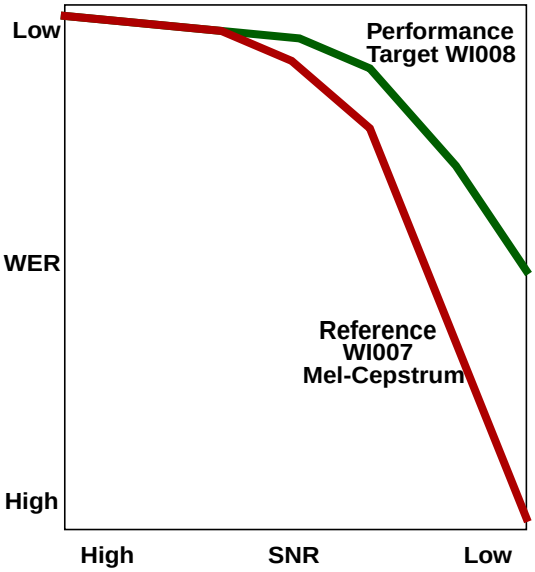

Figure 6. Target performance for second Aurora evaluation.

- extension to a range of European languages (Aurora-3 database; specifically the SpeechDat-Car [26] subsets in Finnish, Italian, Spanish, German and Dannish);
- extension to a large vocabulary task (Aurora-4 database);
- coverage of a range of background noise types typical of the cellular telephony environment;
- compatibility with HMM-based back-end recognizers (word and sub-word based).

More specific requirements are described in [27].

Two databases were defined to cover these requirements — Aurora-3 and Aurora-4. The Aurora-3 database was designed to calibrate the AFE performance in real noisy environments. It included five European languages to calibrate robustness to variation in language. This database was a small vocabulary task selected from a larger SpeechDat-Car database [26] that is recorded in automobiles in motion. Each of the language sets consists of three training sets and corresponding test sets designed to calibrate the following conditions:

- **Well-matched condition:** Both the training and test sets are recorded with the same hands-free microphone over the similar range of vehicle speeds to cover the same noise conditions;

- **Moderate mismatch condition:** The training set consists of a subset of the range of noise types seen in the test set;
- **High mismatch condition:** The training set is recorded with a close-talking microphone while the test set is recorded with a hands-free microphone.

The Aurora-4 database was designed to study performance on a large vocabulary task, namely the WSJ0 subset [28] of the Wall Street Journal Corpus. Seven additive noise conditions (clean, street traffic, train station, car, babble, restaurant and airport) randomly chosen from a range of SNRs and two filtering schemes were employed to simulate the noisy terminal characteristics. This database is the subject of this thesis. The experimental design of the ALV evaluation using this database is discussed in detail in chapter 3.

The ALV evaluation formed a significant portion of the second Aurora evaluations. The goal for the ALV evaluation was to achieve a 25% relative improvement in word error rate (WER) across a variety of noise conditions compared to the MFCC WI007 front end. Two consortia submitted proposals on speech recognition front ends for the ALV evaluation: (1) Qualcomm, ICSI, and OGI (QIO) [29], and (2) Motorola, France Telecom, and Alcatel (MFA) [30]. These advanced front ends used a variety of noise reduction and channel normalization techniques including discriminative transforms, spectral subtraction, feature normalization, voice activity detection, and blind equalization. These noise robust algorithms are discussed in detail in chapter 2.

1.3. Comparison to Previous Speech in Noise Evaluations

The Spoke tasks in 1994 DARPA Continuous Speech Recognition (CSR) evaluations [31] were designed to test a number of challenging conditions involving

adaptation and compensation. These evaluations represented an important milestone in CSR research. In particular, the evaluations referred to as Spoke 5 and Spoke 10 were designed to benchmark algorithms that compensate for channel mismatch and additive noise conditions. Both of these evaluations were derivatives of the same WSJ0 5K task used in the Aurora evaluation. A common language model, identical to the one used in this thesis, was used for these evaluations.

Spoke 5 involved the use of unsupervised channel compensation for a variety of microphones. The baseline microphone condition was a close-talking Sennheiser microphone and the channel mismatch condition consisted of 10 different microphone types. These microphones included four tie-clip microphones, three stand-mounted microphones, two desktop microphones, and one hand-held microphone. Only Carnegie Mellon University (CMU) participated in this evaluation. Two different channel compensation algorithms [32] were evaluated. Without any compensation, the baseline CMU system achieved a WER of 12.4%. The best CMU system with compensation enabled achieved a WER of 9.7% which is about a 20% relative improvement over the no-compensation case. However, this is still 45% worse than the Sennheiser microphone condition, which had a WER of 6.7%.

While Spoke 5 was designed to benchmark channel-mismatch compensation algorithms, Spoke 10 involved the compensation of “clean” data (recorded on close-talking Sennheiser microphone) corrupted with additive noise. Three SNR levels were included: 22 dB, 16 dB, and 10 dB. Three sites participated in Spoke 10 evaluation: Cambridge University (CU), IBM, and SRI International. These groups used model-based

Table 2. A summary of performance on the Spoke 10 task in the DARPA 1994 CSR evaluation.

Condition	CU	IBM	SRI
Baseline (clean)	7.2%	7.2%	6.7%
Without compensation (10 dB SNR)	84.7%	77.4%	35.4%
With compensation (10 dB SNR)	19.8%	12.8%	12.2%

approaches for noise compensation [31,33]. From the results shown in Table 2, it is evident that the best system from SRI achieved a WER of 12.2%, and shows an improvement of 66% relative to the no-compensation case for the worst SNR condition (10 dB). However, this system suffered from a 83% relative degradation when compared to the “clean” baseline condition.

The most important difference from an experimental design point of view between the ALV evaluation and the DARPA 1995 CSR evaluations (Spoke 5 and 10) was the desire to use a fixed recognizer in the ALV Evaluation. The goal of the ALV evaluation was to benchmark signal enhancement approaches in the front end component of a speech recognition system, while the 1995 CSR evaluations allowed model-based approaches for noise compensation within the recognizer. An important design constraint for the ALV evaluation was the fact that the participating advanced front ends were required to meet the ETSI latency requirements [15]. Other significant differences are tabulated in Table 3.

As we will observe in chapter 5, the advanced front end with the best performance on the ALV evaluation achieved a WER of 34.5% (averaged across all conditions). This advanced front end performance represents a 130% relative degradation when compared

Table 3. A comparison of the experimental setup between the DARPA 1995 CSR evaluations (Spoke 5 and 10) and the ALV evaluation.

Condition	DARPA 1995 CSR Spoke	ALV
Additive Noise Conditions	One condition: car	Seven Conditions: clean, street traffic, train station, car, babble, restaurant, and airport
SNR Levels	Three levels: 22 dB,16 dB,10 dB	Randomly chosen between: 10-20 dB for training sets; 5-15 dB for test sets
Evaluation Tests	Two Tests: Spoke 5 for mic. mismatch; Spoke 10 for additive noise	Mixed microphone mismatch and noise conditions. Terminal filtering of the data.

to the performance on the matched microphone (Sennheiser microphone) and clean (without noise) conditions. On the CSR evaluation, an 83% degradation for the noise conditions was observed. The range of noise and microphone conditions on the CSR evaluation was limited compared to Aurora. Hence, the slightly larger degradation in the Aurora evaluation is not unexpected.

1.4. Thesis Scope and Contributions

The primary goal of this thesis is to analyze and evaluate the noise robustness algorithms employed in the QIO and MFA advanced front ends. Though these advanced algorithms improve recognition performance significantly over the MFCC baseline front end, this improvement is not operationally significant. It has been shown in several studies that human performance is stable for SNRs as low as 10 dB [34,35,36]. Machine

performance degrades rapidly below 15 dB SNR. Despite the progress made recently in the development of noise robust front ends, there are still significant challenges ahead in closing the gap between machine and human performance.

The key contributions of this thesis are:

- **Development of the Aurora baseline system:** This system was designed to minimize computation time without significantly compromising the overall system performance or the ability of the evaluation to rank front end algorithms. The baseline system achieved a WER of 14.0% on the standard 5K WSJ0 task, and required 4 xRT for training and 15 xRT for decoding (on an 800 MHz Pentium processor).

- **Analysis of the WI007 (MFCC) front end:** The performance of the WI007 front end on six focus conditions is calibrated and analyzed. These six focus conditions are: sampling frequency reduction (16 kHz and 8 kHz), utterance detection (influence of endpointing), compression (a vector quantization-based compression scheme), model mismatch (mismatched training and testing conditions), microphone variation (two microphone conditions available in the WSJ0 task [28]), and additive noise (six noise types collected from street traffic, train stations, cars, babble, restaurants and airports at varying signal-to-noise ratios).

- **Analysis of noise robustness algorithms:** A theoretical analysis of techniques that reduce degradations due to convolutional and additive noise is provided for the QIO and MFA front ends. The performance of these front ends is also evaluated on the Aurora 4 task and compared to a baseline MFCC front end. It is shown that the performance of these AFEs is significantly (statistical sense) better than the MFCC front end.

- **Parameter Tuning:** The influence of front end-specific parameter tuning on performance is calibrated and analyzed. It is shown that the front end-specific tuning does not significantly influence recognition performance for the ALV evaluations described in this thesis.

The Aurora baseline system allows users to calibrate the performance of AFEs through extensive experimentation in a reasonable amount of time without the need for a large cluster of compute servers. The large gap between the performance of the AFEs and

humans establishes the need for further research towards the development of better noise robust algorithms. Because the experimentation was performed without any front end-specific parameter tuning, it can be argued that the performance obtained by these AFEs is suboptimal. Optimizing well-known recognition system parameters [37] such as the language model scale and word insertion penalty often improves performance. This thesis establishes that front end-specific parameter tuning does not result in a significant improvement in recognition performance for the algorithms analyzed.

1.5. Structure of the Thesis

Chapter 2 discusses the need for perceptually-motivated signal parametrization and reviews an industry-standard feature extraction algorithm used in the baseline system and the Aurora WI007 standard. It also presents an analysis of the advanced noise robust algorithms used in the QualComm-ICSI-OGI (QIO) and Motorola-FranceTelecom-Alcatel (MFA) front ends. Chapter 3 describes the experimental framework used in the ALV evaluations. A detailed discussion is presented on the development of the short training and test sets that were used to facilitate large scale evaluations. Chapter 4 presents the design and development of the ALV baseline LVCSR system that was used a common testbed to benchmark the performance of the front ends. Chapter 5 presents the experimental results in the ALV evaluation, and an analysis of attempts to optimize the performance of these front ends with the baseline system. It is shown that the performance of both QIO and MFA front ends is significantly better than the baseline MFCC front end,

but this improvement is not operationally significant. Chapter 6 summarizes the key contributions of this work and suggests some directions for future research.

CHAPTER II

FRONT END ALGORITHMS

The term “front end” in the speech recognition literature is commonly used to describe a collection of signal modeling techniques [9] that transform an audio signal into a sequence of feature vectors as shown in Figure 7. These features capture the spectral and temporal variations of the speech signal. Many signal modeling techniques are designed to approximate human auditory phenomena known to be an integral part of the human speech recognition apparatus.

Front end design has been an area of active research for the past quarter century. The two dominant front end approaches in speech recognition are based on the mel frequency cepstral coefficient (MFCC) representation [38] and perceptual linear prediction (PLP) [39]. The popularity of these two front ends is attributed to their ability to deliver good performance while maintaining a fairly simple and

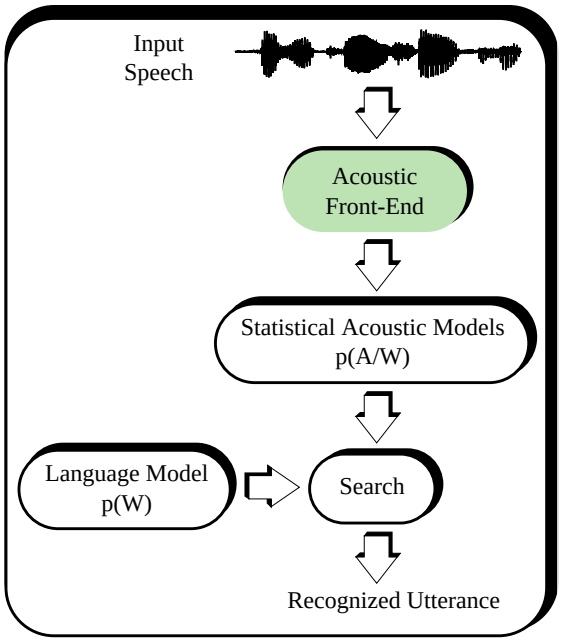

Figure 7. A front end converts a speech signal to a sequence of feature vectors that serve as input to the acoustic modeling component of a speech recognizer.

computationally efficient implementation in a real-time framework. The PLP front end has been reported to perform marginally better than the MFCC front end in demanding environments though it has been shown that after a few passes of adaptation, the performance of both front ends is comparable [40].

This chapter presents overviews of the three approaches studied in this thesis: an industry-standard MFCC front end and two advanced front ends featuring algorithms intended to improve robustness to noise. It also describes the differences between a standard MFCC front end (ISIP) [41] and the WI007 front end [15] that was used as a baseline for the ALV evaluation.

2.1. The Mel-Scaled Cepstral Coefficient Front End

A detailed discussion of various signal modeling techniques used in modern speech recognition systems can be found in [9,42,43]. The most popular approach for transforming the input signal into a sequence of feature vectors uses the mel-scale frequency cepstral coefficient (MFCC) representation [41] shown in Figure 8. In the following sections, we briefly describe each component in this block diagram.

2.1.1. Zero-mean

The first step in conversion of the speech signal to a feature vector is to remove the the DC offset, a process referred to as debiasing of the signal. A mean value is computed every 10 msec using an overlapping 25 msec window. This analysis window begins at the same time as the 10 msec frame, and extends 15 msec after the end of the frame. This is

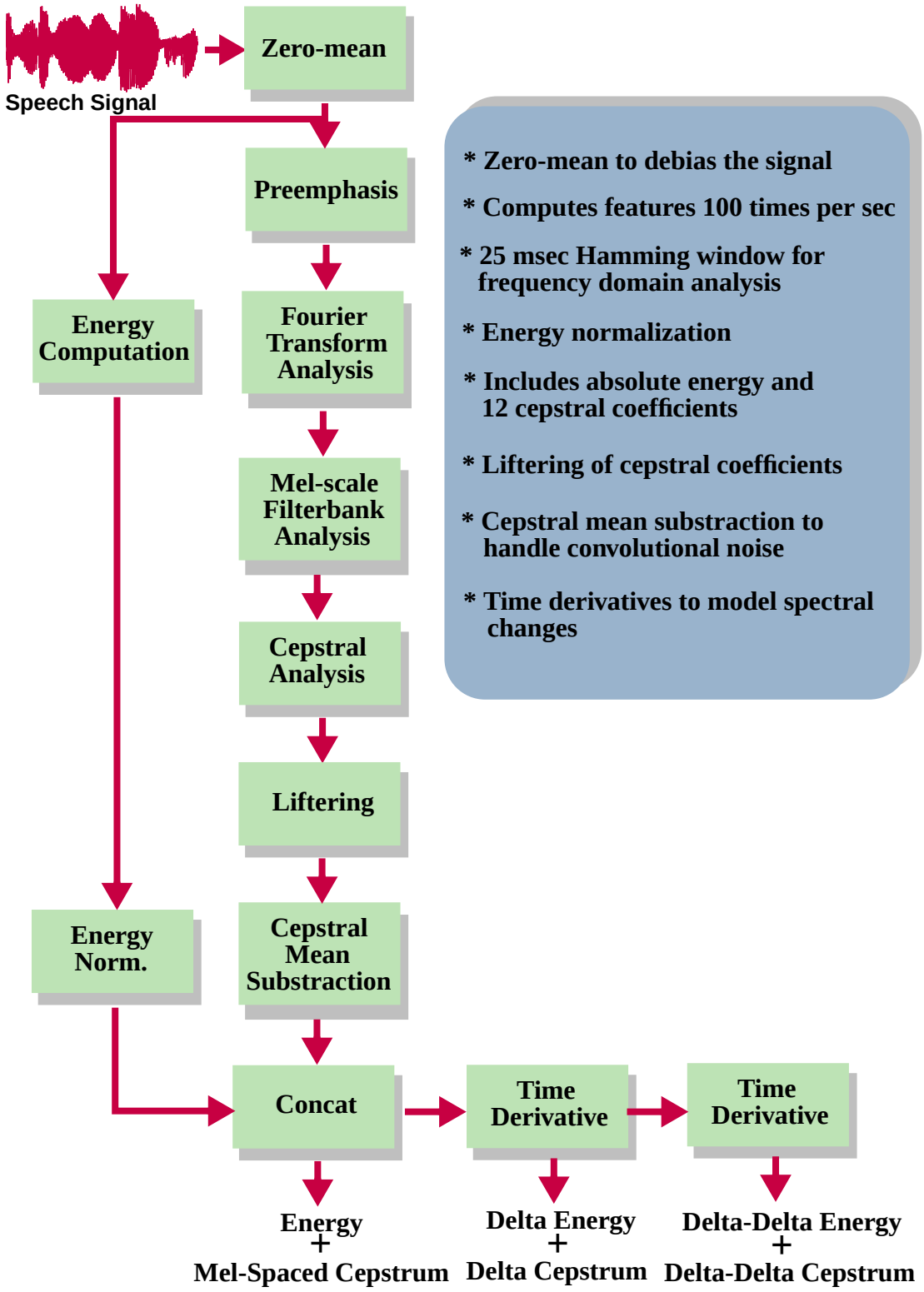

Figure 8. A signal flow graph describing an MFCC front end.

often referred to as a left-aligned window. The mean value computed over this window is subtracted from the signal:

$$x_d[n] = x[n] - \mu_x \ , \tag{1}$$

where μ_x is the mean of the speech sample values within a window, $x[n]$ represents the input speech samples, $x_d[n]$ represents the debiased speech samples, $n = 1, 2, \ldots, N$, and N is the total number of speech samples in the window.

2.1.2. Preemphasis

The next step is to shape the spectrum of the debiased signal using a first-order finite impulse response (FIR) filter given by:

$$H_{pre}(z) = 1 + a_{pre}z^{-1} \ . \tag{2}$$

This filter amplifies portions of the spectrum above 1 kHz at approximately 20 db/decade. Because the human auditory system is more sensitive to frequencies above 1 kHz, preemphasis tends to increase the contribution of the high frequency portion of the spectrum in the overall recognition process [44]. A typical range for a_{pre} in speech recognition applications is [-0.4,-1.0]. The value of this filter coefficient used in the ETSI WI007 front end is 0.97. The computation of the first output sample at time zero, $y[0]$, at each frame varies from one front end implementation to the other. For example,

reference implementations at Mississippi State University [41] and Cambridge University [50] compute $y[0]$ at every frame by assuming $x[-1] = x[0]$.

On the other hand, the ETSI WI007 front end does not make any assumption and computes the $y[0]$ using a circular buffer implementation. Hence, $x[-1]$ needed for the computation of the first sample at each frame is retrieved from the previous frame in the circular buffer. The value of $x[-1]$ needed of the computation for $y[0]$ corresponding to the first output sample for the first frame is assumed to be zero.

2.1.3. Fourier Transform Analysis

The next two processing steps, frequency domain analysis and an absolute energy computation (described in section 2.1.8), are performed in parallel. The signal is transformed from the time domain to the frequency domain using a Fourier Transform (FT). A 25 msec Hamming window is used which corresponds to a spectral resolution of 40 Hz. This choice of a window duration captures the short-term spectral envelope of the speech signal, which is related to the vocal tract shape, while ignoring the spectral harmonics corresponding to the fundamental frequency of the speech waveform [9,43]. A feature vector that represents the time-varying vocal tract shape is critical to achieving high performance speaker independent speech recognition. In practice, a Fast Fourier Transform (FFT) is used to compute the transform because of its computational efficiency [45]. The 25 msec window is zero-padded to the nearest power of two (e.g., for an 8 kHz sample frequency, we use a 256 point FFT).

This spectral estimate of the speech signal is computed every 10 msec. Due to the limited velocity of the articulators in the human speech production apparatus, the speech signal can be regarded as relatively stationary when analyzed over a 5 to 10 msec interval [44]. A 10 msec frame duration has been historically used in speech recognition systems [9] as a compromise between computational efficiency and the temporal resolution necessary to assume the speech signal is stationary.

2.1.4. Mel-scale Filter Bank Analysis

The Fourier Transform (FT) of the signal is then transformed using a mel-scale filter bank analysis. The human auditory system is known to be sensitive to frequency and amplitude on a logarithmic basis. This behavior can be approximated by transforming the signal using a nonlinear frequency scale. Though there are techniques to perform such a scaling in the time domain [46,47], it is more convenient to simply implement this as a table-lookup in the frequency domain. The mel scale [47] is a popular approximation for this non-linear mapping, and is given by:

$$f_{mel} = 2595 \cdot \log 10(1 + f / 700.0) \ . \qquad (3)$$

A logarithmic filter bank analysis is used to approximate the sensitivity of the basilar membrane of the human ear to discrete frequencies [48]. Instead of perceiving individual frequencies on a continuous scale along the basilar membrane, there is evidence that hair cells along this membrane are tuned to specific frequencies. A bank of 24 bandpass overlapping filters arranged linearly along the mel scale represents a crude, but adequate approximation to the frequency resolution of the human ear. The output of each

of the overlapping filters is computed as the weighted sum of the FT coefficients that fall within its bandwidth:

$$S_{avg}(n) = \frac{1}{N_s} \sum_{n=0}^{Ns} w_{FB}(n)|S(f)| , \quad (4)$$

where N_s represents the number of coefficients within the filter width, $w_{FB}(n)$ represents the weighting function (filter gain), and $S(f)$ represents the frequency response given by the FT. A triangular weighting function is the most common form for $w_{FB}(n)$ [38]. Figure 9 depicts a mel-scale triangular filter bank implementation which is used in most MFCC front ends.

2.1.5. Cepstrum Analysis

After the mel-scale filter bank analysis, a cepstrum analysis is performed on the filter bank outputs. Cepstrum analysis is a homomorphic process [48] that is applied to deconvolve the excitation and the vocal tract shape. Speech production can be approximated as the convolution of two impulse responses:

$$s(n) = g(n) \otimes v(n) , \quad (5)$$

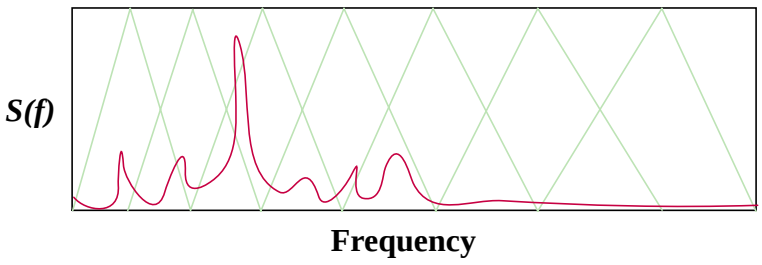

Figure 9. Mel-frequency spaced triangular filter banks for an MFCC front end.

where $s(n)$ represents the speech signal, $g(n)$ represents the excitation, and $v(n)$ represents the vocal tract shape. Only the vocal tract shape information is exploited for speaker independent recognition.

The corresponding frequency domain representation is given by the product of two components:

$$S(f) = G(f) \cdot V(f) \ . \tag{6}$$

This product can be represented as a sum in log domain:

$$\log\{S(f)\} = \log\{G(f)\} + \log\{V(f)\} \ . \tag{7}$$

The spectrum of the excitation signal and the vocal tract shape can be separated using conventional digital signal processing techniques in the log-frequency domain. The vocal tract shape is represented by the low-order cepstral coefficients, while the high-order coefficients contain the spectral information corresponding to the excitation signal. Typically, for most speaker independent speech recognition applications, only the first 13 coefficients (low-order) are retained for further processing.

The classical cepstrum is defined as the inverse Discrete Fourier Transform (IDFT) of the log magnitude spectrum [43]. In a typical MFCC front end, the cepstrum is implemented using a Discrete Cosine Transform (DCT) because the log magnitude spectrum is a real symmetric function [43,50]:

$$C[k] = \sqrt{\frac{2}{N}} c_n \sum_{n=0}^{N-1} S_{avg}(n) \cos\left[\frac{\pi n(2k+1)}{2N}\right], \tag{8}$$

where N represents the total number of filter banks, $k = 0,1,\ldots,N$, $c_n = 1/\sqrt{2}$ when $n = 0$, and $c_n = 1$ elsewhere. The resulting coefficients are an approximation to the classical cepstrum, and compactly represent the log magnitude spectrum of the speech signal. The first thirteen cepstral coefficients are typically adequate to describe the vocal tract shape for most speech recognition applications.

2.1.6. Liftering

The thirteen cepstral coefficients are then weighted using a process known as liftering [38]. While the low-order cepstral coefficients represent the vocal tract shape (e.g., spectral slope and glottal pulse shape), the high order cepstral coefficients are sensitive to the analysis window, fundamental frequency and other artifacts [51]. Hence, for speaker independent recognition, it is advantageous to reduce these speaker-dependent variations. The low-order coefficients are enhanced though a raised-sine weighing function given by:

$$w(n) = G\left\{ \begin{array}{ll} 1 + h\sin((n\pi)/L) & 1 \le n \le L \\ 0 & elsewhere \end{array} \right\}. \qquad (9)$$

Typical values for the parameters G, L, and h in an industry-standard MFCC front end are 1, 22 and 11, respectively.

The zeroth cepstral coefficient, $c[0]$, represents the average value of the spectrum or the root mean square value of the signal [9]. Historically, this term is excluded from the

set of cepstral coefficients. Instead, an absolute energy term, described below in section 2.1.8, is explicitly computed.

2.1.7. Cepstral Mean Subtraction

A simple technique to reduce the influence of convolutional noise due to channel and/or microphone distortion is cepstral mean subtraction (CMS) [52]. CMS is performed on the 12 cepstral coefficients:

$$c_{cms,\,n}[k] = c_n[k] - \mu[k] \,, \tag{10}$$

where $\mu[k]$ represents the mean of the k^{th} cepstral coefficient, $k = 1, 2, \ldots, 12$, $n = 1, 2, \ldots, N$, and N is the total number of frames in the speech utterance. The mean of each cepstral coefficient is computed over an entire utterance (e.g., speech file) or a conversation side [53] depending on the nature of the application.

2.1.8. Absolute Energy and Energy Normalization

The log of the absolute energy term is explicitly computed once per frame using the 25 msec analysis window of the input speech prior to the preemphasis step:

$$E = \log_e\left(\sum_{n=0}^{N-1} x^2(n)\right), \tag{11}$$

where N represents the total number of samples in the 25 msec window.

The logarithm of the absolute energy is then normalized on an utterance basis to reduce the variations in energy levels that may arise due to variation in loudness levels of different speakers:

$$E_{norm}[n] = E[n] - E_{max} , \tag{12}$$

where E_{max} represents the logarithm of maximum energy, $n = 1, 2, \ldots, N$, and N is the total number of frames in the speech utterance. The term representing the normalized log-energy and the twelve CMS-transformed cepstral coefficients are concatenated to form a 13-dimensional absolute feature vector.

2.1.9. Time Derivatives

The final step in the MFCC front end is the computation of the first and second-order time derivatives of the 13-dimensional feature vectors. These time derivatives improve our ability to discriminate between certain classes of sounds, and capture some of the temporal characteristics of the speech signal [54,55]. Linear regression analysis is used to generate these derivatives [56,57]:

$$d_n[k] = \frac{\sum_{w=1}^{dw} w\left\{\left(c_{n+w}[k]\right) - \left(c_{n-w}[k]\right)\right\}}{2\sum_{w=1}^{dw} w^2} , \tag{13}$$

where $d_n[k]$ is a scalar value representing the derivative of the k^{th} feature vector coefficient at frame n, $c_{n+w}[k]$ and $c_{n-w}[k]$ represent past and future values of this

coefficient in time and dw is the number of frames used in the computation. Two adjacent frames on each side of the current frame are sufficient to capture the velocity of the cepstral coefficients. Hence, dw is set to 2. As shown in Figure 10, a five-frame window is needed for the first-order derivative computation.

The second-order derivatives are computed by applying Equation 13 to the output of the first differentiation, with dw set to 2. Hence, the acceleration of the cepstral coefficients is computed by a differentiation of the first-order derivatives. The total extent of the data involved in the second derivative calculation is nine frames of data, or 90 msec. Thus, the overall feature vector for the MFCC front end contains 13 absolute features (energy plus 12 cepstral coefficients), 13 first-order derivatives (velocity) of these absolute features, and 13 second-order derivatives (acceleration), resulting in a feature vector with a dimension of 39.

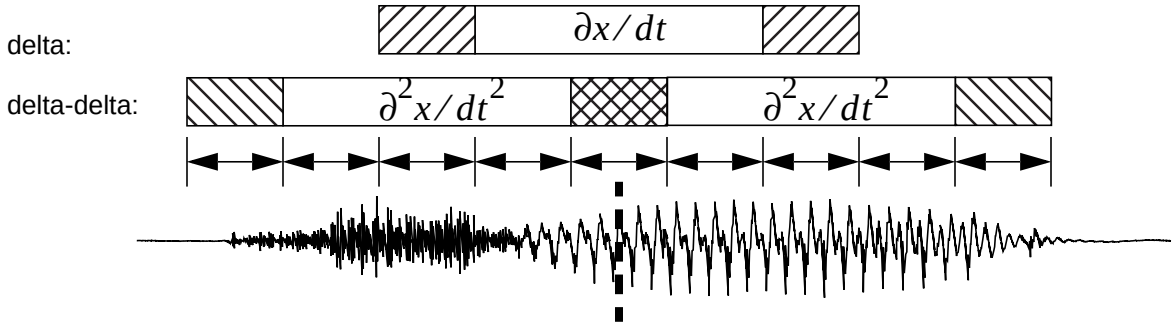

Figure 10. Each temporal derivative is computed using a five frame window (at 10 msec per frame). Hence, the second derivative computation, which requires five frames of first derivative data, involves data extending over nine frames of the input signal.

2.2. The ETSI WI007 Front End Specification

The WI007 MFCC front end [15], shown in Figure 11, is a scaled down version of the standard MFCC front end described in the previous section. Liftering of cepstral coefficients is not implemented in the WI007 front end. Also, no energy normalization or cepstral mean subtraction is incorporated in this front end. However, the ETSI standard split-vector quantization compression algorithm and framing algorithm [15] is implemented in the WI007 front end. Only the 13 absolute features (energy plus 12 cepstral coefficients) are transmitted to the back end server though the channel. At the back end server, the bit-stream is decoded, error-detected, error-corrected, and decompressed to form the final features. The delta and acceleration coefficients are computed from the base features at the back end to form 39-dimensional feature vectors.

2.3. The QIO Advanced Front end

A collaboration between the CDMA Technologies Group at Qualcomm, the Speech Group at International Computer Science Institute (ICSI), and the Antropic Signal Processing Group at Oregon Health and Science University (OGI) produced a front end design referred to as the QIO front end [58]. It features three key components: a 15-dimensional MFCC-based feature vector generated using data-driven LDA-derived filters, on-line mean and variance normalization, and a multilayer perceptron-based voice activity detector (VAD).

A block diagram of the QIO front end is shown in Figure 12. The speech signal is analyzed using a 10 msec frame and a 25 msec window. For each frame of speech data, a

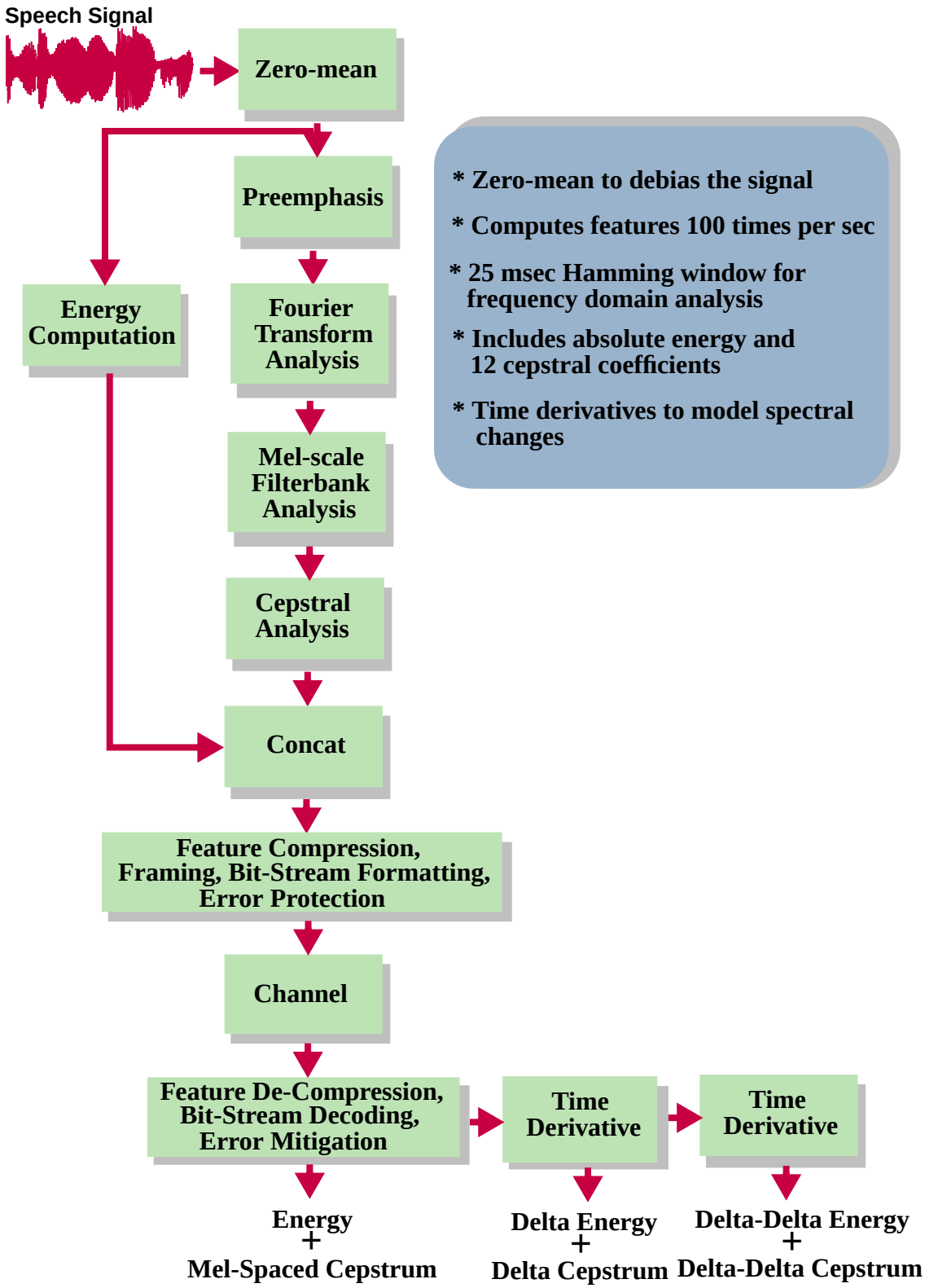

Figure 11. A signal flow graph describing the WI007 MFCC front end implementation.

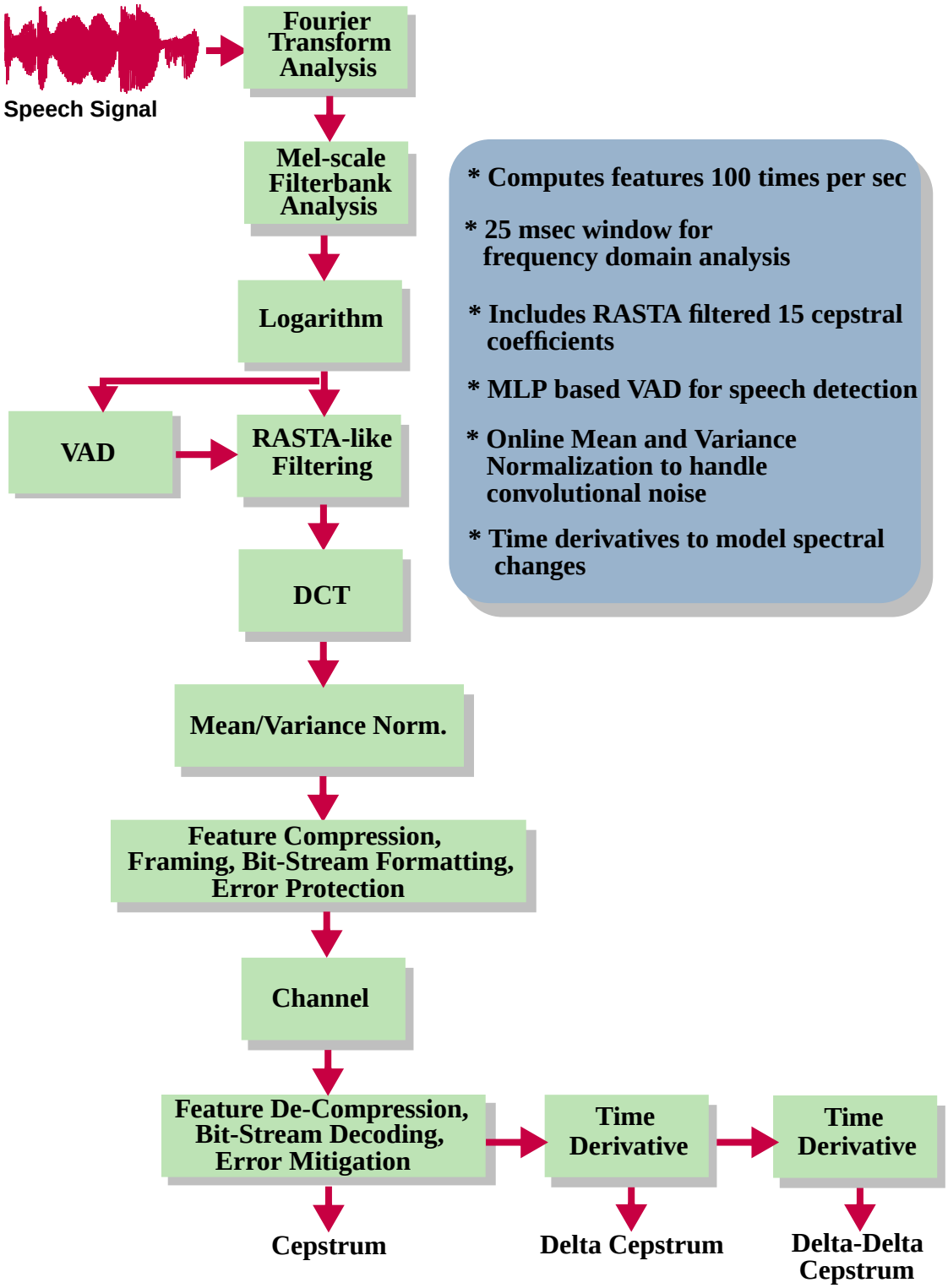

Figure 12. A block diagram of the Qualcomm-ICSI-OGI (QIO) front end.

mel-scaled triangular-weighted filter bank analysis, similar to the standard MFCC front end, is performed. However, the QIO front end uses 23 bins in its filter bank, compared to 24 for the standard MFCC front end. A natural logarithm of the output of each of the 23 bins is performed. Time trajectories of the 23 logarithmic filter bank energies are filtered though linear discriminant analysis (LDA) derived RASTA-like filters. In parallel, a VAD detector detects the speech and non-speech frames. The DCT of the speech frames is then computed and only the lower 15 cepstral coefficients are retained for further processing. An online mean and variance normalization of these 15 cepstral coefficients is performed.

These coefficients are processed through compression, framing, bit-stream formatting and error protection algorithms [15] on the terminal side. These processed frames are then transmitted over a digital channel. On the server side, the frames are processed though the ETSI standard bit-decoding, error mitigation and feature decompression algorithms. Delta and acceleration coefficients are computed on the server side using the 15 reconstructed cepstral coefficients. Thus, the overall dimension of the output feature vector is 45. The three key noise reduction techniques implemented in the QIO front end are described in the following sections.

2.3.1. LDA-derived RASTA-like Temporal Filtering

RASTA filtering is known to compensate for slowly varying convolutional noise introduced due to channel and/or microphone mismatch [59]. RASTA filtering involves temporal filtering of time trajectories of the log mel-frequency filter bank energies. The

overall influence of this filtering process is that it attenuates the frequencies of the filter bank energies below 1 Hz and above 12 Hz. Typically, the frequency response of the RASTA filter is optimized on a series of ASR experiments on a noisy database [59,60]. Because these optimizations are expensive and do not guarantee generalization, the filters used in the QIO front end were derived using a data-driven LDA analysis [60]. The frequency response of these LDA-derived filters match closely to the frequency response of the RASTA filter, and hence, the LDA-derived filters are referred as RASTA-like filters.

The LDA-derived filters are typically computed using a noisy training database and applied on a test database [60]. For the QIO front end, a noisy version of the OGI Stories database [59] was used to compute these filters. This OGI Stories database was corrupted by adding restaurant noise to achieve a 10 dB SNR.

The LDA-derived RASTA-like filters can potentially be derived for each of the log mel-frequency filter bank bins. For each bin, coefficients corresponding to a one-second time duration (e.g., 100 frames) are concatenated to form a sequence of feature vectors. A typical feature vector is formed every frame by using the value of a specific mel-frequency filter bank bin corresponding to the current frame plus the value of the same mel-frequency filter bank bin corresponding to 100 adjacent frames (50 past and 50 future frames). Each of these feature vectors can be interpreted as a center aligned overlapping window (current frame plus 50 past plus 50 future frames). An overlapping window corresponding to the i^{th} frame is shown in Figure 13. A sequence of these feature vectors is constructed by sliding the overlapping window by one frame over all the frames in the training database. The class of each of the feature vector is labeled by the phoneme

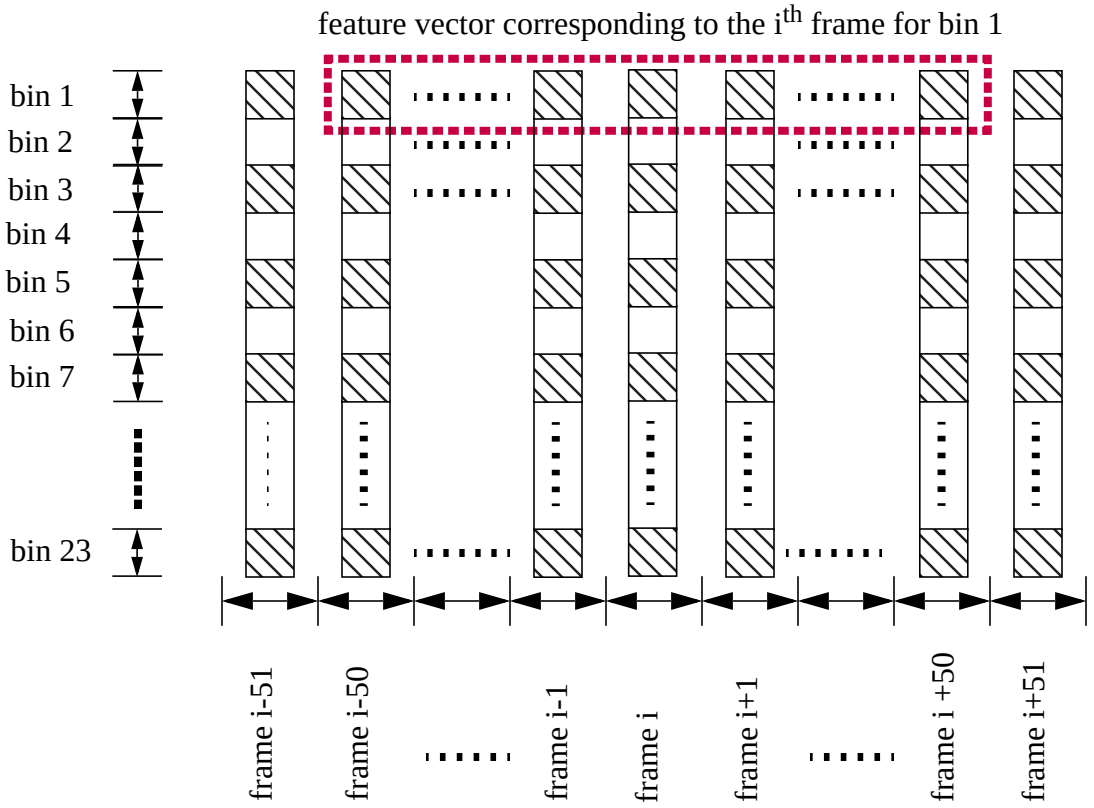

Figure 13. Feature-vector generation for LDA-derived RASTA-like filters.

corresponding to the current frame. LDA analysis is then applied to these 101 dimensional feature vectors that correspond to a specific bin. The transformation vector computed through this LDA analysis represents the coefficients of the RASTA-like filters for that specific bin.

Though each mel-frequency filter bank bin could potentially be filtered using a unique filter corresponding to this bin, only two filters are actually used in the QIO front end. Both of these filters are approximated as a 41-tap symmetric FIR to meet the ETSI latency requirement [8]. The filter corresponding to the second bin was selected and used

to filter the time trajectories of first and second filter bank bins. The remaining 21 bins were filtered though the filter corresponding to the fourth bin.

2.3.2. Voice Activity Detection

A multilayer perceptron (MLP) based voice activity detector (VAD) eliminates non-speech segments [61]. For extremely noisy speech, this reduces insertion errors by reducing the opportunity for noisy speech to be misinterpreted as speech. The input to this MLP consists of three frames of features. Two adjacent frames are used to incorporate contextual information during the decision-making process. The MLP consists of 6 input units, 15 hidden units and one output unit. A threshold is applied to the output posterior probability from the MLP to create a binary-valued output. This output is then smoothed using an 11-point median filter. The MLP was trained on multiple databases, representing both clean and noisy conditions.

2.3.3. Online Mean/Variance Normalization

Online mean and variance normalization is known to reduce the influence of the convolutional noise [52,62] such as channel distortion. The initial estimates of the mean and variance are computed using the first four frames of the training utterances. These initial estimates are then updated for each frame using the following equations:

$$\mu[t] = \mu[t-1] + \alpha(x[t-1] - \mu[t-1]) \ , \qquad (14)$$

$$\sigma^2[t] = \sigma^2[t-1] + \alpha((x[t] - \mu[t])^2 - \sigma^2[t-1]) \ , \quad (15)$$

$$x'[t] = \frac{x[t] - \mu[t]}{\sigma[t] + \theta} \ , \quad (16)$$

where $x[t]$ is a scalar representing a cepstral coefficient, and the $x'[t]$ is the corresponding normalized cepstral coefficient at frame t. The terms $\mu[t]$ and $\sigma^2[t]$ are the estimated mean and variance of $x[t]$. The constant α is an adaptation constant needed to guarantee a positive estimate of the variance. The scalar θ is an empirically-derived variance floor. For the ALV evaluation, these two parameters were set to 0.1 and 1.0, respectively.

2.4. The MFA Advanced Front end

The second advanced front end studied in this thesis resulted from a collaboration between the Human Interface Lab at Motorola Labs, France Telecom R&D, and Alcatel SEL AG (Germany). The front end produced by this collaboration is referred to as the MFA front end [63,64]. It is based on a 12-dimensional MFCC feature vector plus a weighted average of log-energy and the zeroth cepstral coefficient.

The input speech signal is first processed through a noise reduction block that uses a time domain two-stage Wiener filter, as shown in Figure 14. This analysis is performed on a frame basis with a frame duration of 10 msec, and uses a 25 msec Hanning window. The SNR of the resultant signal is then enhanced using a process referred to as “Waveform Reduction” that weights the speech segments of the speech signal higher than the non-speech segments through the use of Teagor energy operator. Thirteen cepstral coefficients

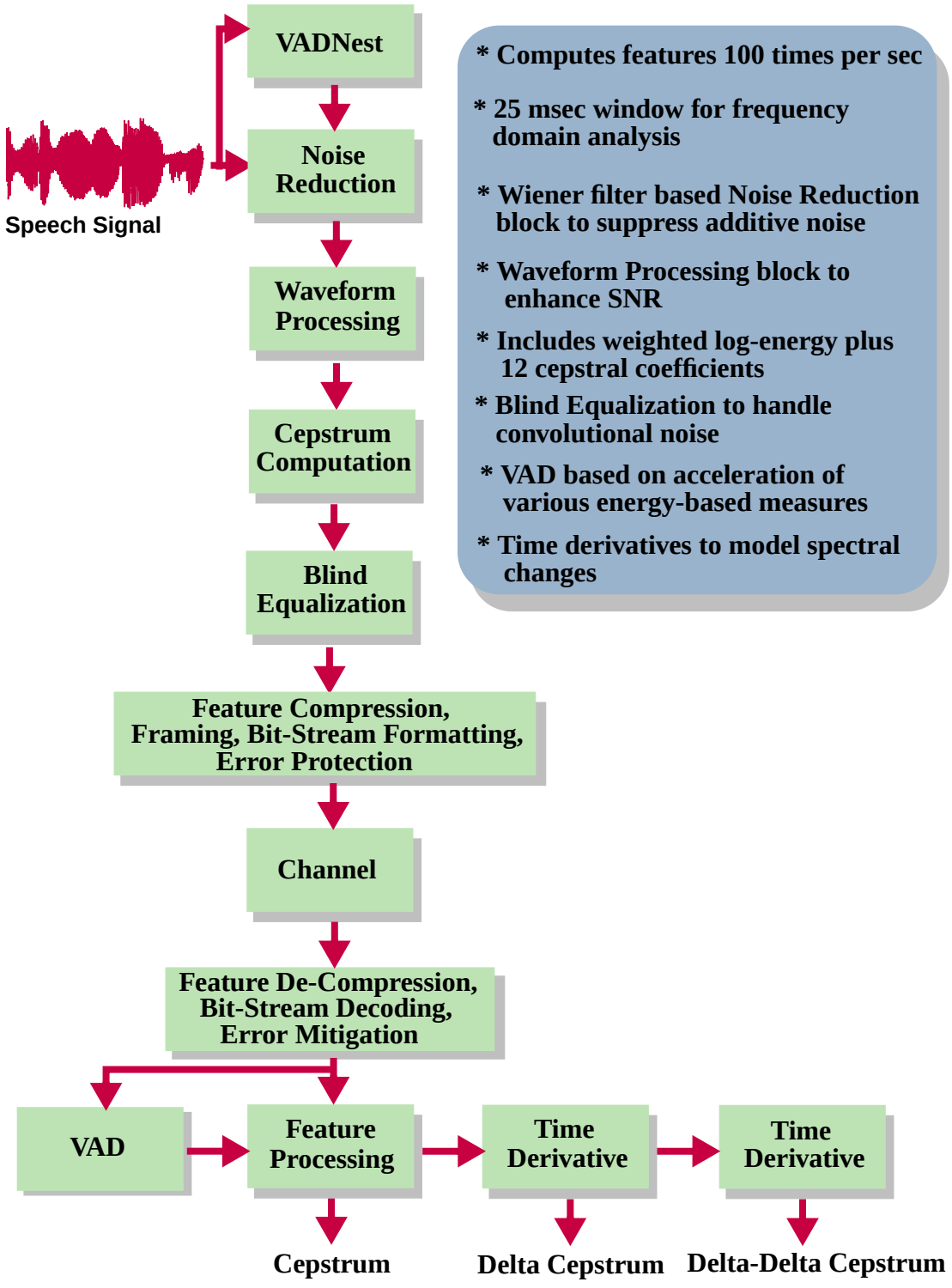

Figure 14. A block diagram of the Motorola-France Telecom-Alcatel (MFA) front end.

and energy are computed using the methodology described in section 2.1. A least mean square error-based blind equalization is applied to these coefficients to produce the final feature vector.

The resulting feature vectors are then processed though the ETSI standard feature compression, framing, bit-stream formatting, and error protection algorithms [64], and digitally transmitted over the channel. The received bit stream is decoded and error corrected at the server. The resulting frames are then decompressed and processed though the Feature Processing block that performs three operations — a weighted log-energy computation, delta and acceleration computations, and voice activity detection. The output frames from this block are 13-dimensional (weighted energy plus 12 cepstral coefficients). These 13-dimensional feature vectors are differentiated to generate coefficients representing the first and second-order derivatives. The novel individual components of this front end are discussed in the following sections.

2.4.1. Noise Reduction

The noise reduction process consists of a two-stage time domain mel-warped Wiener filtering process [63,64] that uses a frame-based noise reduction approach. The first stage is a mel-warped classical Wiener filter that reduces noise but introduces a white residual noise [65]. This white residual noise is removed using a second stage of the mel-warped classical Wiener filtering.

The frequency responses of the two Wiener filters are derived using estimates of the noise and speech spectra. In the first stage, the noise spectrum is estimated using only

the non-speech frames. A log energy-based voice activity detector (VADNest) [63] is used to detect the speech frames. The spectrum of the clean speech signal is estimated by subtracting the estimate of the noise spectrum from the estimate of the spectrum of the input signal. However, in the second stage, the noise spectrum is estimated every frame (speech and non-speech). The estimate of the clean speech from the first stage is improved by applying the first-stage Wiener filter. This improved estimate of the clean speech, along with the estimate of the noise spectrum, is used for computation of the second-stage Wiener filter. The frequency response of the two Wiener filters are smoothed and time-warped using a filter bank that incorporates 23 mel-scale bins. The impulse response of these filters is then computed by applying an inverse DCT transform to the frequency response.

The second stage uses an additional gain factorization stage that operates on the noise-reduced signal at the input of this stage to accomplish a dynamic noise reduction. Frames are classified as speech or non-speech based on the SNR. For speech frames, the gain of the Wiener filter's frequency response is set to 0.1 whereas for the non-speech frames, the gain is set to 0.8. The overall influence of this processing is that more aggressive noise reduction is applied to non-speech frames than speech frames.

2.4.2. Waveform Processing

The waveform processing block improves the SNR of the signal by emphasizing voiced speech segments and deemphasizing the non-speech segments of the signal. A smoothed instant energy contour computed though a Teagor energy operator [66] is used

to detect the speech and non-speech segments. The voiced segments of speech signal display quasi-periodic maxima and minima [67]. The smoothed instant energy value corresponding to the voiced segments exhibit a quasi-periodic property and have a period corresponding to the fundamental frequency. The contour corresponding to the unvoiced and silence/noise segments is relatively flat or random. The maxima in the energy contour correspond to the high SNR portions of the signal and hence, are classified as speech segments. The speech segments are then given more weight than the non-speech segments.

2.4.3. Blind Equalization

Blind equalization is applied to reduce the influence of convolutional noise. It is known that the response of the Wiener filter compensates for variations in the channel/ microphone response [68]. The Wiener filter accomplishes this deconvolution by reducing the mean square error between the reference and the recovered signal. In the cepstral domain, it has been shown that the adaptive filter that minimizes the mean square error between the current (recovered) cepstrum and a reference cepstrum [69] compensates for convolutional noise.

2.4.4. Feature Processing

The block named Feature Processing has three main functions — a weighted log-energy computation, delta and acceleration computations, and voice activity detection. The weighted energy is computed using the weighted sum of the zeroth cepstral coefficient and a logarithm of the absolute energy:

$$E_w[n] = 0.6\frac{(c[0])}{23} + 0.4(\ln E[n]) \ . \quad (17)$$

The first and second derivatives of these coefficients are computed and appended to the 13-dimensional base features (weighted energy plus 12 cepstral coefficients) using techniques previously described in section 2.1.9. The non-speech frames are dropped using a two-stage voice activity detector (VAD). In the first stage, three measures of speech activity are computed. Each measure generates a binary decision whether the input frame is speech or non-speech. In the second stage, a heuristic VAD logic combines these three complementary decisions to make a final decision.

The first measure is an acceleration (second derivative) of the energy that is computed across the entire spectrum. The energy is computed by summing the square of the coefficients of the mel-warped Wiener filter corresponding to the first stage of the Noise Reduction block. This filter is described in section 2.4.1. A thresholding mechanism based on the acceleration of the energy is used to make a binary decision. Because this decision is based on the entire spectrum, this measure accurately detects plosive and unvoiced sounds.

The second measure is an acceleration of an energy-based measure that is measured over a group of sub-bands of the spectrum likely to contain the fundamental frequency. An energy-based measure is computed by averaging the coefficients of the first stage mel-warped Wiener filter corresponding to the second, third and fourth bins. Similar to the first measure, a thresholding mechanism is used to make a binary decision. The advantage of this measure is that the high SNR in these three sub-bands makes it highly

robust to noise. On the other hand, it is susceptible to microphone characteristics (low-pass), speaker characteristics and band-pass noise that can significantly alter the energy content of these three sub-bands.

The third measure uses the acceleration of the variance of the Wiener filter [63] coefficients computed over the lower half of the frequency band. Note that the Wiener filter coefficients for this computation are selected before they are mel-warped. Similar to the first and second measures, a thresholding mechanism is used to make a binary decision. This measure accurately detects voiced sounds because it is computed using the portion of the spectrum (e.g., the lower half of the spectrum) that is likely to contain most of the harmonics of the fundamental frequency.

In this chapter, we reviewed the signal modelling techniques employed in three front ends: WI007, MFA and QIO. We also presented a comparison between the WI007 front end and an industry-standard MFCC front end. In the next chapter, we describe the experimental framework used to evaluate these front ends.

CHAPTER III

EXPERIMENTAL DESIGN

This chapter presents the design and development of the WSJ0-derived Aurora-4 database used to evaluate these advanced front ends. The construction of the lexicon and language models is also discussed. The final selection of the Aurora-4 database was a balance between the need to train on large amounts of data and the desire for participants to be able to run experiments quickly. This chapter reviews the experiments that guided the selection of the final subset of the WSJ0 data.

3.1. Corpus Design

The first step in the ALV evaluation was to define an evaluation paradigm. The Aurora Working Group decided to build on a standard evaluation paradigm based on the DARPA Wall Street Journal Corpus (WSJ) [28], and to evaluate noise conditions by postprocessing the clean data using digitally added noise [70]. WSJ is a large vocabulary continuous speech recognition corpus consisting of high-quality recordings of read speech. Two-channel recordings of the same utterances were made at 16 kHz. The first channel consisted of the same microphone for all speakers — a Sennheiser HMD-414 close-talking microphone that was extremely popular at the time. The second channel included a sampling of 18 different types of microphones. The text material for this corpus

was drawn from newspaper articles appearing in the Wall Street Journal. A portion of the data included utterances containing verbalized punctuation (“John COMMA who came home early COMMA decided to read the newspaper PERIOD”).

The data is divided into a sequence of training (train), development (dev test) and evaluation (eval) sets. Further, the Aurora Working Group decided to focus on the 5,000 word evaluation task, popularly known as WSJ0. This is an interesting task in that the evaluation set is defined in such a way that a 5,000 word vocabulary, which is distributed with the corpus, is sufficient to give complete coverage of the evaluation set. This means there are no out of vocabulary words (OOVs) in the evaluation set. This task is often referred to as the 5k closed vocabulary task. It is a popular approach when one wants to focus on acoustic modeling problems, and remove language modeling issues from the evaluation.

A standard bigram backoff language model (LM) [71] is also distributed with the corpus as a reference language model. It consists of 826,002 bigrams and 4,988 unigrams with corresponding backoff weights. This bigram language model yields a perplexity [43] of 147.

The standard training set for the WSJ0 is defined as SI-84. This set contains 7,138 utterances from 83 speakers, totaling 14 hours of speech data. The SI-84 set contains a mixture of utterances with and without verbalized punctuation. A typographic error in the training transcriptions, “EXISITING” instead of “EXISTING”, was fixed. While the dev test set consists 1206 utterances from 10 speakers, the eval set is defined by the November 92 NIST evaluation set [72] consisting of 330 utterances from 8 speakers.

For the ALV evaluation, processed versions [73] of the training, dev test, and evaluation utterances were generated at 8 kHz and 16 kHz. G.712 filtering [24] was used to simulate the frequency characteristics at an 8 kHz sample frequency and P.341 filtering [74] was used for simulation at 16 kHz. The filtering was applied to the noisy data as well. As shown in Figure 15, Training Set 1 consisted of the filtered version of the complete SI-84 training set (7138 utterances) recorded with the Sennheiser microphone.

Training Set 2 was used to study the effects of variation in microphone and noise. Its data distribution is also shown in Figure 15. The filtered 7,138 training utterances are divided into two blocks: 3569 utterances (half) recorded with the Sennheiser microphone, and the remaining half recorded with a different microphone (18 different microphone types were used). No noise is added to one-fourth (893 utterances) of each of these subsets. To the remaining three-fourths (2,676 utterances) of each of these subsets, 6 different noise types (car, babble, restaurant, street, airport, and train) were added at randomly selected SNRs between 10 and 20 dB. The goal was to attain an equal distribution of noise types and SNRs. Thus, we had one clean set (893 utterances) and 6 noisy subsets (446 utterances each) for both the microphone conditions.

There is one irregularity in Training Set 2. The speech file 408o303.wv1, recorded with the Sennheiser microphone exists, but the file 408o303.wv2, recorded with a second microphone, did not exist on the original WSJ0 CDs. To keep the number of files constant across both the training sets, the file 408o302.wv2 was selected instead of 408o303.wv2. Thus both files 408o302.wv1 and 408o302.wv2 were used in Training Set 2.

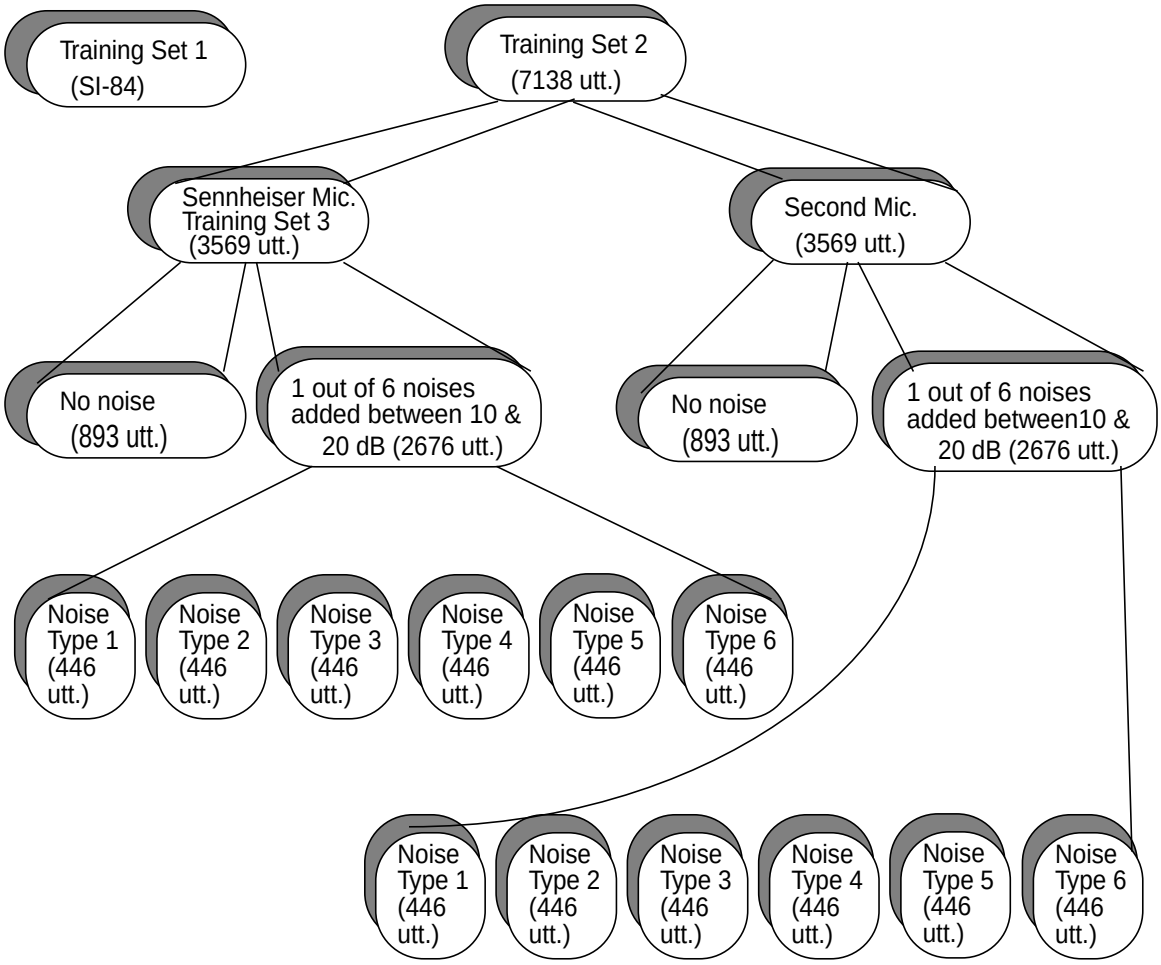

Figure 15. Definition of Training Set 1 (Clean Training) and Training Set 2 (Multi-condition Training).

Training Set 3 was defined to study the impact of using utterances recorded only with the Sennheiser microphone for training. The Sennheiser microphone block of the Training Set 2 was referred to as Training Set 3. We will see chapter 4 that the results on this set were poor because of the reduction in the number of training utterances. Consequently, this training set was not considered for further experimentation in the ALV evaluation.

Fourteen evaluation sets were defined in order to study the degradations in speech recognition performance due to microphone conditions, filtering and noisy environments. Each of the filtered versions of the evaluation set recorded with the Sennheiser microphone and the secondary microphone were selected to form two evaluation sets. The remaining 12 subsets were defined by randomly adding each of the 6 noise types at randomly chosen SNRs between 5 and 15 dB for each of the microphone types as shown in Figure 16. The goal was to have an equal distribution of each of the 6 noise types and the SNR with an average SNR of 10 dB. Following the same process that was used for the

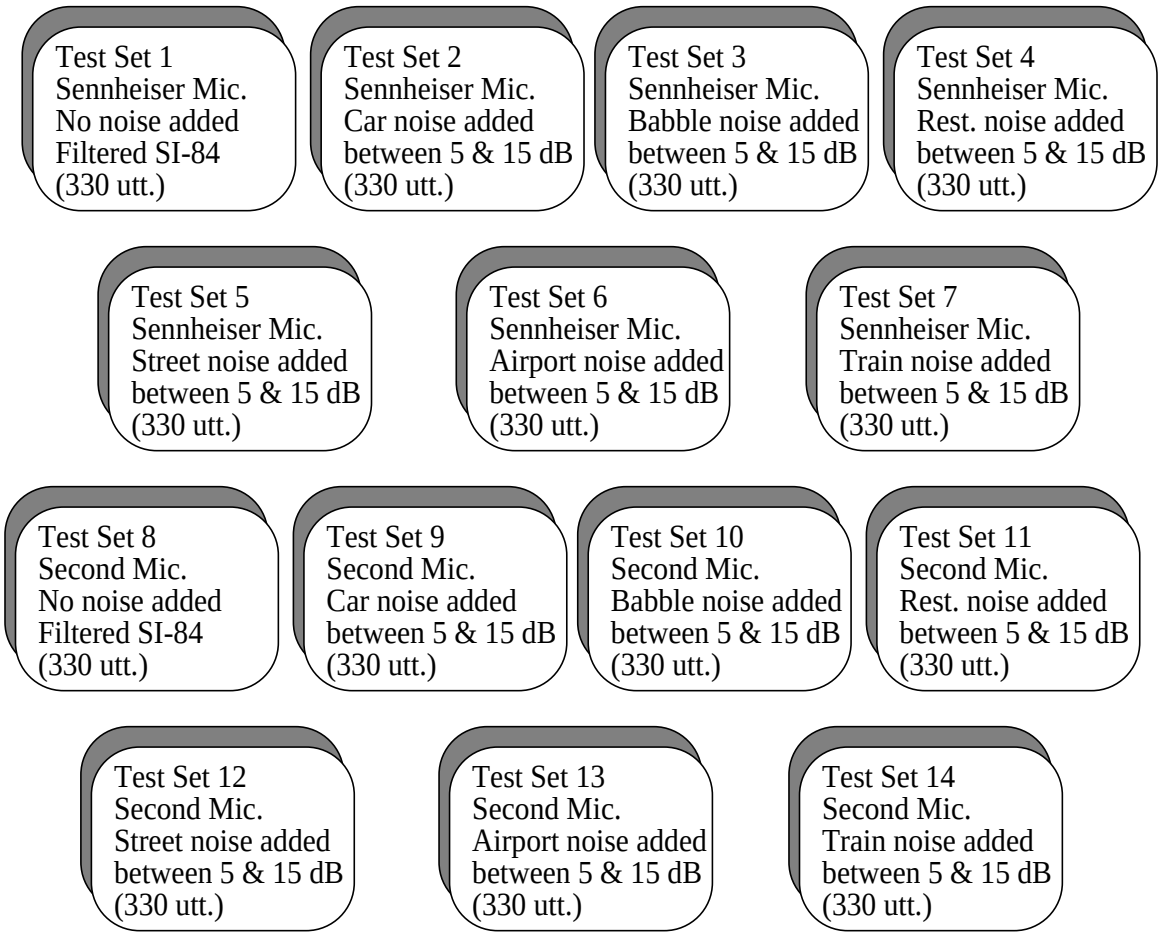

Figure 16. Definitions of 14 Test Sets that include 6 noise types and different mic types.

definition of the 14 evaluation sets, 14 dev test sets, each consisting of 1206 utterances, were also created to allow for future research.

3.2. Language Model and Lexicon

The pronunciations contained in the lexicon were prepared using the publicly available CMU dictionary (v0.6) [75] with some local additions made to give full coverage of the training set. The additions needed for the training lexicon are shown in Table 4. All stress markers in the CMU dictionary were removed and the words "!SENT_START" and "!SENT_END" were added to follow the ISIP prototype system lexicon format. Each pronunciation was replicated twice in the lexicon (one ending with the sil phoneme and one with sp) to model both long and short inter-word silences (a requirement for the technology being used in the baseline system). Similarly, an evaluation lexicon was prepared from the CMU dictionary with local additions as shown in Table 5.

Table 4. Local additions to the CMU lexicon needed for coverage of the SI-84 training set.

Word	Pronunciation
PHILIPPINES	F IH L IH P IY N Z
PHILIPS	F IH L AH P S
PURCHASING	P ER CH AH S IH NG
ROUTE	R AW T R UW T
ROUTINE	R UW T IY N
ROVER	R OW V ER

Table 5. Local additions to the CMU lexicon needed for coverage of the November 92 eval set.

Word	Pronunciation
PURCHASING	P ER CH AH S IH NG
ROUTES	R AW T S R UW T S
ROUTINELY	R UW T IY N L IY
ROVING	R OW V IH NG

The 5K bigram LM and associated lexicon do not give complete coverage of the dev test set. Since our goal was to conduct all experiments with no OOVs, we decided to augment the LM with the missing words. There are several ways this can be done. We chose a static linear interpolation technique supported in the SRI Language Modeling Toolkit (SRILM) [76]. We constructed an interpolated bigram LM by generating an LM on the test set, and interpolating it with the existing bigram such that the overall perplexity of the modified LM was comparable to the original LM. The original LM had a perplexity of 147. The interpolated LM was constructed by setting the interpolation factor such that the final perplexity was the same. The resulting value of this interpolation factor was 0.998. This interpolated LM was only used for tuning experiments on the dev test set.

3.3. Aurora-4 Database Development and Definitions

LVCSR experiments are computationally expensive and require a fairly large amount of infrastructure. Most of the sites participating in ALV evaluation did not have such a large infrastructure but they wanted a rapid turnover of experiments while developing their front ends. Hence, a goal was established to define a small subset that would provide results comparable to a full evaluation and yet run in a single day's worth of CPU time on an 800 MHz Intel CPU.

Though such short sets are notoriously misleading, it was considered a priority to provide such sets to the working group. Below we describe the development of various short sets, and other modifications made to the standard evaluation data set to meet the needs of the Aurora evaluation. For the ALV evaluation, only training set and eval set definitions were required and hence, these sets were defined as Aurora-4a database. The

development-test set definitions were later included in the Aurora-4 database as Aurora-4b for future experimental purposes. We will show that these reduced sets represented a good compromise between computation time and the integrity of the experimental results.

3.3.1. Training Subset Selection

The WSJ0 SI-84 training set consists of 7,138 utterances, 83 speakers and over 14 hours of data. There are more than 129,000 word tokens and about 10,000 unique words. The average number of words per utterance is 17.8, and the average utterance duration is 7.6 secs. The average speaking rate is about 2.4 words per second. The training set includes utterances with verbalized punctuation. The distribution of the number of words per utterance for the entire training set is shown in Figure 17. Figure 18 summarizes the distribution of the utterance durations.

One major design decision in the construction of the short set was to preserve all 83 speakers since we are concentrating on speaker independent recognition. A major constraint for the evaluation was that a complete experiment should be able to be run in one day using a single 800 MHz Pentium cpu. We decided to select 415 training utterances and 30 dev test set utterances to meet this constraint. Since training on the full SI-84 set up to 16 mixture cross-word models requires about 10 days (275 hours), the training time required for 415 utterances was 275 x 415/7138 = 16 hours. Similarly, the decoding time of 50 hours for 330 utterances was approximately 50 x 330/30 = 5 hours for 30 utterances. These compromises reduced the compute time for one complete experiment to approximately one day.

This, in turn, motivated a second major design decision: uniformly sample each speaker, resulting in 5 utterances per speaker. Since the average number of words per utterance was 18, we decided to throw out utterances that were extremely short (less than eight words) and long (greater than 24 words) with respect to the average utterance length. This reduced SI-84 to 4,944 utterances. We then randomly sampled the remaining utterances from each speaker to obtain a total of 415 utterances (83 speakers x 5 utterances per speaker). We will refer to this set as short-415 [77]. Its word count and duration statistics are compared to the full training set in Figures 19 and 20, respectively. Both distributions for the short-415 set are peaky compared to the distributions for the SI-84 training set because extremely short and long utterances with respect to average length were not included in the short-415 set.

Unfortunately, performance of system trained on 415 utterances even for the (1-mixture cross-word models) was poor — 44% WER as shown in Table 6. This system was tested on a short development set consisting of 30 utterances. Hence, we followed a similar paradigm but doubled the training set size to 830 utterances (short-830). The performance on short-830 for 1-mixture cross-word models was also poor — 36.0% WER. We then decided to increase the training set size to a quarter of the total training utterances in SI-84. This yielded short-1785 with an error rate of 25.5%.

Table 6. Performance as a function of the training set for the baseline system (with ISIP's standard front end and unfiltered audio data).

Acoustic Models	Training Set	Devtest-30
CI-Mono-1-mix	415	46.0%
CD-Tri-1-mix	415	44.1%
CI-Mono-1-mix	830	46.6%
CD-Tri-1-mix	830	36.0%
CD-Tri-1-mix	1785	25.5%

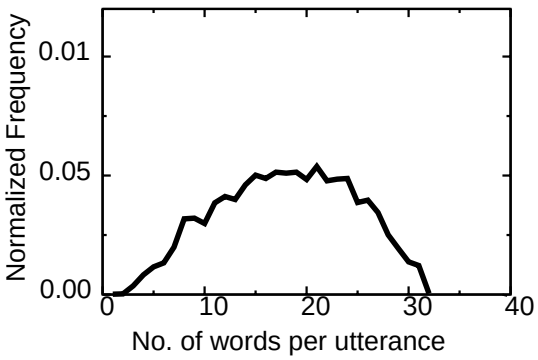

Figure 17. A histogram of the word counts for the full training set (SI-84).

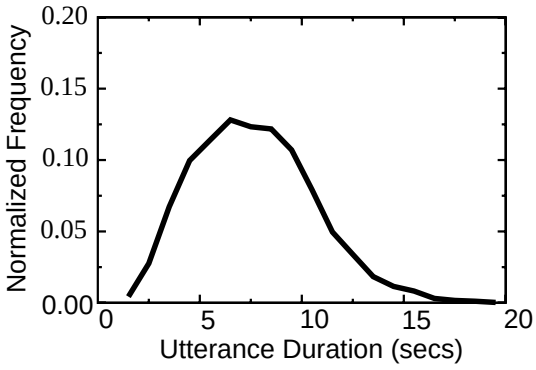

Figure 18. A histogram of the utterance durations for the full training set (SI-84).

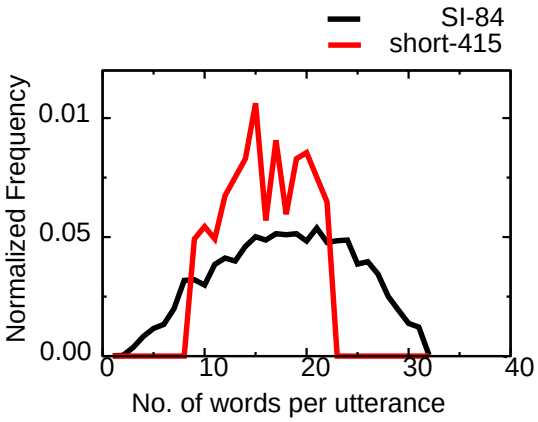

Figure 19. Comparison of the histograms of the word counts for the full training set (SI-84) and the short-415 training set.

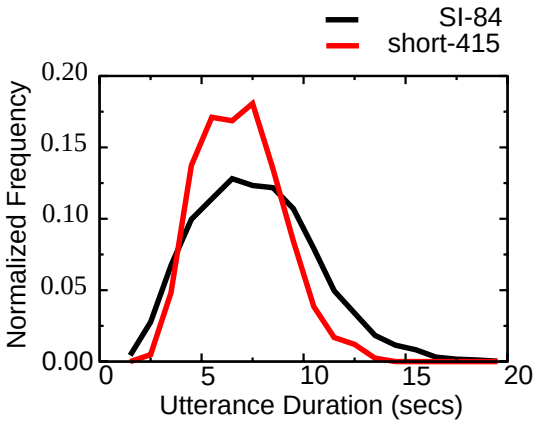

Figure 20. Comparison of the histograms of the utterance durations for SI-84 and short-415.

Since the training subset has to be consistent for both the Training Set 1 (clean) and Training Set 2 (multi condition data), we decided to sample one-fourth of each of the clean and noisy utterances from both the Sennheiser as well as the second microphone conditions. The distribution of data in Training Set 2 is shown in Figure 15. We alternately picked 112 utterances from each of the 12 noisy blocks and 224 utterances from each of the two clean blocks to obtain 1,792 utterances. We refer to this set as short-1792. This is

what was used for acoustic training in the evaluations. A summary of the word count and duration statistics are shown in Figures 21 and 22, respectively. Note that the word count distribution as well as the utterance duration distribution for the short-1792 is very similar to the respective distributions for the SI-84 training set.

Key statistics for all short training sets are provided in Table 7. Although the average duration and speaking rate is almost constant across all the training sets, the total number of unique words drastically reduce as the number of utterances decreases. This often results in undertrained acoustic models since there are an insufficient number of

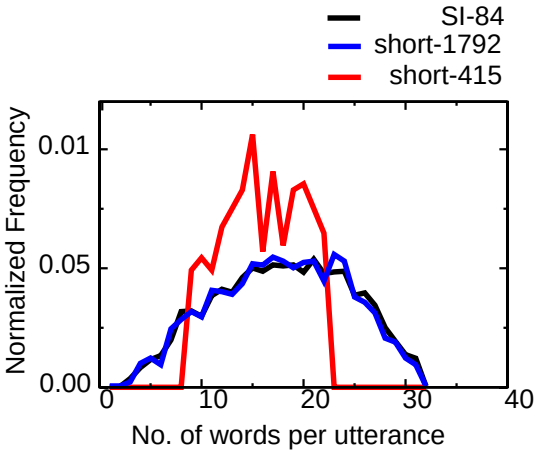

Figure 21. Comparison of the histograms of the word counts for SI-84 and short-1792.

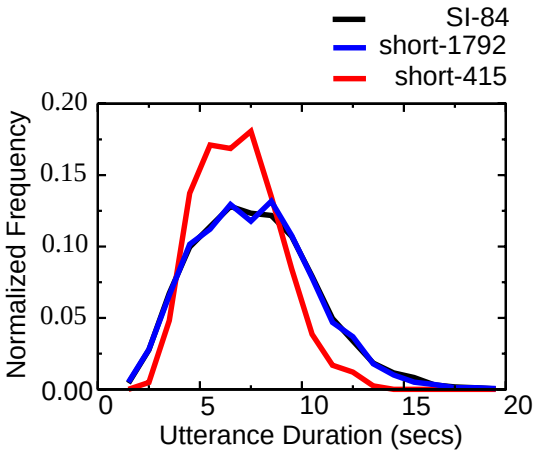

Figure 22. Comparison of the histograms of the utterance durations for SI-84 and short-1792.

Table 7. A comparison of some vital statistics for various training subsets.

Training Set Size	Total Number of Words	Average No. Words/ Utterance	Number of Unique Words	Average Duration (secs)	Average Speaking Rate (words/sec)
415	6,797	16.4	2,242	6.85	2.4
830	14,996	18.1	3,626	7.67	2.4
1785	32,085	18.0	5,481	7.59	2.4
1792	32,012	17.9	5,444	7.63	2.4
SI-84	128,294	18.0	8,914	7.62	2.4

instances of each phonetic context to support reliable training. Hence, all 7,138 utterances were included in each of the two training sets (Training Set 1 and Training Set 2) defined in the Aurora-4a database [78].

3.3.2. Devtest Subset Selection

The Nov'92 development test set consisted of 1,206 utterances, and included 10 unique speakers, and totals over 134 minutes of data. Similarly, the Nov'92 evaluation set consists of 330 utterances, 8 speakers, and about 40 minutes of data. Our goal was to produce a short set that was a reasonable match to the statistics of both of these sets. Following the same strategy described in section 3.3.1, we selected 3 utterances per speaker for a total of 30 utterances.

Due to time constraints and the large number of experiments that needed to be run to effectively tune a system, we decided to reduce the 1206 utterance dev test set to a 330 utterance set which was comparable in size to the evaluation set. To do this, we decided to preserve all 10 speakers represented in the dev test, and select 33 utterances per speaker. These utterances were selected such that the duration profile of the 330 utterance subset was a good model of the entire 1206 utterance set (measured in words per utterance). The 14 noisy subsets corresponding to these 330 utterances were defined. Each of these 14 subsets corresponds to the 14 noisy sets defined in section 3.1, and these 14 subsets are collectively defined as Aurora-4b database [78].

In the previous paragraph, we described the definition of devtest-330 by sampling the 1206 utterance Nov'92 dev test set. We decided to create devtest-30, a subset of

devtest-330. More importantly, we decided not to throw out long or short utterances this time, because we wanted this subset to be representative of the devtest-330. Hence, we attempted to sample the entire distribution. In Figures 23 and 24, we compare the word counts and duration statistics for these short sets to the full Nov'92 dev test set.

In Table 8, we analyze the statistics of these three sets. Most of the important statistics such as the number of speakers, average utterance duration and average speaking

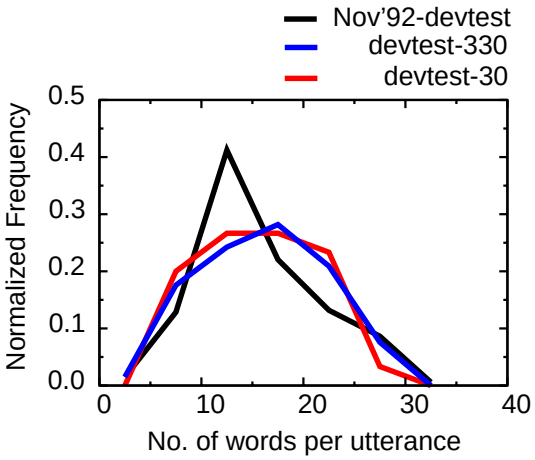

Figure 23. Comparison of the histograms of the number of words per utterance for the full dev test set and two dev test subsets.

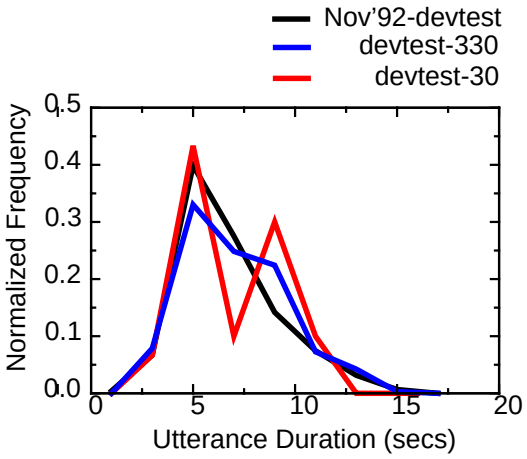

Figure 24. Comparison of the histograms of the utterance durations for the full dev test set and two dev test subsets.

Table 8. A comparison of the complexity of several subsets of the eval and devtest data.

Description	Nov'92 eval	eval-166	Nov'92 devtest	devtest-330	devtest-30
No. of Speakers	8	8	10	10	10
No. of Utterances	330	166	1206	330	30
Amount of Data (mins.)	40.19	20.69	134.42	38.33	3.35
No. of Word Tokens	5,353	2,715	19,254	5,468	493
No. of Unique Words	1,270	936	2,404	1,444	290
Avg. No. of Wd per Utt.	16.2	16.3	16.0	16.6	16.1
Avg. Utt. Duration (secs)	7.3	7.5	6.7	7.0	6.7
Avg. Spk. Rate (wd/sec)	2.2	2.3	2.4	2.4	2.4
Test Set Perplexity	134.9	139.0	146.8	143.7	151.5

rate for these three sets are comparable. Although the number of unique word tokens reduces as the size of the test set decreases, the perplexities [43] of the dev test sets are comparable.

3.3.3. Eval Subset Selection

In order to reduce the computing requirements for the evaluations, we decided to reduce the size of the evaluation set by 50%. To obtain this shortened version of the evaluation set, we began by sampling in such a way that every speaker in the eval set was represented in the shortened set. This is shown in Table 9. We randomly sampled utterances from each of the speakers. This random sampling process was repeated four times to get four different eval short lists (A, B, C, D). Next, we computed the WER for these short sets based on results for the complete evaluation set for a series of experiments on various noise conditions. These results are shown in Table 10.

We then analyzed each set using a number of statistical distance measures to determine the set that is closest to the original eval set. These results are shown in Table 11. We chose subset “A” since this was closest to the results for the full evaluation set for the normalized statistical measures 3 and 4. The word count and duration statistics are compared to the full evaluation sets in Figures 25 and 26, respectively. Both the word count distribution and utterance duration distribution for eval-166 match closely to the respective distributions for Nov’92-eval set. Fourteen noisy versions corresponding to this 166 utterance eval-166 set are defined as the 14 eval sets in the Aurora-4a database [78].

Table 9. Distribution of the number of utterances for each speaker in the eval set.

Speaker Identity	Number of Utterances (full eval)	Number of Utterances (eval-166)
440	40	20
441	42	21
442	42	21
443	40	20
444	41	21
445	42	21
446	40	20
447	43	22
Total	330	166

Table 10. WER on various noisy conditions for the complete November 92 eval set and its four subsets (A, B, C, and D).

Test Set	Training Set	Eval Set	A	B	C	D
1	1	10.1%	10.2%	10.4%	9.7%	9.4%
2	1	55.4%	56.1%	54.9%	56.2%	58.0%
3	1	64.6%	64.8%	63.2%	66.1%	66.9%
4	1	58.4%	59.2%	59.5%	62.0%	58.8%
6	1	61.0%	61.7%	60.4%	63.5%	62.4%
8	1	53.7%	54.6%	52.5%	54.6%	55.5%
9	1	71.6%	72.5%	71.5%	72.8%	73.2%
10	1	76.2%	77.5%	75.5%	77.4%	79.5%
11	1	76.7%	78.5%	74.0%	77.5%	77.5%
13	1	74.5%	77.1%	74.1%	75.8%	76.9%
1	2	27.2%	27.9%	28.0%	28.0%	26.9%

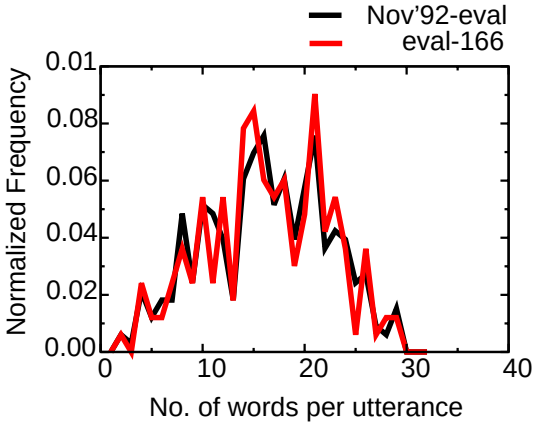

Figure 25. Comparison of the histogram. of number of words per utterance for the full November 92 eval set and eval-166.

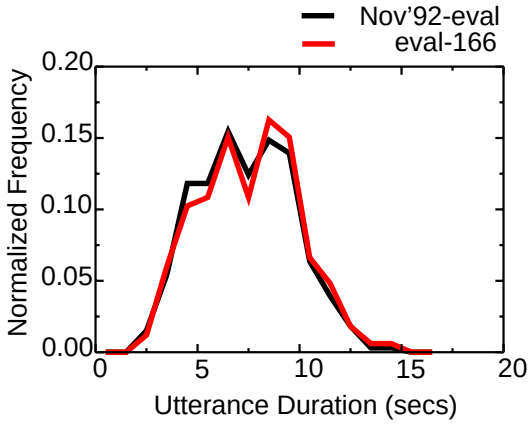

Figure 26. Comparison of the histograms of the utterance durations for the full November 92 eval set and eval-166.

These 14 eval sets are subsets of the fourteen 330 utterance noisy eval sets defined in section 3.1.

3.3.4. Endpointing the Aurora-4a Database

When speech recognition systems are subjected to severe amounts of noise, the non-speech data preceding and following the utterance tends to cause insertion errors. The insertion errors can often be the dominant reason for an increased WER. It was decided to generate an endpointed version of the Aurora-4a database by removing these non-speech segments to evaluate the influence of endpointing on the recognition performance of the baseline ETSI MFCC front end.

To remove the effects of this noise, it was decided to endpoint all speech data, so that complete experiments (training and evaluation) could be performed on endpointed data. We generated endpoint information using our best WSJ0 baseline recognition system, described in section 4 using a forced-alignment mode [79]. The endpoints were

then extended by 200 msec on each side of an utterance. Table 12 summarizes the amount of silence in WSJ data. The first row represents the average number of seconds of audio data removed from each file (amount of data removed per utterance averaged across all utterances). The second row represents the amount of silence removed as a percentage of the total available audio data. The third row represents the total amount of data discarded as a percentage of the total audio data. Note that the eval data tended to have more silence than the training data.

In this chapter, we presented the process for the development of the Aurora-4 database. CPU constraints played a primary role in our decisions to reduce the amount of data. In the next chapter, we will describe the development of a speech recognition system that served as the baseline system for the Aurora evaluations. This system was carefully designed to produce statistically significant results on these evaluations while minimizing the CPU requirements for the evaluation.

Table 11. Results of several distance measures applied to select a subset of the full eval set.

Distance Measure	A	B	C	D
$\sum abs(x - x_i))$	10.70	9.80	15.00	17.60
$\sum (x - x_i)^2$	15.47	13.90	28.92	37.84
$\sum abs\left(\frac{(x - x_i)}{x_i}\right)$	0.18	0.19	0.28	0.33
$\sum \left(\frac{(x - x_i)}{x_i}\right)^2$	0.22	0.23	0.48	0.60

Table 12. A summary of the amount of silence detected in the WSJ Corpus.

Data Set	SI-84	dev-330	eval
Average Silence/ Utt. (secs)	0.82	0.97	1.37
Average Silence/ Utt. (%)	11.2	16.4	25.3
Total Silence (%)	10.8	13.9	18.8

CHAPTER IV

WSJ0 BASELINE SYSTEM

The WSJ0 baseline system was developed as a primary step towards achieving the final goal of developing the ALV baseline system. The WSJ0 baseline system provided a comparison point to insure that the future results on ALV baseline system were credible. This system demonstrated performance that is sufficiently close to start-of-the-art on the WSJ0 task. In this chapter, we describe the WSJ0 baseline system and the results of the tuning experiments on this system.

4.1. System Description

The baseline system to be used for the Aurora evaluations is based on a public domain speech recognition system that has been under development at the Institute for Signal and Information Processing (ISIP) at Mississippi State University for several years. This system is referred to as the prototype system [79] since it was the first recognition system developed in ISIP and served as a test bed for developing ideas for implementing conversational speech recognition systems. This system is implemented entirely in C++ and is fairly modular and easy to modify. It has been used on several evaluations conducted by NIST [80,81] and the Naval Research Laboratory [82].

The prototype system uses hidden Markov model-based context-dependent acoustic models [42], lexical trees [83] for cross-word acoustic modeling, N-gram language models with backoff probabilities [13,42] for language modeling (finite state networks are also supported), and a tree-based lexicon for pronunciation modeling [84]. The core of the system is a hierarchical dynamic programming-based time synchronous network search engine [85,86] that implements a standard beam-pruning approach for maximizing search accuracy while minimizing memory requirements.

The signal processing component of the prototype system is the industry standard MFCC front end described in section 2.1. To adjust to varying channel and speaker conditions, cepstral mean subtraction [61] was performed on the 12 cepstral features with the mean being computed and subtracted separately for each utterance. Other normalization techniques, such as vocal tract length normalization [87] and variance normalization [88], were not used for the WSJ0 baseline system. Further, adaptation techniques, such as Maximum Likelihood Linear Regression (MLLR) [89] and Linear Discriminant Analysis (LDA) [90], were not employed.

Using the feature data, a set of context-dependent cross-word triphone models were trained. Each triphone model was a 3-state left-to-right model with self-loops with the exception of two models as shown in Figure 27. The silence model, *sil*, has a forward and backward skip transition to account for long stretches of silence containing transitory noises. The short, inter-word silence model, *sp*, contains a forward skip transition that allows it to consume no data when there is no silence between consecutive words. Each

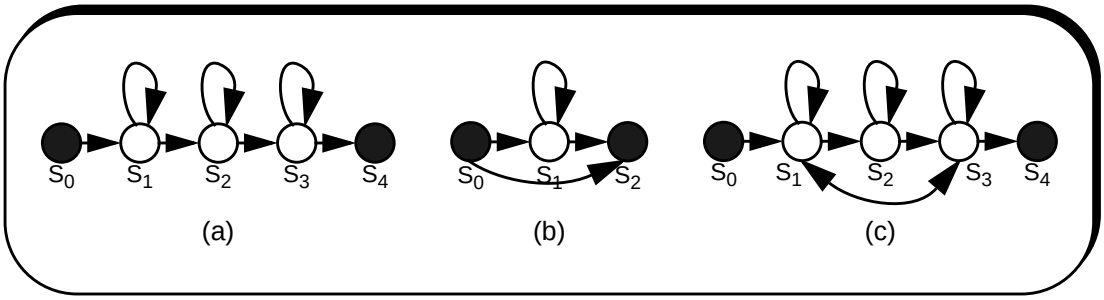

Figure 27. Typical HMM topologies used for acoustic modeling: (a) typical triphone, (b) short pause, and (c) silence. The shaded states denote the start and stop states for each model.

state in the models contains a Gaussian mixture model where the number of mixtures is initially set to one and is trained up to sixteen mixtures.

The triphone models were trained using a standard Baum-Welch Expectation Maximization (EM) training algorithm [12,91,43]. A typical training schedule is summarized in Table 13. However, for the baseline system, we modified the forced alignment step. Instead of aligning the word transcription using monophone models to get the monophone transcriptions, we produced cross-word triphone transcriptions by aligning the word transcription with cross-word triphone models. These triphone models were previously generated from the best performing system tuned on the eval set. These aligned triphone transcriptions were then converted to monophone transcriptions by removing the left and right context for each central phone. Models for all possible triphone contexts were generated using the decision trees produced during the state-tying phase of the training process — one of the distinct advantages of the decision tree based state-tying approach. The trained models were then used in conjunction with a bigram language model to perform recognition on the evaluation data.

Table 13. An overview of the training paradigm for a typical cross-word context-dependent large vocabulary speech recognition system.

1. **Flat-start models:** Initialize a set of single-mixture Gaussian monophone models. Seed the mean and variance of each Gaussian to be equal to the global mean and variance computed across a small set of the training data. This provides a reasonable starting point for the model optimization. Random initialization would also work but would converge less quickly.
2. **Monophone training:** Train monophone models on the entire training set using four iterations of Baum-Welch reestimation. In this phase, the 'sp' model is not trained and it is assumed that 'sil' only occurs at the beginnings and ends of utterances. This gives the 'sil' model a chance to learn the parameters of silence before we attempt to force it to learn interword silence.
3. **'sp' model training:** The single state of the 'sp' model is tied to the central state of the 'sil' model. The monophone models are then trained for four more iterations. In this phase, it is assumed that the 'sp' model occurs between every pair of sequential words while 'sil' only occurs at the beginnings and ends of utterances. This allows the 'sp' model to learn the parameters of interword silence.
4. **Forced alignment:** The transcriptions are force aligned to the acoustic data and the aligner is allowed to choose the most likely pronunciation for each word in the transcription. New phonetic transcriptions are generated from this forced alignment process and are used throughout the remainder of the training regime.
5. **Final monophone training:** The monophone models are trained using the new phonetic transcriptions and five iterations of Baum-Welch reestimation.
6. **Cross-word triphone training:** Cross-word triphone models are seeded from the monophone models. Only triphones seen in the training data are created. Four iterations of Baum-Welch reestimation are used to get initial estimates of the triphone models.
7. **State-tying:** To reduce the parameter count and to provide sufficient training data to undertrained states, we employ a maximum likelihood decision tree-based state tying procedure [93]. Those states that are statistically similar to one another are tied into a single state and the training data previously attributed to each is now shared in the single tied state. The state-tied models are trained for four more iterations of Baum-Welch reestimation.
8. **Mixture training:** The single mixture models are successively split until 16 mixtures are generated using incremental stages of 1, 2, 4, 8 and 16 mixtures. At each stage, four iterations of Baum-Welch reestimation are used on the multi-mixture models.

4.2. WSJ0 Baseline System Tuning Experiments

Most HMM-based recognition systems provide a set of parameters that can be used to tune performance for a given application. These parameters include thresholds for state-tying and beam pruning, and scaling factors for the language model and word insertions. The first parameter we tuned is the state-tying threshold [92,93]. A problem often associated with training context-dependent models in a speech recognition system is the lack of sufficient training data for the large number of free parameters in the system. To avoid this problem the prototype system employs a maximum likelihood phonetic decision tree-based state-tying procedure [93] to pool HMM states. Each node of the phonetic decision tree is associated with a set of states. These states are iteratively separated into child nodes using phonetic questions. When the tree is trained, the states in a single leaf node of the tree represent acoustically similar states that can be tied together. This leads to better parameter estimates. The parameters governing the state-tying process are the thresholds for splitting and merging a node [93].

All the experiments for baseline system were conducted on SI-84 training set and Nov'92 eval set. Table 14 shows the performance improvement due to varying the number of states after tying. Normally this parameter has a much more dramatic effect on performance. In this case, the improvements in performance were marginal. The number of tied states found to give best performance was 3,215. The number of initial states was 46,346, which implies that less than one out of every 10 states were preserved in the final models.

The second parameter we optimized is the language model scaling factor. During the decoding process, the language model probability (as determined by the bigram language model) is computed for each bigram pair. This probability is multiplied by a language model scale factor that weights the relative contribution of the language model to the overall path score. Increasing this scale value tends to cause the language model to dominate the ranking of search paths — essentially boosting the importance of the language model relative to the acoustic model. Decreasing the scale causes the language model to play a lesser role. A word insertion penalty is added to the scaled language model score. This penalty is used to help inhibit the insertion of common, poorly articulated words such as “the”, “a”, and “uh”. Decreasing the value of this parameter will tend to decrease the number of words hypothesized. For the experiments presented in Table 14, the language model scale factor [92] was set to 12 and the word insertion penalty [92] set to -10.

Table 14. A comparison of experimental results obtained by tuning the number of tied states retained after the state-tying process (language model scale = 12.0, word insertion penalty = -10 and pruning thresholds set to 300, 250 and 250).

Number of Tied-States	State-Tying Thresholds			xRT	WER	Sub.	Del.	Ins.
	Split	Merge	Occup.					
1,157	1250	1250	2400	171	9.4%	7.1%	1.6%	0.7%
1,882	650	650	1400	151	11.0%	8.0%	1.7%	1.2%
3,024	150	150	900	149	10.7%	8.0%	1.6%	1.1%
3,215	165	165	840	138	8.6%	6.8%	1.1%	0.7%
3,580	125	125	750	123	8.9%	6.7%	1.4%	0.8%
3,983	110	110	660	120	8.7%	6.6%	1.0%	1.1%
4,330	100	100	600	116	9.1%	6.5%	1.4%	1.2%
5,371	75	75	450	106	9.0%	6.7%	1.0%	1.3%

In Table 15, we present results from varying the language model scale factor. The first six experiments were run with the number of tied states set to 3,215, the word insertion penalty set to -10, and the pruning thresholds set to 300, 250 and 250. The last two experiments were run with a word insertion penalty of 10, and gave slightly better performance. An LM scale of 18 was chosen because insertions and deletions are balanced in addition to achieving the lowest overall WER. A 0.6% absolute reduction in error rate was observed by adjusting this parameter. This is probably a result of the fact that the language model has some predictive power for the WSJ data (more so than in a conversational speech application), and hence can be relied upon to a greater degree. Tuning the scale factor also reduced xRT by approximately 30%, which is advantageous.

The next parameter to be tuned was the word insertion penalty. An interesting bit of folklore in speech research is that optimal performance is almost always achieved when one balances insertions, and deletions. In Table 16, we summarize some experiments in which we optimized the value of this parameter. These experiments were run with the number of tied states set to 3,215, the LM scale factor set to 16, and the pruning thresholds

Table 15. A comparison of experimental results for tuning the language model scale factor. The best error rate that was achieved was 8.0%.

LM Scale	Word Penalty	xRT	WER	Sub.	Del.	Ins.
12	-10	138	8.6%	6.8%	1.1%	0.7%
14	-10	108	8.2%	6.3%	1.2%	0.7%
16	-10	103	8.0%	6.1%	1.4%	0.6%
18	-10	85	8.1%	6.1%	1.5%	0.5%
18	10	85	8.0%	6.2%	0.9%	0.9%
20	10	85	8.0%	6.2%	1.0%	0.9%

Table 16. A comparison of experimental results for tuning the word insertion penalty.

Word Ins. Penalty	xRT	WER	Sub.	Del.	Ins.
-20	98	8.4%	6.3%	1.7%	0.4%
-10	103	8.0%	6.1%	1.4%	0.6%
0	107	8.1%	6.3%	1.0%	0.7%
10	117	8.2%	6.3%	0.9%	1.0%

set to 300, 250 and 250. Though the best performance on this isolated experiment was obtained with a setting of -10, a word insertion penalty of 10 was selected because it produced near optimal results and balanced insertions and deletions. The fact that this was the optimum point was verified when these results were combined with other parameter settings.

Once these basic parameters were adjusted, we turned our attention to beam pruning [92], which allows users to trade off search errors and real-time performance. Tight beams result in fast decoding times but less accuracy. Beam pruning is a heuristic technique that removes low scoring hypotheses early in the search process so the computational resources associated with those hypotheses can be used for more promising paths. The decoder allows the user to specify a beam at each level in the search hierarchy (typically state, phoneme, and word-level). A higher beam threshold will allow more paths to be considered during the search process. Using too low of a threshold can result in search errors (i.e. where the correct hypothesis is pruned).

A summary of some basic experiments on the impact of the beam pruning thresholds are shown in Table 17. For these experiments, we trained on the SI-84 training

Table 17. A summary of beam pruning experiments on the SI-84 training set and the Nov'92 dev test set.

Beam Pruning			xRT	WER	Sub.	Del.	Ins.
State	Model	Word					
200	150	150	14	8.6%	6.7%	0.9%	1.1%
300	250	250	85	8.0%	6.2%	0.9%	0.9%
400	350	350	230	8.0%	6.2%	0.9%	0.9%

set, and evaluated on the 330-utterance Nov'92 dev test set. As can be seen in Table 17, there is a substantial impact on real-time rates by reducing the beam pruning thresholds. For the WSJ task, we find the combination of 300, 250, and 250 gives near-optimal performance at a reasonable real-time rate. Many other systems implemented in ISIP use these same beam pruning values [86]. Since CPU requirements are an issue in the ALV evaluation due to the large number of experiments needed to be run, it is important to find ways to reduce computations without significantly impacting performance.

In Table 18, we compare the results of our tuned system to state of the art. Our overall best system, as shown in Table 17, achieves a WER of 8.0% on the dev test set, and 8.3% on the evaluation set. The best published results for comparable technology, highlighted in Table 18, are in the range of 6.8% WER. By directly tuning the WSJ0 baseline system on the evaluation set, we have achieved error rates of 7.7%. However, tuning on the evaluation set is not reasonable.

We believe that the primary difference that accounts for the discrepancy in the error rates is the lexicon used by the respective systems. When WSJ research was at its peak, most sites were using proprietary lexicons that had been tuned to optimize performance, normally by implementing some basic form of pronunciation modeling. We

Table 18. A comparison of performance reported in the literature on the WSJ0 SI-84 Nov'92 evaluation task.

Site	Acoustic Model Type	Language Model	Adaptation	WER
ISIP	xwrd/gi	bigram	none	8.3%
CU [94]	wint/gi	bigram	none	8.1%
UT [95]	wint/gd	bigram	none	7.1%
CU [94]	xwrd/gi	bigram	none	6.9%
LT [96]	xwrd/gi	bigram	none	6.8%
CU [94]	xwrd/gd	bigram	none	6.6%
UT [97]	xwrd/gd	bigram	none	6.4%
UT [97]	xwrd/gd	bigram	VTLN	6.2%
LT [98]	xwrd/gi	trigram	none	5.0%
LT [98]	xwrd/gd	trigram	none	4.8%
LT [98]	xwrd/gd/tag	trigram	none	4.4%

do not, however, believe that the lexicon is solely responsible for this large difference (18% relative). Diagnosing the reasons there is a performance gap will take more time since we need to conduct additional experiments which are outside the scope of this work. The difference in performance is a fairly consistent bias that should not mask algorithm differences in the front end processing. Possible reasons for this gap include a difference in the results of the state-tying process, and issues in silence/noise modeling. We have not seen such a large difference with state-of-the-art systems for other tasks we have run (Resource Management and OGI Alphadigits) [1].

4.3. ALV System Design

As described in section 1.2, the goal for the ALV evaluation was to achieve a 25% relative improvement in word error rate (WER) across a variety of noise conditions

compared to the MFCC WI007 front end. Hence, the ALV system was developed to benchmark the advanced front ends (QIO and MFA) relative to the baseline ETSI WI007 front end [15] in a reasonable amount of time. These three front ends have been described extensively in chapter 2 of this work.

The system used in the ALV evaluation was modeled after a 16-mixture WSJ0 system described in the previous section that attained a WER of 8.3%. Training this 16-mixture cross-word context-dependent phone HMM system involves 36 passes of Baum-Welch training. Training on Training Set 1, shown in Figure 15, requires approximately 275 hours, or 10 days, on an 800 MHz Pentium processor. Decoding the 330 utterances that constitute Test Set 1, described in Figure 16, requires about 50 CPU hours. Considering the increase in the decoding time on noisy test sets because of the poor acoustic match between the models and the data to be approximately three times, the total decoding time for 14 test sets was estimate as 150 x 14 hours = 84 days. For the 11 training conditions required for the ALV baseline system, mentioned in Table 19, the total CPU time required would have been 94 x 11 = 1034 days. This computational load was not feasible for most of the sites involved in the Aurora evaluations. Hence, we explored three ways to reduce this time without compromising the integrity of the system or the results:

- **Evaluation set size:** In section 3.3.3 we described the selection process used to reduce the evaluation set from 330 utterances to 166 utterances. This resulted in a 50% reduction in runtime requirements.

- **Number of mixtures:** Mixture generation and training is another time consuming process, since it involves multiple passes through the data. For example, reducing the number of mixtures from 16 to 4 would reduce the

Table 19. Training conditions that were evaluated for the ALV baseline system.

Training Conditions	Compression	Training Set	Sampling Frequency	Utterance Detection
1	no	1	16 kHz	No
2			16 kHz	Yes
3			8 kHz	Yes
4		2	16 kHz	No
5			16 kHz	Yes
6			8 kHz	Yes
7		3	16 kHz	No
8	yes	1	16 kHz	Yes
9			8 kHz	Yes
10		2	16 kHz	Yes
11			8 kHz	Yes

number of training passes from 36 to 28. Hence, the computation time during training by a factor of 7/9, and result in minimal degradations in performance. An analysis of performance as a function of the number of mixtures is given in Table 20. We decided to select 4 mixtures for the final baseline system.

- **Beam pruning:** Decreasing beam widths in the search process is a straightforward way to reduce computational complexity. In Table 17, we evaluated performance as a function of a selected number of combinations of beam pruning parameters. We selected the settings “200 150 150” because it was observed that these settings reduced runtime by a factor of 6 with minimal degradations in performance.

Table 20. A summary of performance on the Nov’92 dev test set using the SI-84 training set as a function of the number of mixtures.

Number of Mixtures	xRT	WER
2	115	11.8%
4	113	9.5%
8	116	8.7%
16	114	8.0%

Hence, after incorporating these optimizations, we were able to reduce the expected total computation time required to generate Table 19 from 1,034 days to 163 days. The impact of these changes on performance is summarized below in Table 21.

This chapter described the design and development of a baseline LVCSR system which was used in the ALV evaluations. We also presented results on how this system was tuned to improve its speed and to allow rapid evaluation of advanced front ends in a reasonable amount of time. In the next chapter, we will present the results and analysis of the baseline MFCC (ETSI WI007) and two advanced front ends (QIO, and MFA) that were included in the ALV evaluation. The next chapter also describes the front-end specific tuning experiments that were designed to evaluate the influence of the sub-optimal parameter tuning on the performance of the advanced front ends.

Table 21. Relative degradation in WER due to the three-step approach used to reduce computational requirements.

Factor	WER	Relative Degradation
WSJ0 Baseline system (ISIP front end)	8.3%	N/A
Terminal filtering (ISIP front end)	8.4%	1%
ETSI WI007 front end	9.6%	14%
Beam adjustments (15 xRT)	11.8%	23%
Reduce 16 to 4 mixtures	14.1%	20%
50% reduction of eval set	14.9%	6%
Endpointing silences	14.0%	-6%

CHAPTER V

EXPERIMENTS, RESULTS, AND ANALYSIS

The first two chapters provided an overview of the ALV evaluation and a theoretical overview of the front ends included in this study. The third chapter presented the design and development of the Aurora-4 database. In chapter 4, we described the design and development of the baseline system used in the ALV evaluation. In this chapter, we show that the performance of the advanced front ends on the ALV evaluation is significantly better than the baseline MFCC front end, but that these improvements are not operationally significant. It is also shown that front end-specific parameter tuning for the baseline recognition system did not result in a change in ranking of the advanced front ends.

5.1. Performance of the Baseline MFCC Front End

The stated goal for the ALV evaluation was to achieve a 25% relative improvement over the baseline system. This improvement was measured by averaging WER across a variety of evaluation conditions [14], including additive noise, sample frequency reduction, microphone variation, compression, model mismatch and utterance detection. Summaries of this experimentation are provided in Tables 22 and 23. Table 22 contains results for experiments conducted without any feature value compression, and Table 23

provides results with compression. Each row in these tables consists of seven different test conditions: clean data plus six noise conditions. As described in chapter 3, the original audio data for test conditions 1-7 was recorded with a Sennheiser microphone while test conditions 8-14 were recorded using a second microphone that was randomly selected from a set of 18 different microphones. Noise was digitally added to this audio data to simulate operational environments.

The impact of using endpointed speech, described in previous section, was also evaluated as an independent variable. For the “no compression” case, the seven test conditions were then evaluated for several combinations of these conditions, resulting in a total of 98 conditions: 7 noise conditions x 2 microphone types x (3 training conditions for Training Set 1 + 3 conditions for Training Set 2 + 1 condition for Training Set 3). For the “with compression” case, the seven test conditions were then evaluated using only endpointed speech, resulting in a total of 56 conditions: 7 noise conditions x 2 microphone types x (2 training conditions for Training Set 1 + 2 conditions for Training Set 2). Hence, a total of 154 test conditions were evaluated. These tables constitute a total of 4,580 hours (191 days) of CPU time on a 800 MHz Pentium processor. Note that the actual CPU time for 191 days is little higher than the estimated CPU time of 163 days, described in chapter 3. The real time rate for decoding on mismatched conditions was higher than anticipated.

In the following sections, we analyze the results for specific contrastive conditions. All results were generated using the standard NIST scoring software [99], and the NIST MAPPSWE significance test [100], which is included in the scoring software package.

Table 22. A summary of results (in terms of WER) obtained by the ALV baseline system (ETSI MFCC WI007 front end) on Aurora-4a task. Training Set 2 with endpointed data and 16 kHz sampling frequency is the overall best condition.

Performance Summary (Without Compression)															
Training Set			Test Set												
Set	Sam Freq in kHz	Utt Det	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	16	N	14.9	65.2	69.2	63.1	72.3	69.4	73.2	61.3	81.7	82.5	75.4	83.8	81.0	84.1
	16	Y	14.0	56.6	57.2	54.3	60.0	55.7	62.9	52.7	74.3	74.3	67.5	75.6	71.9	74.7
	8	Y	16.2	49.6	62.2	58.7	58.2	61.5	61.7	37.4	59.7	69.8	67.7	72.2	68.3	67.9
2	16	N	23.5	21.9	29.2	34.9	33.7	33.0	35.3	49.3	45.2	49.2	48.8	51.7	49.9	49.0
	16	Y	19.2	22.4	28.5	34.0	34.0	30.0	33.9	45.0	43.9	47.2	46.3	51.2	46.6	50.0
	8	Y	18.4	24.9	37.6	39.3	38.8	38.2	40.4	29.7	37.3	48.3	46.1	50.6	44.9	49.3
3	16	N	20.6	23.2	34.4	40.1	38.2	34.7	41.3	46.8	49.1	53.5	53.4	57.2	53.2	56.1

Table 23. A summary of results for the ALV baseline system with feature value compression. Training Set 2 with endpointed data and 16 kHz sampling frequency is the overall best condition.

Performance Summary (With Compression)															
Training Set			Test Set												
Set	Sam Freq in kHz	Utt Det	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	16	Y	14.5	58.4	58.8	53.8	62.5	56.9	65.5	53.3	75.1	76.3	68.5	77.8	73.5	75.9
	8	Y	15.4	49.4	60.6	59.0	57.4	61.9	62.0	36.6	59.9	71.6	67.8	72.5	70.2	69.5
2	16	Y	19.1	23.4	31.7	35.5	35.3	33.1	36.4	40.9	47.4	50.3	48.9	54.7	49.3	51.8
	8	Y	20.7	26.4	38.6	41.6	43.8	41.1	43.4	30.9	38.7	47.1	50.1	53.6	47.3	50.7

5.1.1. Sample Frequency Reduction

Most telephony applications use a sample frequency of 8 kHz even though state-of-the-art ASR systems use speech data digitized at a sample frequency of 16 kHz. Spectral information above 4 kHz can be exploited to provide modest improvements in performance. For example, the third formant for several speech sounds, such as the consonant “s”, has significant energy above 4 kHz. In state-of-the-art systems, a sample frequency of 16 kHz is often used in conjunction with a Sennheiser close-talking microphone to achieve better performance. Hence, we measured performance at both 8 kHz and 16 kHz to analyze whether trends in recognition performance were consistent at both sample frequencies.

A comparison of performance for Training Sets 1 and 2 is shown in Figure 28 for the “no compression” case. A similar comparison for the compression condition is shown in Figure 29. For Training Set 1, degradations due to a reduction in sampling frequency did not follow any trend. However, for Training Set 2, statistically significant degradations in performance were observed on the Sennheiser microphone conditions (Test Sets 3-7) in both the “no compression” and “compression” cases. The Sennheiser HMD-414 is an expensive close-talking microphone which does a good job of maintaining a relatively flat frequency response from DC to 8 kHz. The spectrogram of a typical utterance (e.g., 441c020b) recorded with the Sennheiser microphone, shown in Figure 30, demonstrates that this microphone preserves high frequency information better than the microphones used for the second channel condition. This observation is supported by Figure 31, which

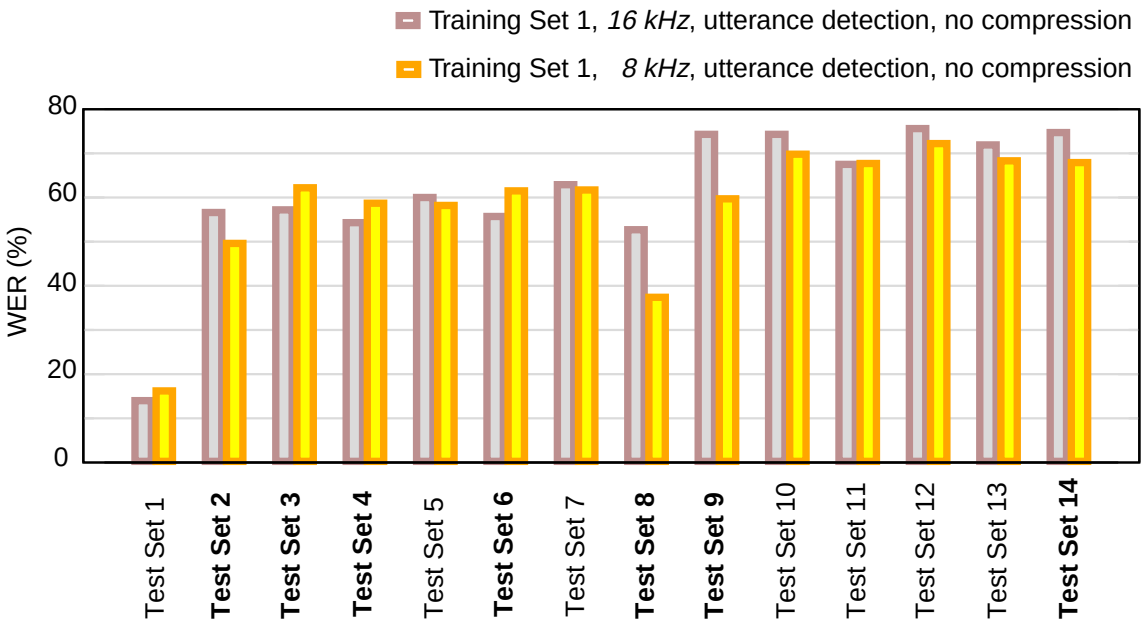

Figure 28(a). A comparison of the WER for *16 kHz* and *8 kHz* sample frequencies for Training Set 1 without feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

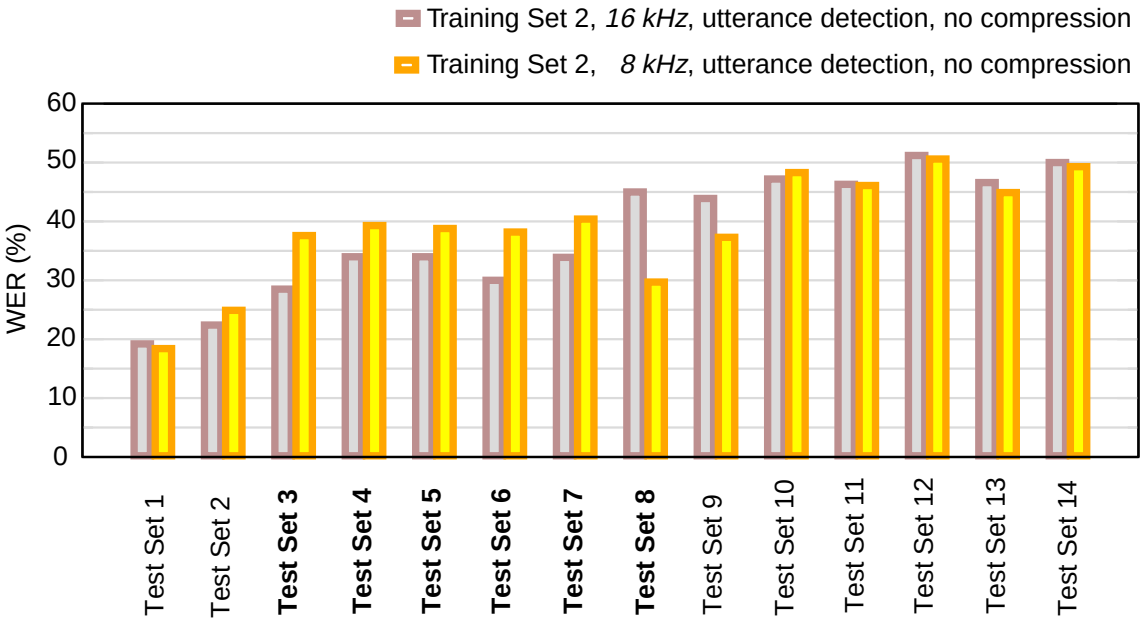

Figure 28(b). A comparison of the WER for *16 kHz* and *8 kHz* sample frequencies for Training Set 2 without feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

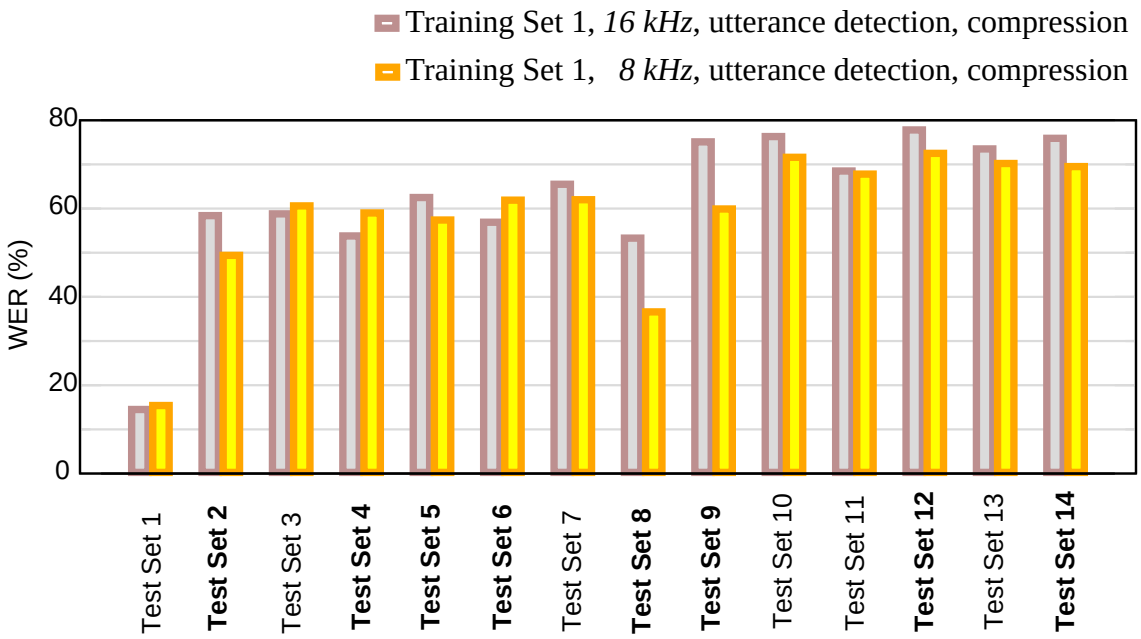

Figure 29(a). A comparison of the WER for *16 kHz* and *8 kHz* sample frequencies for Training Set 1 with feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

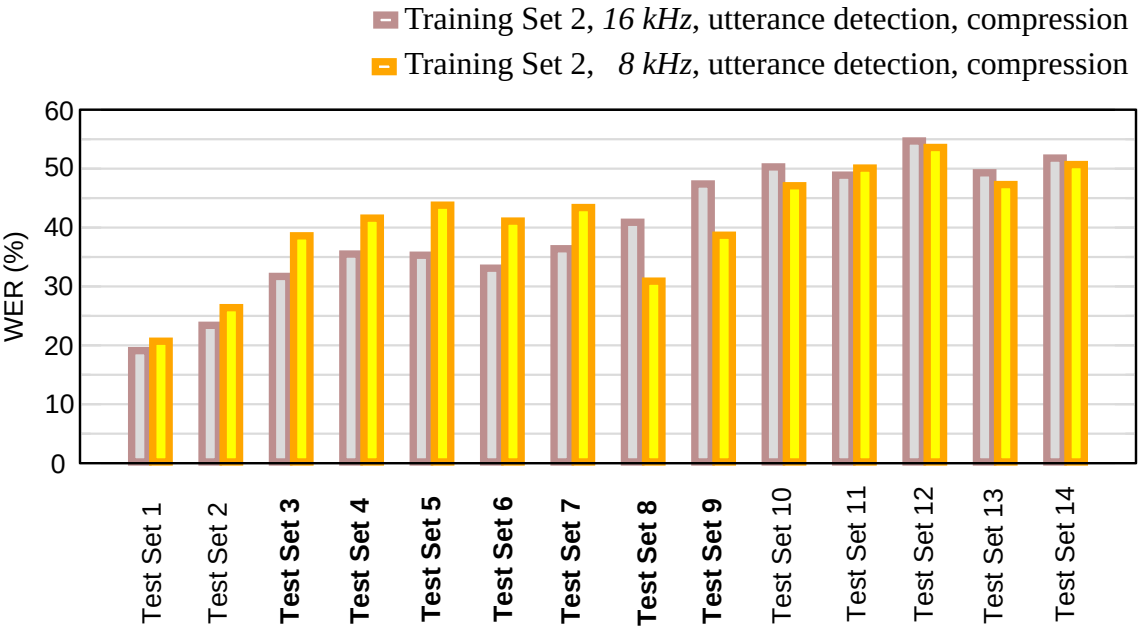

Figure 29(b). Acomparison of the WER for *16 kHz* and *8 kHz* sample frequencies for Training Set 2 with feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

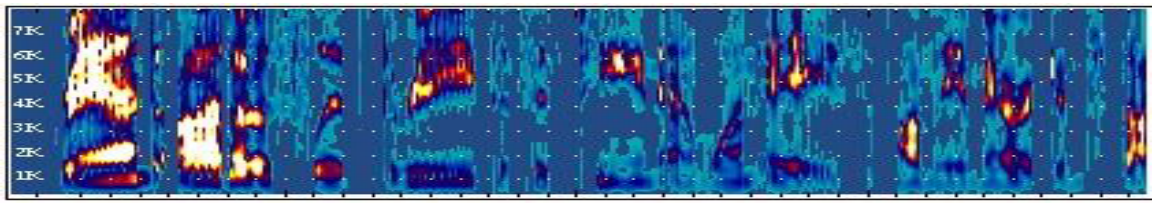

Figure 30(a). Spectrogram for utterance *441c020b* that was recorded on Sennheiser microphone, digitized at 16 kHz and filtered using the ETSI P.341 standard.

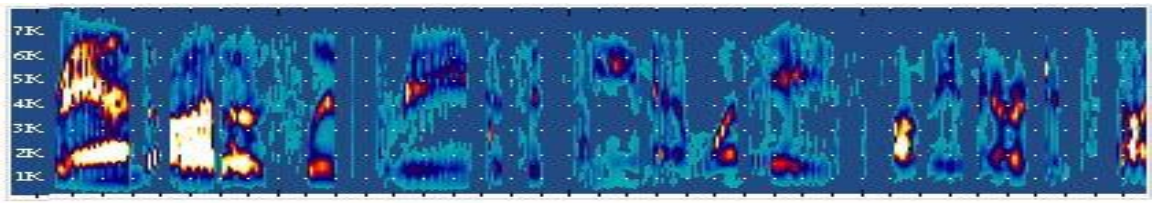

Figure 30(b). Spectrogram for utterance *441c020b* that is recorded on a second microphone, digitized at 16 kHz and filtered using the ETSI P.341 standard.

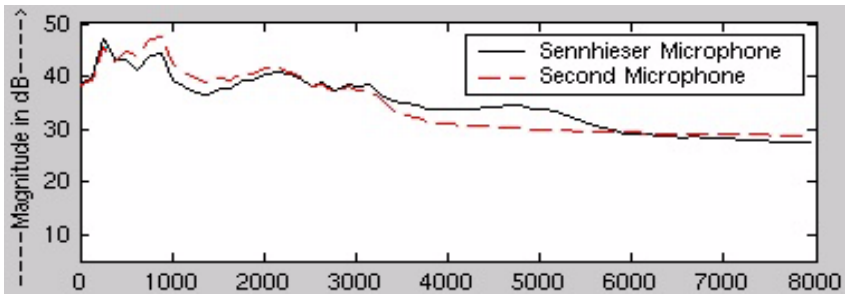

Figure 31(a). Comparison of the magnitude of the frequency response of the Sennheiser microphone and the second microphone derived from the speech segments from the utterance id *441c020b*. Both the utterances were digitized at 16 kHz and filtered using the P.341 standard. The Sennheiser microphone preserves frequencies above 3.5 kHz.

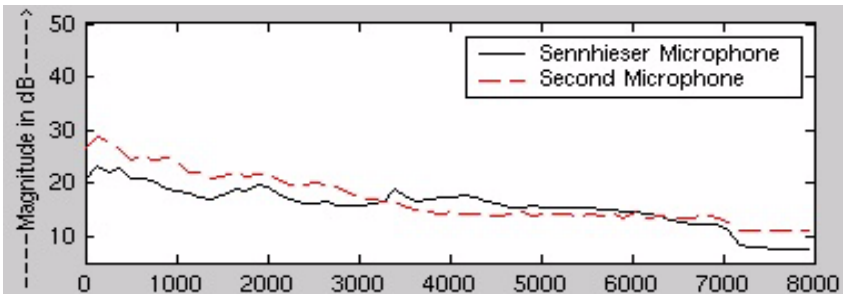

Figure 31(b). Comparison of the magnitude of the frequency response of the Sennheiser microphone and second microphone derived from the non-speech segments from the utterance id *441c020b*. Both the two utterances were digitized at 16 kHz and filtered using P.341 standard. The Sennheiser microphone preserves frequencies above 3.5 kHz.

provides the overall frequency response of the microphones on speech and non-speech data, respectively.

However, no significant improvement is observed when the sampling frequency is increased from 8 kHz to 16 kHz on matched conditions — training on Training Set 1 and decoding on Test Set 1, as shown in Figure 28 and Figure 29. These sets are matched since both consist of clean utterances recorded on Sennheiser microphone. The additional information provided by high frequencies (between 4 kHz and 8 kHz) does not result in an improvement in performance. The spectral information provided by low frequencies (below 4 kHz) is sufficient to reach the upper bound on performance.

5.1.2. Utterance Detection

In addition to the investigation whether the trends in the recognition performance were consistent at both sampling frequencies, we also investigated whether the recognition performance improved due to utterance detection. The non-speech segments of a signal recorded in noisy environments often result in an increase in insertion errors. These non-speech segments were removed from the audio data using the methodology known as endpointing, described in section 3.3.4, with an expectation that the insertion errors would reduce in noisy environments.

As expected, utterance detection resulted in a significant improvement in performance on Test Sets 2-14 when the system was trained on Training Set 1, as shown in Figure 32. Table 24 shows that the reduction in insertion errors is primarily responsible for improvement in the performance. In this case, the “silence” model learned only pure

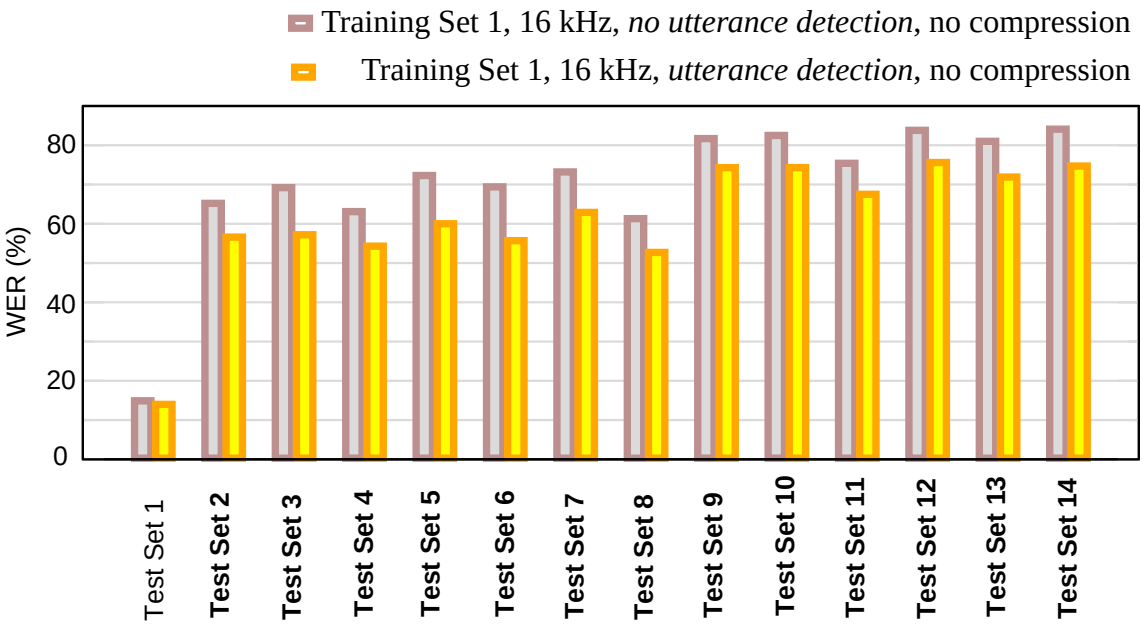

Figure 32(a). Comparison of the WER between *without* and *with utterance detection* for Training Set 1 at 16 kHz with no feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

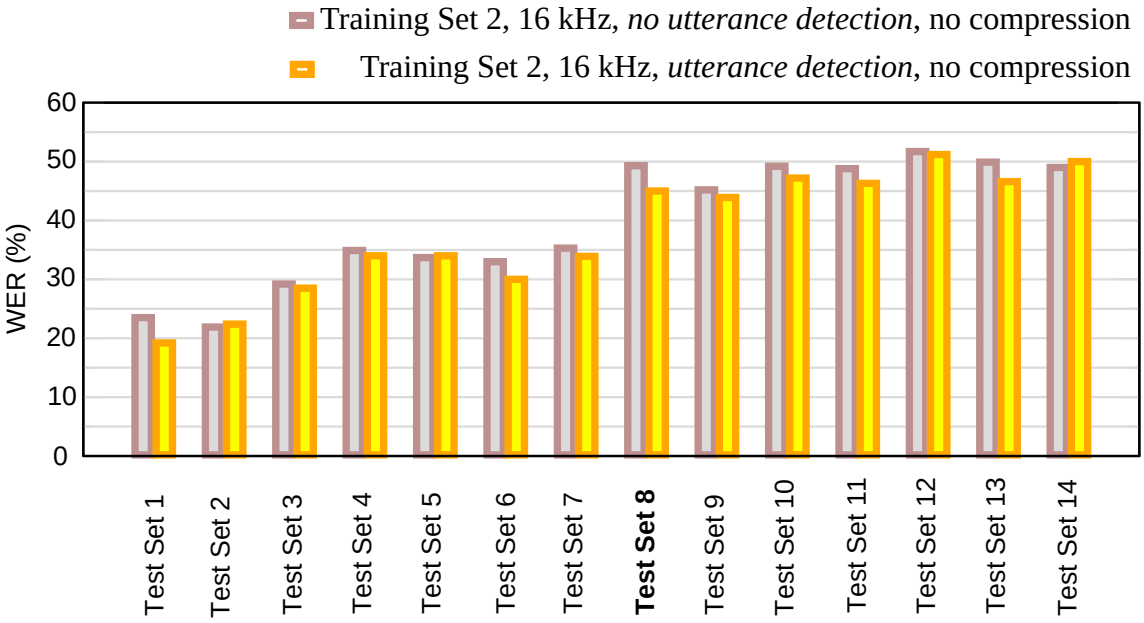

Figure 32(b). Comparison of the WER between *without* and *with utterance detection* for Training Set 2 at 16 kHz with no feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

Table 24. A comparison of experimental results for endpointed data for Training Set 1 at 16 kHz with no feature vector compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated by shaded cells.

Test Set	Training Set 1							
	Without Utterance Detection				With Utterance Detection			
	WER	Sub.	Del.	Ins.	WER	Sub.	Del.	Ins
1	14.9%	8.8%	1.0%	5.1%	14.0%	9.0%	0.8%	4.1%
2	65.2%	41.4%	3.6%	20.1%	56.6%	40.0%	3.6%	13.0%
3	69.2%	46.0%	6.5%	16.7%	57.2%	40.7%	6.2%	10.2%
4	63.1%	40.5%	12.0%	10.6%	54.3%	36.7%	10.8%	6.9%
5	72.3%	47.0%	11.2%	14.1%	60.0%	39.2%	13.8%	7.1%
6	69.4%	44.6%	7.8%	17.0%	55.7%	37.9%	8.2%	9.6%
7	73.2%	46.6%	14.1%	12.5%	62.9%	42.1%	13.7%	7.1%
8	61.3%	34.7%	14.6%	12.1%	52.7%	36.7%	8.7%	7.3%
9	81.7%	54.4%	12.3%	15.1%	74.3%	49.1%	15.1%	10.1%
10	82.5%	57.0%	12.2%	13.3%	74.3%	53.1%	13.0%	8.1%
11	75.4%	48.1%	17.9%	9.4%	67.5%	44.9%	17.5%	5.1%
12	83.8%	48.4%	26.7%	8.8%	75.6%	41.3%	30.5%	3.8%
13	81.0%	52.3%	15.5%	13.1%	71.9%	46.0%	18.4%	7.4%
14	84.1%	47.6%	26.2%	10.2%	74.7%	41.4%	28.5%	4.9%

silence during training because Training Set 1 consists of only clean data, and hence did not represent a good model of the actual background noise. Without endpointing, the noisy silences were interpreted as the non-silence words instead of silences, resulting in insertion errors. Endpointing reduced the amount of non-speech data and hence reduced insertion errors.

In contrast to Training Set 1, for Training Set 2, a significant improvement in performance was detected only for Test Set 8. A reduction in the number of deletions, rather than insertions, was primarily responsible for this improvement in performance. In other words, because the training conditions contained ample samples of the noise

conditions, the non-speech segments were modeled adequately by the silence model and hence the insertion error rate did not increase significantly on the noisy test conditions.

5.1.3. Compression

Continuing our investigation into the six focus conditions, we investigated the effects of compression on the features. It is desirable to compress feature values before transmission over a communications channel to conserve bandwidth. The compression algorithm employed in the DSR client-server application is a lossy split vector quantization (VQ) algorithm [15] that allows the quantized features to be transmitted at 4800 bps. Since this compression algorithm is lossy, the recovered features are a distorted version of the original features, and will result in a degradation in recognition performance. This degradation in performance was calibrated through a series of experiments described in Figures 33 and 34.

No significant degradation in performance due to compression was detected for Training Set 1 for both the 8 kHz and 16 kHz sampling frequencies. Since there was no significant degradation for Test Set 1, which was a matched condition, it is natural to draw a conclusion that the split VQ algorithm will not significantly degrade the performance of the system. However, Figure 34 shows that there was a significant degradation in performance for four noisy conditions at a 16 kHz sampling frequency and two noisy conditions at an 8 kHz sampling frequency on Training Set 2. We have not found a consistent explanation as to why these particular noise conditions were adversely affected.

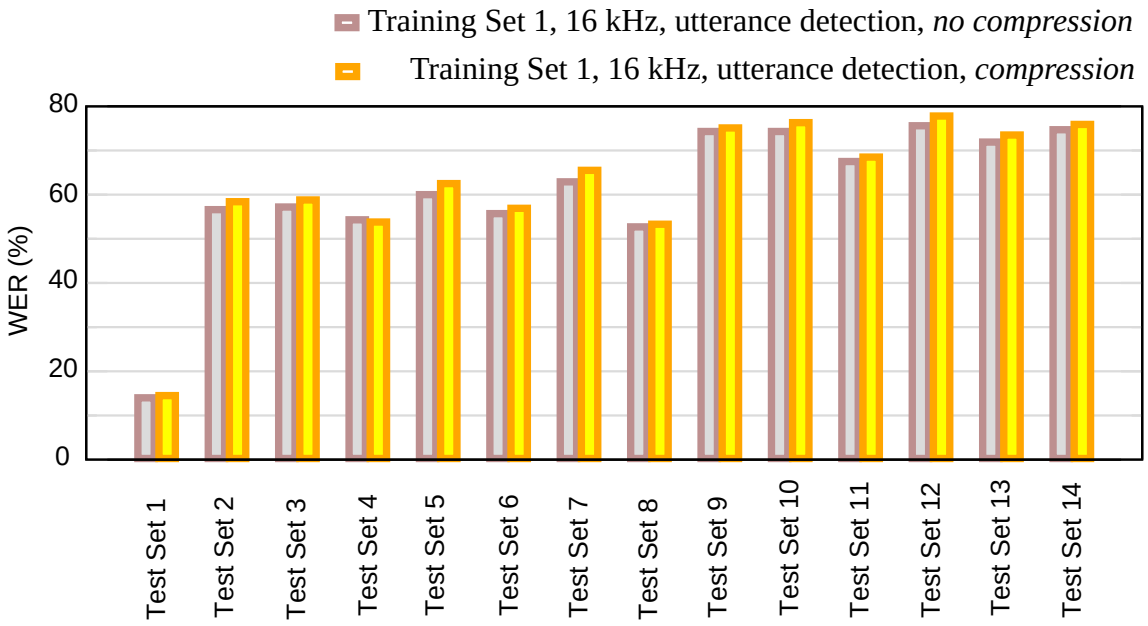

Figure 33(a). Comparison of the WER between *without* and with *compression* of feature values on Training Set 1 at 16 kHz. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

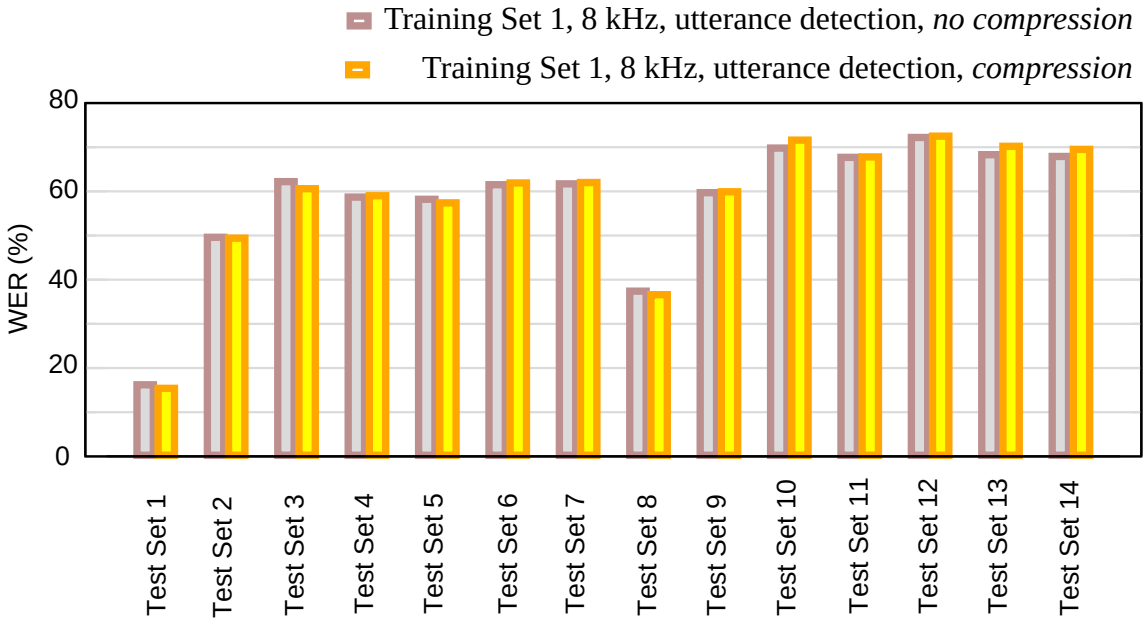

Figure 33(b). Comparison of the WER between *without* and *with compression* of feature values on Training Set 1 at 8 kHz. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

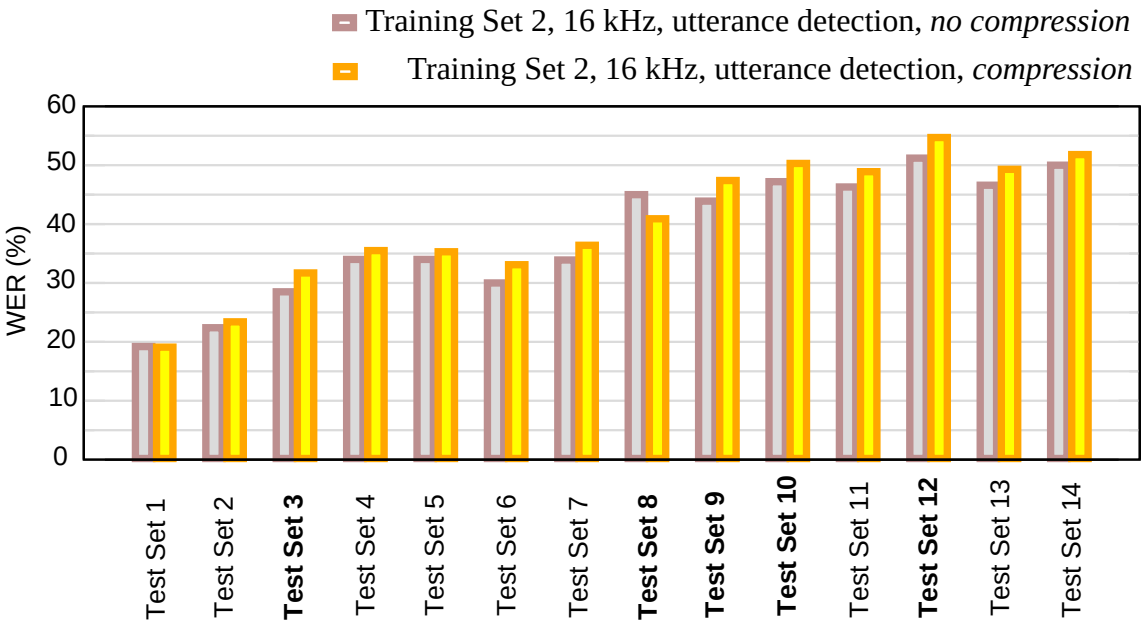

Figure 34(a). Comparison of the WER between *without* and *with compression* of feature values on Training Set 2 at 16 kHz. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

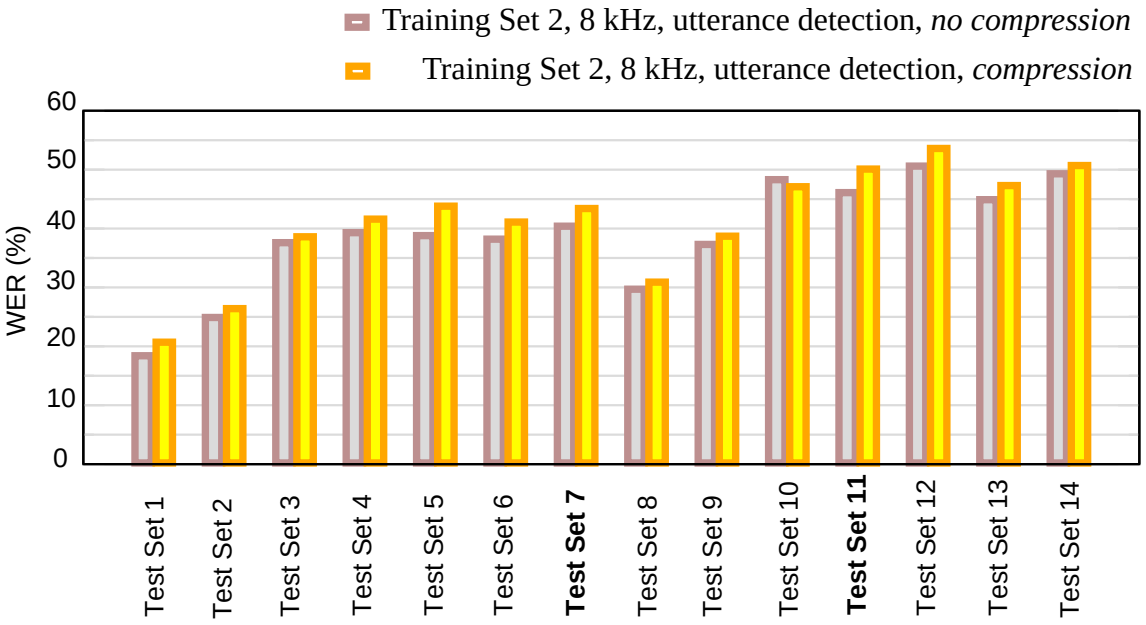

Figure 34(b). Comparison of the WER between *without* and *with compression* of feature values on Training Set 2 at 8 kHz. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

5.1.4. Model Mismatch

In addition to investigating the trends in recognition performance due to reduction in sampling frequency, utterance detection, and compression of feature values, we also investigated the effects of model mismatch on the recognition performance. One would expect to attain high recognition performance on matched conditions, defined as an experimental condition in which both the training and the test data were recorded under identical conditions. Since training is based on a maximum likelihood parameter estimation process [13,42,43], high performance recognition can only be achieved when the test conditions generate feature vectors that are similar in terms of means, variances, etc. If there are consistent differences in SNR, background noise, or microphone, there will be a significant degradation in performance if some form of adaptation is not used. In these evaluations, it was decided not to consider adaptation within the recognition system. We calibrated the degradation in performance using a series of experiments summarized in Figures 35 and 36.

As expected, the best recognition performance was observed on matched training and testing conditions (Training Set 1 and Test Set 1) in which all utterances were recorded with a Sennheiser microphone. For all other conditions involving Training Set 1, recognition performance degraded significantly. Systems trained on Training Set 2 performed significantly better than those trained on Training Set 1 across all noise conditions. These trends were consistent for both sampling frequencies and both compression conditions. Reducing this degradation from mismatched conditions through front end processing was a major goal in this evaluation.

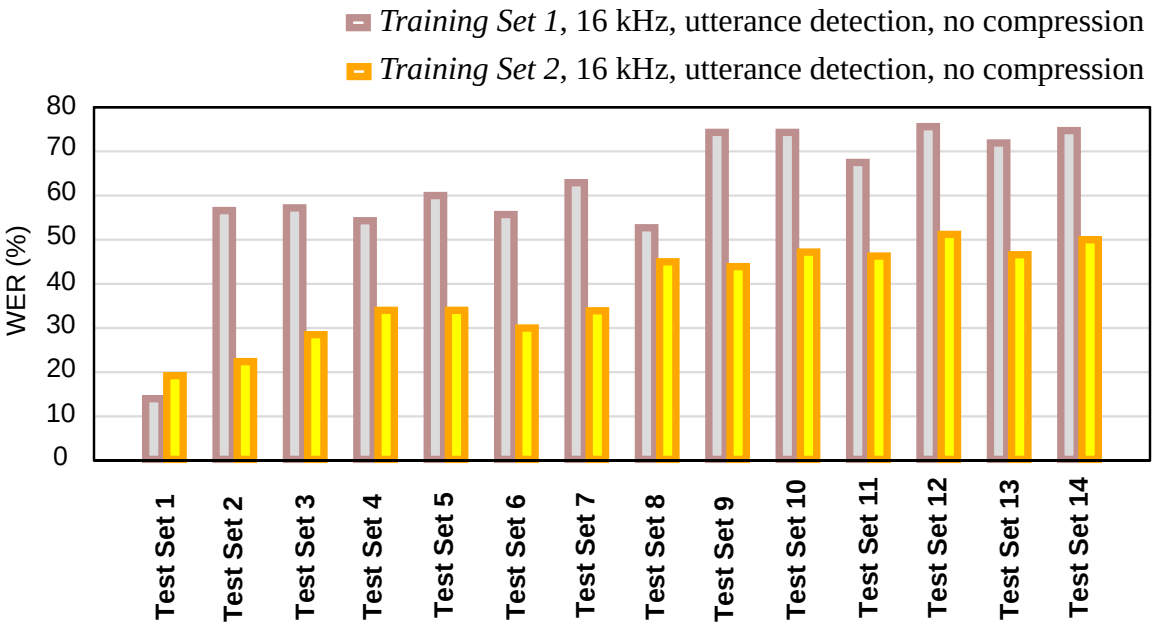

Figure 35(a). Comparison of the WER between *Training Set 1* and *Training Set 2* at 16 kHz with no feature value compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

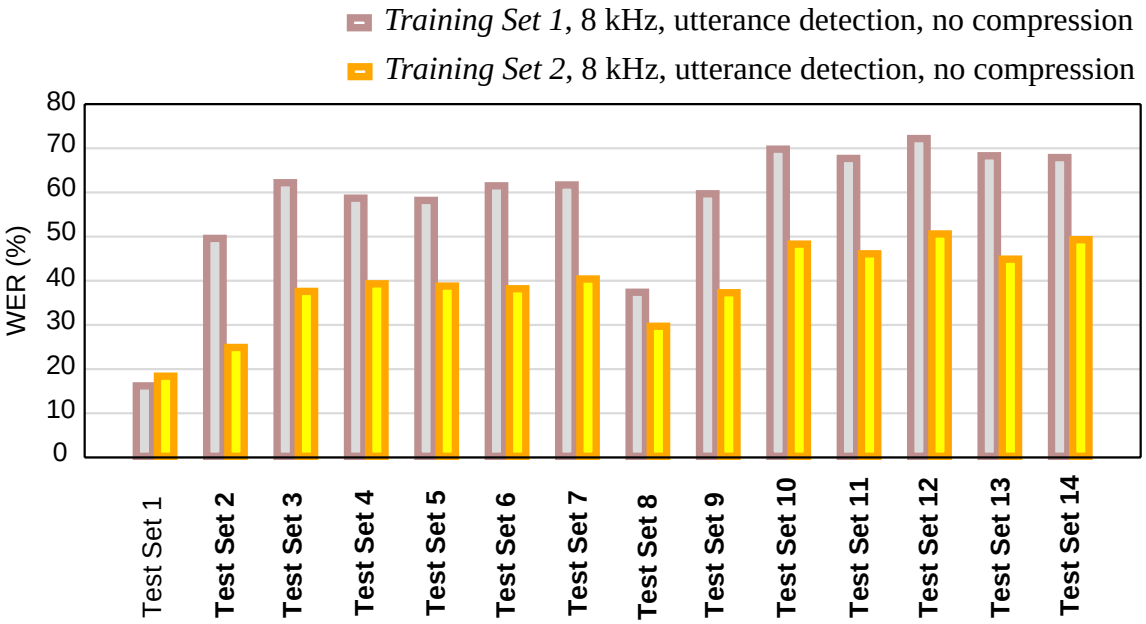

Figure 35(b). Comparison of the WER between *Training Set 1* and *Training Set 2* at 8 kHz with no feature value compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

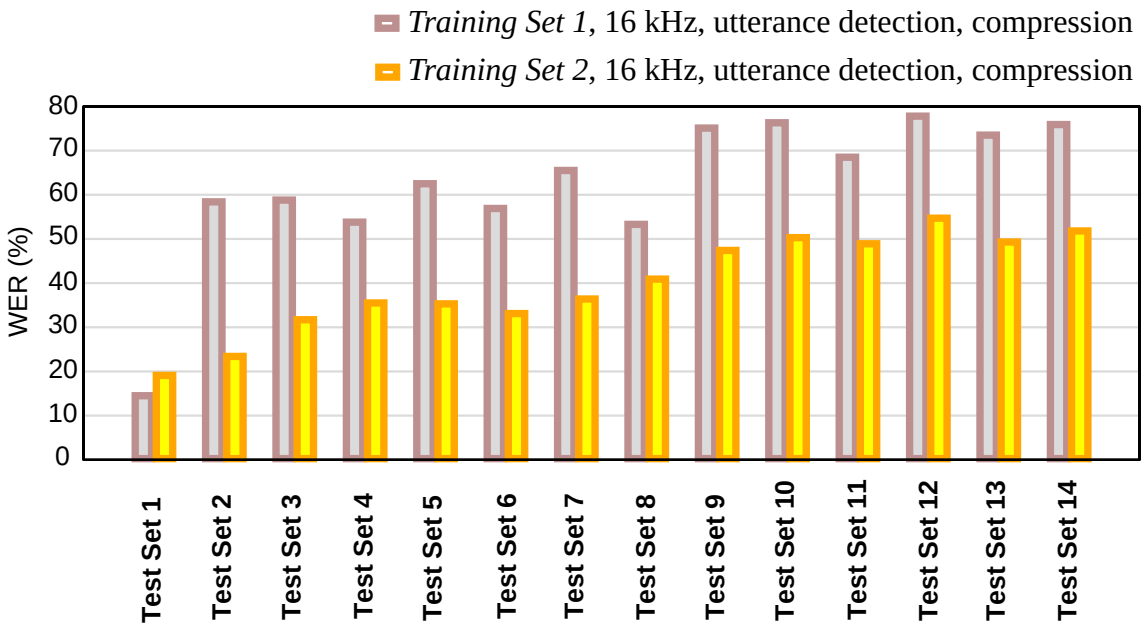

Figure 36(a). Comparison of the WER between *Training Set 1* and *Training Set 2* at 16 kHz with feature value compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

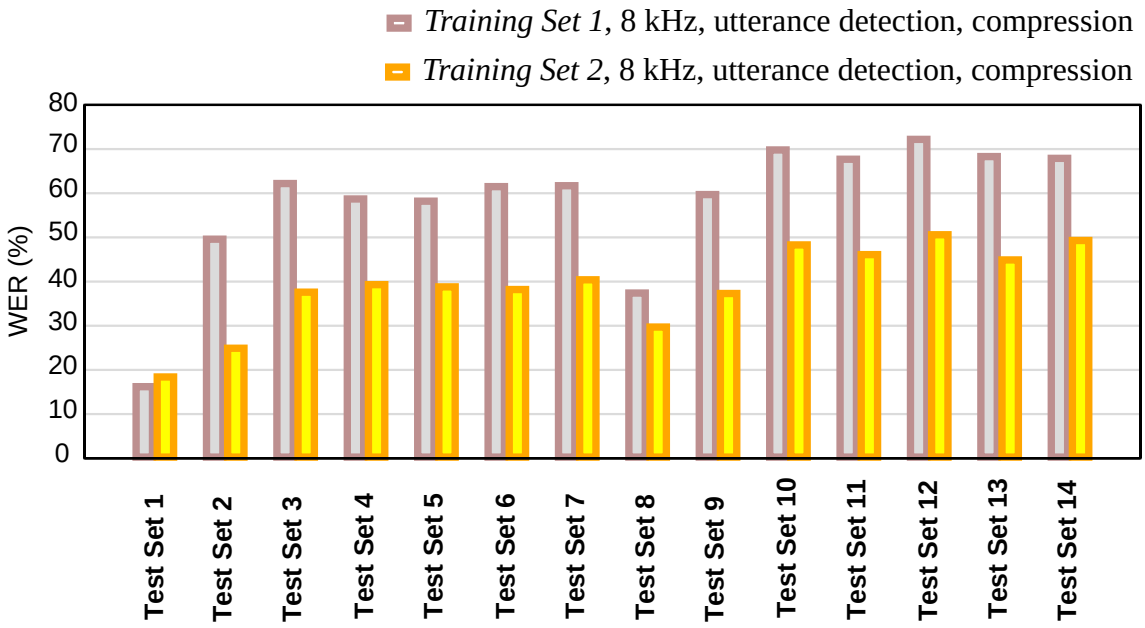

Figure 36(b). Comparison of the WER between *Training Set 1* and *Training Set 2* at 8 kHz with feature value compression. Test set conditions which are statistically significant at a 0.1% significance level are indicated in bold.

We tried to isolate the model mismatch due to additive noise from the mismatch due to microphone by training the models on Training Set 3 which consists of half of utterances from Training Set 2 recorded on Sennheiser microphone only. But as shown in Table 22, the performance was worst than the Training Set 2, even on matched microphone conditions (Test Sets 1-7), because of the reduction in training data by half. Hence, it was decided not to continue any experimentation on Training Set 3.

5.1.5. Microphone Variation

Next, we investigated the effects of microphone variation on speech recognition performance. In general, the Sennheiser microphone performed significantly better than the second microphone condition for all conditions, as shown in Table 25. The first cell in this table corresponds to Training Set 1, which consists of clean utterances recorded with a Sennheiser microphone, and Test Set 1, which consists of similar data. The second cell in the first row represents a mismatched condition in which the test set contained a different microphone. There was a significant increase in the word error rate, from 16.2% to 37.4%.

Table 25. A significant performance degradation occurs for the second microphone condition on both training sets. No compression of feature values is employed.

Performance (Without Compression)						
Training Set			Test Set			
Set	Sampling Frequency	Utterance Detection	1 (Sennheiser, Clean)	8 (Second, Clean)	2 (Sennheiser, Car)	9 (Second, Car)
1	8 kHz	Yes	16.2	37.4	49.6	59.7
2	8 kHz	Yes	18.4	29.7	24.9	37.3

The same argument of model-mismatch discussed in the previous section can be extended to explain the degradation in the performance. The same trend is observed on the car noise condition (Test Sets 2 and 9).

While Training Set 1 consists of utterances recorded with a Sennheiser microphone, Training Set 2 has half of the utterances recorded on the same Sennheiser microphone and the other half on any one of the 18 microphone types described in chapter 3. With the Baum-Welch training algorithm, which is a maximum likelihood-based parameter estimation method, this fact implies that models trained on Training Set 2 quickly converge towards the Sennheiser microphone in terms of their means and the covariances [101]. Hence, both the clean and car test conditions for the second microphone result in significant degradation in recognition performance, as shown in the second row of the Table 25. Note also that the last three cells in the second row, which correspond to various noise conditions, show less of a degradation in performance than the corresponding conditions in the first row. So there is some value in exposing the models to noise during the training process, but not as much as had been hoped for in the initial design.

5.1.6. Additive Noise

In addition to calibrating the effects on recognition performance of many signal processing issues such as sampling frequency reduction and utterance detection, we also calibrated recognition performance in presence of various background additive noise conditions. Figure 37 demonstrates the effect of these six noise conditions for two sample

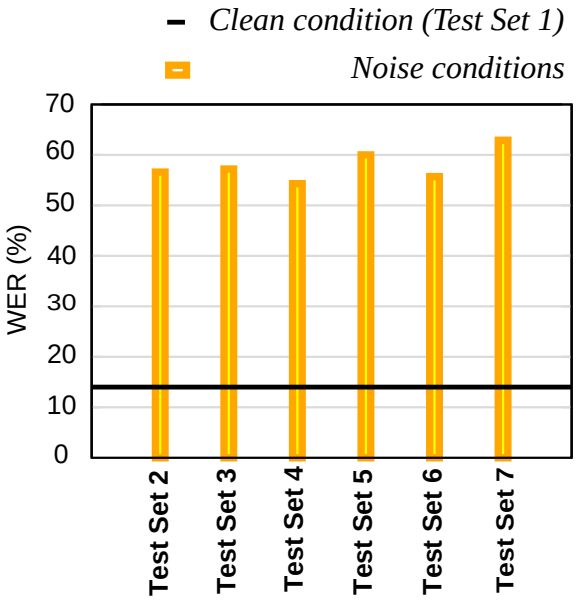

Figure 37(a). Comparison of the WER for selected noise conditions at *16 kHz* with *no feature value compression. Training Set 1* was used for training.

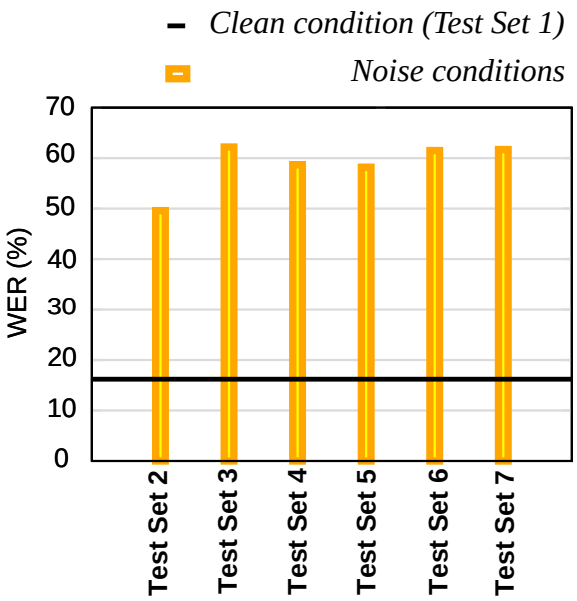

Figure 37(b). Comparison of the WER at *8 kHz* with *no feature value compression*.

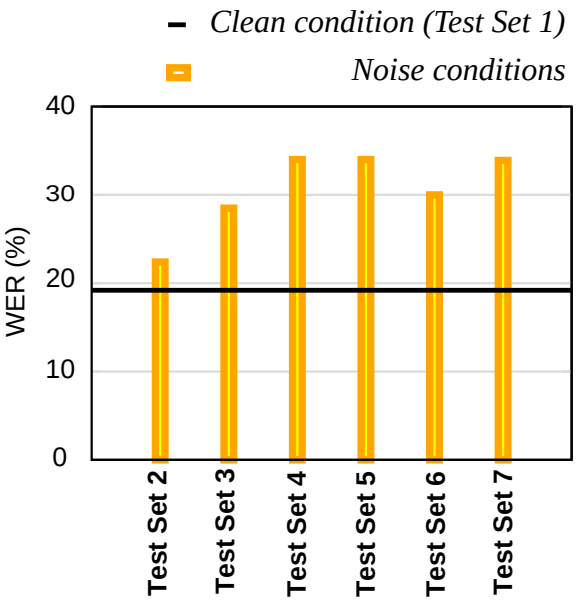

Figure 37(c). A similar comparison at *16 kHz* for *Training Set 2*.

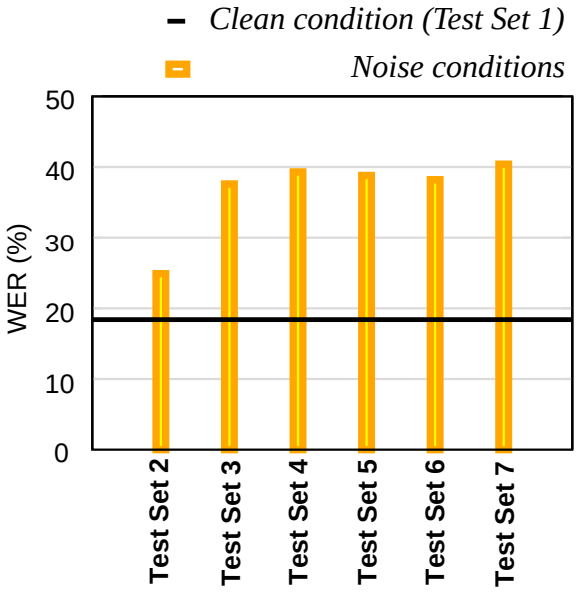

Figure 37(d). A similar comparison at *8 kHz* for *Training Set 2*.

frequencies — 8 kHz and 16 kHz. As expected, severe degradation is observed at both sample frequencies.

However, the severity of this degradation can be limited by exposing the models to noise conditions during training. In Figures 37(c) and (d), we demonstrate that the severity of the degradation in the noisy conditions is reduced by training the models on Training Set 2, which contains samples of the noise conditions. An important point to note is that these degradations are still significant compared to the clean condition. Similar trends were observed when the feature vectors were compressed.

5.2. ALV Evaluation Results

A summary of the results presented at the post evaluation ALV Workshop held in Stuttgart, Germany in February 2002 are shown in Table 26. The overall performance measure for a system was computed as an average of several WERs. First, an average WER was computed across the 14 test sets used in the evaluation for each training condition. Next, the WER for each training condition was averaged. Since the evaluation was conducted at two sample frequencies (8 and 16 kHz), the final WER was the average across both sample frequencies. This number is denoted Overall WER in Table 26. The detailed ALV results for QIO and MFA front ends are tabulated in Table 27 and Table 28, respectively.

It is obvious from results in Table 26 that the overall performance of MFA front end was slightly better than the performance of QIO front end. Both the advanced front ends achieved the overall goal of the ALV evaluation of at least 25% relative improvement

Table 26. A summary of results of the ALV evaluation using a generic baseline speech recognition system (presented at the Feb. 2002 Aurora post-evaluation meeting).

Baseline MFCC: Overall WER — 50.3%			
8 kHz — 49.6%		16 kHz — 51.0%	
TS1	**TS2**	**TS1**	**TS2**
58.1%	41.0%	62.2%	39.8%
QIO: Overall WER — 37.5%			
8 kHz — 38.4%		16 kHz — 36.5%	
TS1	**TS2**	**TS1**	**TS2**
43.2%	33.6%	40.7%	32.4%
MFA: Overall WER — 34.5%			
8 kHz — 34.5%		16 kHz — 34.4.%	
TS1	**TS2**	**TS1**	**TS2**
37.5%	31.4%	37.2%	31.5%

Table 27. Results of the QIO front end submitted to the ALV evaluation.

QIO Front End Performance Summary in ALV Evaluation															
Training Set		**Test Set**													
Set	**Samp. Freq.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**16 kHz**	13.5	22.7	35.9	41.8	37.2	40.3	39.6	33.8	40.1	51.1	52.6	55.6	52.4	53.7
	8 kHz	16.5	27.7	45.0	47.9	43.8	46.3	44.4	28.7	39.3	50.3	54.3	55.2	52.2	52.7
2	**16 kHz**	16.7	17.7	25.3	31.5	29.3	26.9	27.9	31.5	33.4	42.2	43.6	43.6	41.3	42.0
	8 kHz	20.8	22.4	33.0	37.2	35.0	33.3	35.3	23.6	27.6	37.9	43.5	42.1	37.1	41.3

Table 28. Results of the MFA front end submitted to the ALV evaluation.

MFA Front End Performance Summary in ALV Evaluation															
Training Set		**Test Set**													
Set	**Samp. Freq.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**16 kHz**	13.6	23.0	31.8	39.0	34.1	33.3	34.1	32.3	39.7	45.9	49.5	49.8	47.8	46.2
	8 kHz	15.0	21.1	36.3	43.9	36.6	42.9	37.4	23.1	29.5	44.7	52.1	49.2	48.1	45.2
2	**16 kHz**	15.4	18.7	24.8	32.5	27.8	26.6	27.8	29.7	32.6	38.3	43.5	42.2	40.4	40.4
	8 kHz	17.2	19.0	29.9	38.5	31.5	33.8	30.9	22.7	24.8	34.7	43.7	39.8	36.4	36.9

over the baseline MFCC front end. However, the overall performance of the MFA front end at 34.5% WER is almost ten times worse than human performance reported on a similar task — human transcription of broadcast news speech [34,31]. Hence, the performance of the advanced front ends is far from what is needed in practical applications. Further research is needed to develop noise robust algorithms that close this gap in performance.

5.3. Analysis of the ALV Evaluation

To do further analysis of the ALV evaluation, we acquired the code used to produce the results previously discussed, and re-ran the evaluations within our laboratory. Due to bug fixes and other changes made by the algorithm developers, the results fluctuated slightly. A summary of the results of these new experiments at a sampling frequency of 8 kHz are shown in Table 29. The changes in performance were not statistically significant.

Table 30 and Table 31 present the detailed results for QIO front end and MFA front end, respectively. Significance tests [100] on the 14 test conditions for Training Set 1 showed that the performance of the MFA front end was significantly better than the QIO front end performance on all 14 test conditions. However, on Training Set 2, the MFA front end was significantly better for only Test Sets 5 and 14. Training Set 2 is representative of all noise conditions and includes microphone mismatches. Hence, Training Set 1 is a good measure for front-end robustness, and perhaps more informative than the matched conditions (Training Set 2). For the ALV evaluations, WERs on the two

Table 29. A summary of results of the experiments that represented the replication of the ALV evaluation using a generic baseline speech recognition system.

Baseline MFCC	
8 kHz — 49.6%	
TS1	**TS2**
58.1%	41.0%
QIO	
8 kHz — 38.4%	
TS1	**TS2**
43.1%	33.6%
MFA	
8 kHz — 34.7%	
TS1	**TS2**
37.5%	31.8%

Table 30. Results of the experiments that represented the replication of the QIO front end results submitted to the ALV evaluation.

QIO Front End Performance Summary in ALV Evaluation														
Training Set		**Test Set**												
Set	**Samp. Freq.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**8 kHz**	17.1	27.2	44.1	47.0	43.1	48.9	44.6	27.5	39.5	49.8	54.9	55.9	52.1	52.0
2	**8 kHz**	20.9	22.1	32.8	37.4	35.4	33.6	35.2	24.2	27.4	37.5	42.7	42.2	37.3	41.1

Table 31. Results of the experiments that represented the replication of the MFA front end results submitted to the ALV evaluation.

MFA Front End Performance Summary in ALV Evaluation														
Training Set		**Test Set**												
Set	**Samp. Freq.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**8 kHz**	14.5	22.1	37.0	43.2	36.6	43.3	38.2	24.3	29.8	43.4	50.6	48.7	48.6	44.9
2	**8 kHz**	18.1	20.6	30.9	36.8	31.6	33.8	31.7	24.3	24.8	34.7	43.3	40.3	38.1	35.7

training sets were weighted equally, thereby decreasing the gap between the two front ends.

In Table 32, we calibrate the degradation in the performance of the three front ends due to the microphone mismatch. Training Set 1 consisted of clean data recorded with a Sennheiser microphone. Test Set 1 also represents clean data recorded through the same microphone. Test Set 8 represents a mismatched condition since it consists of clean data recorded through the second microphone condition. Though both front ends degraded significantly due to microphone mismatch, this degradation is less severe than the MFCC-based baseline system. The baseline system did not employ any channel normalization techniques such as cepstral mean subtraction.

As shown in Figures 38 and 39, the presence of additive noise resulted in a significant degradation in performance for both the QIO and MFA front ends. The bold labels in these figures represent differences which are statistically significant. This trend is similar to the tend observed on the Aurora MFCC front end-based baseline system as discussed in section 5.1 though the degradations are less severe. The degradation is also less severe when the systems are exposed to noise during training. Performance on the same noisy test sets was much better when training on Training Set 2 because Training Set 2 contains examples of all noise and microphone types.

Table 32. A performance comparison for a mismatched microphone condition.

Train Set	**WI007 MFCC Baseline**		**QIO**		**MFA**	
	Test Set 1 (Senn Mic.)	**Test Set 8 (Sec. Mic.)**	**Test Set 1 (Senn. Mic.)**	**Test Set 8 (Sec. Mic.)**	**Test Set 1 (Senn. Mic.)**	**Test Set 8 (Sec. Mic.)**
1	15.4%	36.6%	17.1%	27.5%	14.5%	24.3%

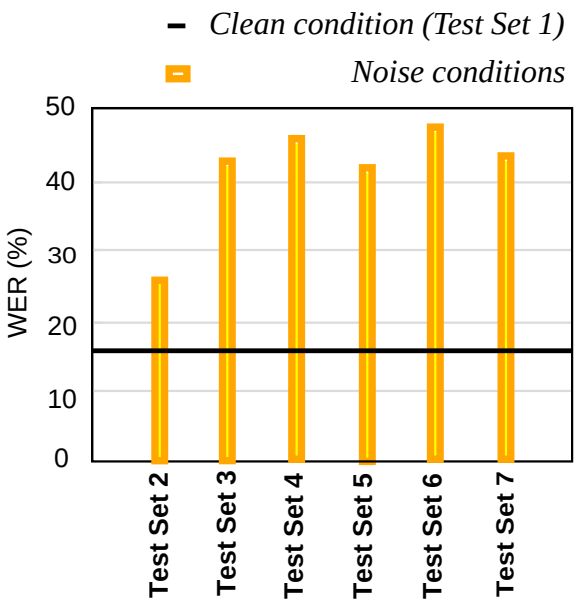

Figure 38(a). Comparison of the WER for selected noise conditions at *8 kHz* on the *QIO* front end. *Training Set 1* was used for training.

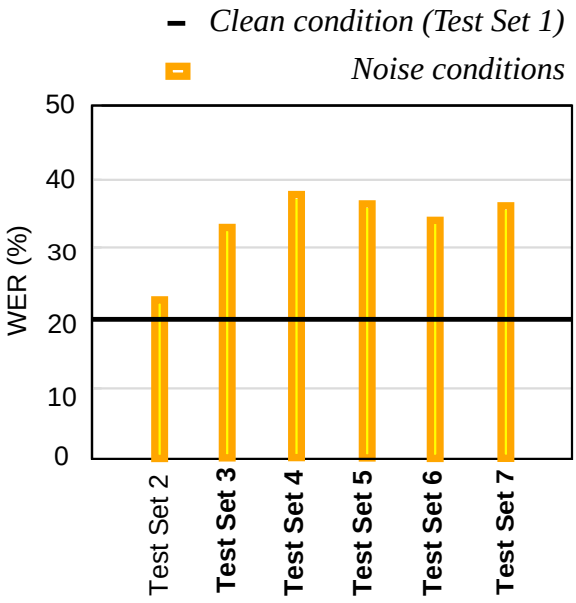

Figure 38(b). Comparison of the WER for selected noise conditions at *8 kHz* on the *QIO* front end. *Training Set 2* was used for training.

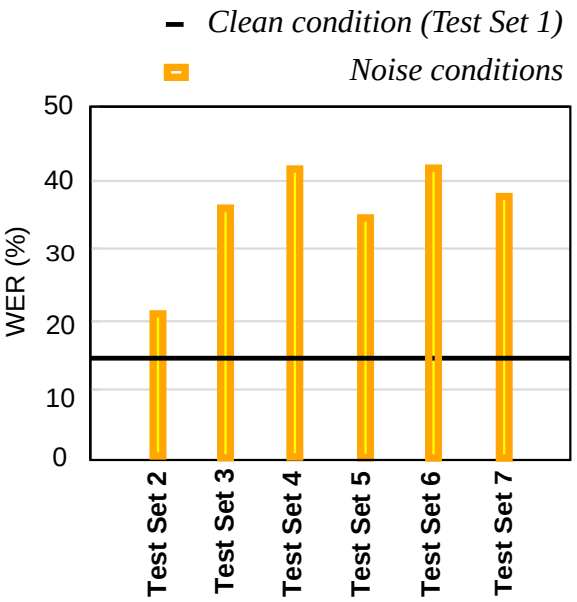

Figure 39(a). Comparison of the WER for selected noise conditions at *8 kHz* on the *MFA* front end. *Training Set 1* was used for training.

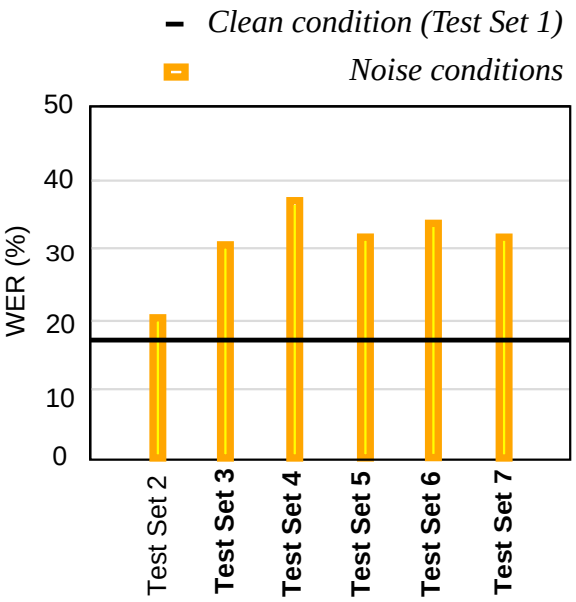

Figure 39(b). Comparison of the WER for selected noise conditions at *8 kHz* on the *MFA* front end. *Training Set 2* was used for training.

5.4. Front End-specific Tuning Experiments

There are four classes of parameters that are most relevant to the tuning performed for this evaluation. Two of these relate to language model and acoustic scores. The language model scale factor controls the relative weight of the language model probabilities compared to the acoustic model probabilities. The word insertion penalty is applied to every word hypothesis and is used to balance insertion and deletion errors. The language model scale factor typically ranges from 5 on tasks such as DARPA's Resource Management corpus [102] to 20 on tasks such as WSJ [28]. The word insertion penalty usually ranges from -10 on Resource Management to +10 on WSJ.

The second class of parameters, which have perhaps the most significant impact on performance, relate to the state tying process. The number of tied states can normally be adjusted to improve performance. This parameter balances sparsity and generalization of the data in the phonetic decision tree state tying process. We typically reduce the number of states by an order of magnitude. We can also control the degree to which states are merged or split by adjusting parameters related to the likelihood of the state.

In Table 33 and Table 34, we show the difference in performance between the baseline system and the tuned system for the QIO and MFA front ends, respectively. The tuning process is described in more detail in section 4.2. Parameter tuning was performed on the matched training condition at 8 kHz (Training Set 1) using the 330-utterance short development test set. The beam pruning parameters (state, model and word) were opened during the tuning process to reduce the influence of pruning. As shown in Table 33 and Table 34, parameter tuning resulted in a small overall improvement — about 1% absolute

Table 33. A comparison of the optimized system parameters to the baseline system parameters for the QIO front end. Beam pruning parameters were set to 300 (state), 250 (model), and 250 (word).

QIO	Num. States	State Tying Thresholds			LM Scale	Word Ins. Penalty	WER
		Split	Merge	Occu.			
Baseline	3209	165	165	840	18	10	16.1%
Tuned	3512	125	125	750	20	10	14.9%

Table 34. A comparison of the optimized system parameters to the baseline system parameters for the MFA front end.

QIO	Num. States	State Tying Thresholds			LM Scale	Word Ins. Penalty	WER
		Split	Merge	Occu.			
Baseline	3208	165	165	840	18	10	13.8%
Tuned	4254	100	100	600	18	05	12.5%

and 8% relative. The amount of improvement was about the same for both systems — 7.5% relative for QIO and 9.4% relative for MFA. Hence, the ranking of the systems remained the same.

The tuned systems were then benchmarked on the Aurora-4 database. A summary of the results on these benchmark experiments is shown in Table 35, and the detailed results are provided in Table 36 and Table 37. The pruning beams were scaled back to the values used in the ALV baseline system: 200 (state), 150 (model), and 150 (word). It was observed that the overall relative ranking of the two competitive front ends is not influenced by the tuning process. The average performance of the MFA front end without tuning was better than QIO by 9.6% relative. Front end-specific tuning resulted in an increase in the relative performance gap between the two front ends from 9.6% to 15.8%. While the average performance of the MFA front end remained relatively constant (34.7%

Table 35. A summary of the performance of the QIO and MFA front ends after front end-specific system tuning.

Baseline MFCC	
8 kHz — 49.6%	
TS1	**TS2**
58.1%	41.0%
QIO	
8 kHz — 40.5%	
TS1	**TS2**
45.7%	35.3%
MFA	
8 kHz — 34.1%	
TS1	**TS2**
37.0%	31.1%

Table 36. Performance of the QIO front end after front end-specific system tuning.

QIO Front End Performance Summary in ALV Evaluation														
Training Set		**Test Set**												
Set	**Samp. Freq.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**8 kHz**	19.1	31.7	46.8	49.2	45.7	51.1	46.6	30.0	42.2	52.9	55.5	58.3	54.8	55.8
2	**8 kHz**	22.5	23.8	33.6	38.1	36.4	36.2	37.7	25.0	29.5	39.1	44.5	45.0	40.5	41.8

Table 37. Performance of the MFA front end after front end-specific system tuning.

MFA Front End Performance Summary in ALV Evaluation														
Training Set		**Test Set**												
Set	**Samp. Freq.**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	**8 kHz**	14.4	21.5	36.8	42.1	36.5	44.1	36.4	23.3	30.2	43.0	50.2	48.9	47.0	43.6
2	**8 kHz**	16.8	20.7	29.7	36.0	31.0	33.3	32.0	22.5	24.6	34.1	42.3	39.4	37.1	36.1

to 34.1%), the average performance of the QIO front end dropped by 5.5% relative (38.4% to 40.5%). One possible reason for this drop can be attributed to overfitting of the system parameters on the specific database employed for the tuning process (matched conditions: Training Set 1 and short devtest set 1).

All these results provide sufficient evidence to conclude that the front end-specific tuning process did not result in a change in the ranking of the advanced front ends. We also showed that though the advanced front end achieved significant improvement (greater than 25% relative) in performance over the baseline MFCC front end, the performance of these advanced front ends is very high (~35%) compared to human performance (~1%) in noisy environments. Hence, we conclude that the noise robust technology implemented in the advanced front ends is not operationally significant in practical applications.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

This thesis analyzed the performance of advanced front ends and demonstrated that the performance of the advanced front ends is significantly better than the baseline industry standard MFCC front end, but is not operationally significantly. It was also shown that front end-specific tuning of a recognition system did not significantly change the results of the ALV evaluation.

6.1. Thesis Contributions

There are three major contributions in this thesis. These are described in detail below.

Development of the Aurora Baseline System

A baseline large vocabulary continuous speech recognition system was developed. This system design was tuned to reduce computation time without significantly degrading the overall system performance. The real time performance of the baseline system was 4 xRT for training and 15 xRT for decoding on an 800 MHz Pentium processor. On the standard 5K WSJ0 task, the ALV baseline system WER performance was 14.0%.

Analysis of Performance

An extensive analysis of the performance of the ETSI WI007 MFCC baseline front end and two advanced front ends (QIO and MFA) was presented. For the baseline front end, it was shown that increasing the sampling frequency from 8 kHz to 16 kHz results in a significant performance improvement only for the noisy test conditions. Utterance detection resulted in significant improvements only on the noisy conditions for the mismatched training case. The DSR standard VQ-based compression algorithm did not result in a significant degradation in performance. A mismatch between training and testing conditions resulted in a 300% relative increase in WER whereas the mismatches in microphones resulted in a 200% relative increase in WER.

Both the QIO and MFA advanced front ends did not degrade as dramatically as the baseline MFCC front end on mismatched microphone and additive noise conditions though these degradations were significantly worse than the matched conditions. In fact, both advanced front ends met the goals set forth in the ALV evaluation — a 25% improvement in performance over the baseline system.

The performance of the MFA front end for Training Set 1 was significantly better than the QIO front end performance on all 14 test conditions. Training Set 1 consists of clean utterances recorded on a Sennheiser microphone while the 14 test conditions are representative of all noise and microphone conditions. Hence, training on Training Set 1 and decoding on 14 test conditions represents highly mismatched evaluation conditions. The overall performance on these highly mismatched conditions is a good measure of front end robustness.

However, on Training Set 2, the MFA front end was significantly better only for Test Sets 5 and 14. Training Set 2 is representative of all noise conditions and includes microphone mismatches. While training on Training Set 2, the models learned these noise and microphone conditions under a maximum likelihood framework and hence, the performance of both advanced front ends was comparable.

Due to microphone mismatch, both advanced front ends degraded significantly. However, this degradation is less severe than the MFCC-based baseline system. The baseline system did not employ any channel normalization technique.

The presence of additive noise resulted in a significant degradation in performance for both the QIO and MFA front ends. This trend is similar to the trend observed on the baseline MFCC front end though the degradations were less severe.

Analysis of Parameter Tuning

It has been shown that the overall relative ranking of the two front ends was not influenced by the tuning process. The average performance of the MFA front end without tuning is better than QIO by 9.6% relative. Front end-specific tuning resulted in an increase in the relative performance gap between the two front ends from 9.6% to 15.8%. While the average performance of the MFA front end remained relatively constant (34.7% to 34.1%), the average performance of the QIO front end dropped by 5.5% relative (38.4% to 40.5%). One possible reason for this drop can be attributed to overfitting of the system parameters on the specific database (e.g., matched conditions using Training Set 1 and short devtest set 1) employed in the tuning process.

6.2. Future Work

A major limitation in this work was the lack of access to a modular source code implementation of each of these front ends. These front ends contain many techniques that individually or collectively improve performance. The contribution of each of these algorithms to the overall improvement in performance can be calibrated by benchmarking these algorithms in isolation. In this way, a more detailed understanding of the efficacy of these approaches can be established.

Due to CPU limitations, recognition system parameter tuning was performed only on one condition: training on Training Set 1 and testing on short devtest set 1. It might be argued that because the tuning conditions are different than the actual test conditions, an improvement in the performance of the advanced front ends can be obtained by tuning on the mismatched conditions that have ample samples of the noise and microphone type. This conjecture can be tested with appropriate additional experimentation.

Finally, the improvements in these algorithms needs to be verified with a recognition system that utilizes more state of the art features, such as speaker and channel normalization [61,87,88], speaker and channel adaptation [31,88,103], and discriminative training [104,105,106].

REFERENCES

[1] F. Zheng, J. Picone, “Robust Low Perplexity Voice Interfaces,” h*ttp:// www.isip.msstate.edu/publications/reports/mitre_robust/2001/*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, USA, May 2001.

[2] A. Martin, M. Przybocki, “The 2001 NIST Evaluation for Recognition of Conversational Speech Over the Telephone,”, *Proceedings of the 2001 Speech Transcription Workshop*, Gaithersburg, Maryland, USA, May 2001.

[3] D. Pallett, *et. al*., “Overview: Speech Transcription Workshop,” *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, Maryland, USA, May 2000.

[4] P. C. Woodland, *et. al.*, “CU-HTK April 2002 Switchboard System,” *http://svr-www.eng.cam.ac.uk/reports/svr-ftp/woodland_rt02.pdf*, Cambridge University Engineering Department, May 2002.

[5] R. Lippmann, “Speech recognition by machines and humans”, *Speech Communications Journal*, vol. 1, pp. 1-15, 1997.

[6] J. M. Huerta, *Speech Recognition in Mobile Environment*, Ph. D. Dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA, April 2000.

[7] D. Pearce, “Enabling New Speech Driven Services for Mobile Devices: An overview of the ETSI standards activities for Distributed Speech Recognition Front-ends,” presented at the Applied Voice Input/Output Society Conference (AVIOS2000), San Jose, California, USA, May 2000.

[8] D. Pearce, “Developing the ETSI Aurora Advanced Distributed Speech Recognition Front-end & What Next ?,”*Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio Trento, Italy, December 2001.

[9] J. Picone, “Signal Modeling Techniques in Speech Recognition”, *IEEE Proceedings*, vol. 81, no. 9, pp. 1215-1247, September 1993.

[10] J. G. Wilpon, B. H. Juang, and L. R. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition," *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 821-824, Dallas, USA, April 1987.

[11] J. Picone, "Continuous Speech Recognition Using Hidden Markov Models," *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 7, no. 3, pp. 26-41, July 1990.

[12] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1993.

[13] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, USA, 1997.

[14] G. Hirsch and D. Pearce, "Second Experimental Framework for the Performance Evaluation of Speech Recognition Front-ends," STQ Aurora DSR Working Group, February 2000.

[15] "ETSI ES 201 108 v1.1.2 Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithm," ETSI, April 2000.

[16] L. Lamel *et. al*., "Recent Developments in Spoken Language Sytems for Information Retrieval," *ESCA Workshop on Spoken Dialog Systems*, Vigsø, Denmark, 1995.

[17] S. Euler and J. Zinke, "The influence of speech coding algorithms on automatic speech recognition," *Proceedings of ICASSP*, vol. 1, pp. 621624, Adelaide, Australia, 1994.

[18] B. Lilly and K. Paliwal, "Effect of speech coders on speech recognition performance," *Proceedings of ICSLP*, vol. 4, pp. 23442347, Philedelphia, PA, USA, October 1996.

[19] B. Milner, "Robust speech recognition in burst-like packet loss," *Proceeding of ICASSP*, vol. I, pp. 261264, Salt Lake City, Utah, USA, May 2001.

[20] Y. Gong, "Speech Recognition in Noisy Environments: A Survey," *Speech Communication*, vol. 16, pp. 261-291, 1995.

[21] P. Haavisto, "Speech Recognition for Mobile Communications," *Proceedings of the COST Workshop on Robust Methods for Speech Recognition in Adverse Conditions*, Tampere, Finland, 1999.

[22] D. Pearce and G. Hirsch, ”The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition System under Noisey Conditions”, *Proceedings of ICSLP*, Beijing, China, 2000.

[23] R. Leonard, “A Database for Speaker-Independent Digit Recognition,” *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 9, pp. 328-331, March 1984.

[24] “Recommendation G.712 — Transmission performance characteristics of pulse code modulation channels,” International Telecommunication Union (ITU), Geneva, Switzerland, November 1996.

[25] M. Woszczyna, *Fast Speaker Independent Large Vocabulary Continuous Speech Recognition*, Ph.D. dissertation, Karlsruhe, Germany, February 1998.

[26] A. Moreno *et. al.*, “SPEECHCAR-DAT. A Large Speech Database for Automotive Environments,” *Proceedings of LREC*, Athens, Greece, May 2000.

[27] D. Pearce, “Overview of Evaluation Criteria for Advanced Distributed Speech Recognition,” ETSI STQ-Aurora DSR Working Group, October 16, 2001.

[28] D. Paul and J. Baker, “The Design of Wall Street Journal-based CSR Corpus,” *Proceedings of ICSLP*, pp. 899-902, Banff, Alberta, Canada, October 1992.

[29] C. Benitez, *et. al.*, “Robust ASR front-end using spectral-based and discriminant features: experiments on the Aurora Task”, *Proceedings of Eurospeech’01*, Aalborg, Denmark, September 2001.

[30] Dusan Macho, *et. al.*, ”Evaluation of a Noise-robust DSR Front-end on Aurora Databases”, *Proceedings of ICSLP*, pp. 17-20, Denver, Colorado, USA, September 2002.

[31] David S. Pallet, *et. al.*, ”1994 Benchmark Tests for the ARPA Spoken Language Program”, *Proceeding of the Eighth Spoken Language Systems Technology Workshop*, Austin, Texas, USA, January 1995.

[32] P. J. Moreno, M. A. Siegler, U. Jain, and R. M. Stern, ”Approaches to Microphone Independence in Automatic Speech Recognition”, *Proceeding of the Eighth Spoken Language Systems Technology Workshop*, Austin, Texas, USA, January 1995.

[33] R. A. Gopinath, *et. al*., ”Robust Speech Recognition in Noise - Performance of the IBM Continuous Speech Recognizer on the ARPA Noise Spoke Task”, *Proceeding of the Eighth Spoken Language Systems Technology Workshop*, Austin, Texas, USA, January 1995.

[34] W.J. Ebel and J. Picone, "Human Speech Recognition Performance on the 1994 CSR Spoke 10 Corpus," *Proceedings of the Spoken Language Systems Technology Workshop*, pp. 53-59, Austin, Texas, USA, January 1995.

[35] N. Deshmukh, *et. al.*, "Human Speech Recognition Performance on the 1995 CSR Hub-3 Corpus," *Proceedings of the Speech Recognition Workshop*, pp. 129-134, Harriman, New York, USA, February 18-21, 1996.

[36] N. Deshmukh, *et. al.*, "Benchmarking Human Performance for Continuous Speech Recognition," *Proceedings of the Fourth International Conference on Spoken Language Processing*, pp. SuP1P1.10, Philadelphia, Pennsylvania, USA, October 1996.

[37] N. Parihar and J. Picone, “DSR Front End LVCSR Evaluation,” AU/384/02, Aurora Working Group, December 2002 (*http://www.isip.msstate.edu/projects/aurora*).

[38] S. Davis and P. Mermelstein, “Comparision of Parameteric Representations for Monosyllable Word Recognition in Continuous Spoken Sentences,” *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28(4), pp. 357-366, 1980.

[39] H. Hermansky, “Perceptual Linear Predictive (PLP) Analysis of Speech,” *Journal of the Acoustical Society of America*, vol. 4, pp. 1738-1752, 1990.

[40] V. R. R. Gadde, A. Stolcke, D. Vergyri, J. Zheng, K. Sonmez, & A. Venkataraman, “SRI 2001 SPINE Evaluation System,” *Presentation at the DARPA SPINE Workshop*, Orlando, Florida, USA, November 2001.

[41] V. Mantha, R. Duncan, Y. Wu, J. Zhao, A. Ganapathiraju and J. Picone, "Implementation and Analysis of Speech Recognition Front-Ends," *Proceedings of the IEEE Southeastcon*, pp. 32-35, Lexington, Kentucky, USA, March 1999.

[42] L. R. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1993.

[43] X. Huang, A. Acero and H. Hon, *Spoken Language Processing*, Prentice Hall, Upper Saddle River, New Jersey, USA, 2001.

[44] J. Markel and A. H. Gray, Jr., *Linear Pre-diction of Speech*, Springer-Verlag, New York, New York, USA, 1980.

[45] A. Ganapathiraju, J. Hamaker, A. Skjellum, and J. Picone, “A Comparative Analysis of FFT Algorithms,” submitted to the *IEEE Transactions on Signal Processing*, December 1997. (*http://www.isip.msstate.edu/publications/journals/*).

[46] A. V. Oppenheim and D. H. Johnson, “Discrete Representation of Signals,” Proceedings of the IEEE, pp. 681-691, June 1972.

[47] D. O’Shaughnessy, Speech Communication: *Human and Machine*, Addison Wesley, New York, New York, USA, 1987.

[48] A. R. Møller, *Auditory Physiology*, Academic Press, New York, New York, USA, 1983.

[49] A. V. Oppenheim, R. W. Schafer and T. G. Stockham, “Nonlinear Filtering of Multiplied and Convolved Signals,” *Proceedings of the IEEE*, vol. 56, pp. 1264-1291, 1968.

[50] S. Young, *et. al*, *The HTK Book (Version 3.0)*, Microsoft Corporation, Redmond, Washington, USA, pp. 60, July 2000.

[51] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, “On the Use of Bandpass Liftering in Speech Recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 7, pp. 947-954, July 1987.

[52] B. A. Hanson, T. H. Applebaum, J. C. Junqua, “Spectral Dynamics for Speech Recognition Under Adverse Conditions,” in *Advanced Topics in Automatic Speech and Speaker Recognition*, C.-H. Lee, K. K. Paliwal and F. K. Soong, Eds., Kluwer Academic Publishers, New York, New York, USA, 1995.

[53] J. Hamaker, N. Deshmukh, A. Ganapathiraju and J. Picone, "Resegmentation and Transcription of the SWITCHBOARD Corpus," *Proceedings of Speech Transcription Workshop*, Linthicum Heights, Maryland, USA, September 1998.

[54] S. Furui, “Speaker-Independent Isolated Word Recognition Using Dynamic Features of the Speech Spectrum,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 52-59, February 1986.

[55] E. L. Bocchieri and G. R. Doddington, “Frame Specific Statistical Features for Speaker-Independent Speech Recogni-tion,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 755-764, August 1986.

[56] F. K. Soong and A. E. Rosenberg, “On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition,” *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 877-880, April 1986.

[57] J. Picone, “Adding Temporal Information: Derivatives,” *http://www.isip.msstate.edu/conferences/srstw01/program/session_04/signal_processing/html/sp_15.html*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, USA, May 2001.

[58] C. Benitez, *et al.*, “Robust ASR front-end using spectral-based and discriminant features: experiments on the Aurora Task”, *Proceedings of the European Conference on Speech Communication and Technology*, Aalborg, Denmark, September 2001.

[59] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, num. 4, pp. 578-589, October 1984.

[60] S. van Vuuren and H. Hermansky, “Data-driven Design of Rasta-like Filters,” *Proceedings of the European Conference on Speech Communication and Technology*, vol. 1, pp. 409-412, Rhodes, Greece, 1997.

[61] S. Sharma, D. Ellis, S. Kajarekar, P. Jain and H. Hermansky, “Feature extraction using non-linear transformation for robust speech recognition on the Aurora database,” *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1117-1120, Istanbul, Turkey, 2000.

[62] T. Hain, P. C. Woodland, T. R. Niesler, and E. W. D. Whittaker, “The 1998 HTK System for Transcription of Conversational Telephone Speech,” *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, pp. 57-60, March 1999.

[63] Dusan Macho, *et al.*,”Evaluation of a Noise-robust DSR Front-end on Aurora Databases”, *Proceedings of International Conference on Speech and Language Processing*, pp. 17-20, Denver, Colorado, USA, September 2002.

[64] “ETSI ES 202 050 v1.1.1 Speech Processing, Transmission and Quality aspects (STQ); Distributed Speech Recognition; Advanced Front-end Feature Extraction Algorithm; Compression Algorithms,” ETSI, April 2002.

[65] A. Agarwal and Y. M. Cheng, “Two-stage Mel-Warped Wiener Filter for Robust Speech Recognition,” *Proceedings of Automatic Speech Recognition and Understanding Workshop*, Keystone, Colorado, USA, December 1999.

[66] H. M. Teager, “Some Observations on Oral Air Flow During Phonation,” *IEEE Transactions on Speech and Audio Processing*, October 1980.

[67] D. Macho and Y. M. Cheng, “SNR-dependent Waveform Processing for Robust Speech Recognition”, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, USA, May 2001.

[68] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, John Wiley & Sons, Inc., New York, USA, 1996.

[69] L. Mauuary, “Blind Equalization in the Cepstral Domain for Robust Telephone based Speech Recognition”, *Proceedings of IX European Signal Processing Conference*, vol. 1, pp. 359-363, Rhodes, Greece, September 1998.

[70] G. Hirsch, “Experimental Framework for the Performance Evaluation of Speech Recognition Front-ends on a Large Vocabulary Task,” ETSI STQ-Aurora DSR Working Group, June 2001.

[71] D. Paul and B. Necioglu, “The Lincoln Large-Vocabulary Stack-Decoder HMM CSR,” *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Minneapolis, Minnesota, USA, pp. 660-663, 1993.

[72] D. Pallett, J. G. Fiscus, W. M. Fisher, and J. S. Garofolo, “Benchmark Tests for the DARPA Spoken Language Program,” *Proceedings from the Human Language Technology Conference*, Merrill Lynch Conference Center, Princeton, New Jersey, USA, March 1993.

[73] G. Hirsch, “Experimental Framework for the Performance Evaluation of Speech Recognition Front-ends on a Large Vocabulary Task Version 2.0,” *ETSI STQ-Aurora DSR Working Group*, November 2002.

[74] Recommendation P.341 — Transmission characteristics for wideband (150-7000 Hz) digital hands-free telephony terminals, International Telecommunication Union (ITU), Geneva, Switzerland, February 1998.

[75] “The CMU Pronouncing Dictionary,” *http://www.speech.cs.cmu.edu/cgi-bin/cmudict*, Speech at Carnegie Mellon University, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 2001.

[76] A. Stolcke, “The SRI Language Modeling Toolkit,” *http://www.speech.sri.com/projects/srilm/*, Speech Technology and Research Laboratory, SRI International, Menlo Park, California, USA, July 2001.

[77] N. Parihar and J. Picone, “The Aurora Evaluations,” *http://www.isip.msstate.edu/projects/aurora*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, USA, July 2001.

[78] “Aurora Database”, *http://www.elda.fr/catalog.html*, European Language Resources Association, France, 2003.

[79] N. Deshmukh, A. Ganapathiraju, J. Hamaker, J. Picone and M. Ordowski, “A Public Domain Speech-to-Text System,” *Proceedings of the 6th European Conference on Speech Communication and Technology*, vol. 5, pp. 2127-2130, Budapest, Hungary, September 1999.

[80] R. Sundaram, A. Ganapathiraju, J. Hamaker and J. Picone, “ISIP 2000 Conversational Speech Evaluation System,” presented at *the Speech Transcription Workshop*, College Park, Maryland, USA, May 2000.

[81] R. Sundaram, J. Hamaker, and J. Picone, “TWISTER: The ISIP 2001 Conversational Speech Evaluation System,” presented at *the Speech Transcription Workshop*, Linthicum Heights, Maryland, USA, May 2001.

[82] B. George, B. Necioglu, J. Picone, G. Shuttic, and R. Sundaram, “The 2000 NRL Evaluation for Recognition of Speech in Noisy Environments,” presented at the SPINE Workshop, Naval Research Laboratory, Alexandria, Virginia, USA, October 2000.

[83] H. Murveit, P. Monaco, V. Digalakis and J. Butzberger, “Techniques to Achieve an Accurate Real-Time Large-Vocabulary Speech Recognition System,” *Proceedings of the ARPA Human Language Technology Workshop*, pp. 368-373, Austin, Texas, USA, March 1995.

[84] L. Lamel and G. Adda, “On Designing Pronunciation Lexicons for Large Vocabulary Continuous Speech Recognition,” *Proceedings of the International Conference on Speech and Language Processing*, pp. 6-9, Philadelphia, Pennsylvania, USA, October 1996.

[85] H. Ney and S. Ortmanns, “Dynamic Programming Search for Continuous Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 1, no. 5, September 1999.

[86] N. Deshmukh, A. Ganapathiraju and J. Picone, “Hierarchical Search for Large Vocabulary Conversational Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 84-107, September 1999.

[87] T. Kamm, G. Andreou and J. Cohen, “Vocal Tract Normalization in Speech Recognition Compensating for Systematic Speaker Variability,” *Proceedings of the 15th Annual Speech Research Symposium*, Johns Hopkins University, Baltimore, Maryland, USA, pp. 175-178, June 1995.

[88] T. Hain, P.C. Woodland, T. R. Niesler, and E. W. D. Whittaker, “The 1998 HTK System for Transcription of Conversational Telephone Speech,” *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, pp. 57-60, March 1999.

[89] Y. Hao and D. Fang, “Speech Recognition Using Speaker Adaptation by System Parameter Transformation,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, part 1, pp. 63-68, January 1994.

[90] R. Haeb-Umbach, H. Ney. “Linear Discriminant Analysis for Improved Large Vocabulary Continuous Speech Recognition,” Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, San Francisco, California, USA, vol. 1, pp. 13-16, March 1992.

[91] R. O. Duda, P.E. Hart, D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York City, New York, USA, 2001.

[92] N. Deshmukh, A. Ganapathiraju and J. Picone, “Hierarchical Search for Large Vocabulary Conversational Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 84-107, September 1999.

[93] S. J. Young, J. J. Odell, and P. C. Woodland, “Tree-based State Tying For High Accuracy Acoustic Modelling, *Proceedings of the ARPA Workshop on Human Language Technology*, Plainsboro, New Jersey, USA, pp. 286-291, March 1994.

[94] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, “Large Vocabulary Continuous Speech Recognition using HTK,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Adelaide, Australia, pp. II/125-II/128, April 1994.

[95] K. Beulen, and H. Ney, “Automatic Question Generation for Decision Tree Based State Tying,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 805-808, Seattle, Washington, USA, April 1998.

[96] W. Reichl, and W. Chou, "Decision tree state tying based on segmental clustering for acoustic modeling," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 801-804, Seattle, WA, USA, April 1998.

[97] L. Welling, S. Kanthak, and H. Ney, "Improved Methods for Vocal Tract Normalization," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 761-764, Phoenix, Arizona, USA, March 1999.

[98] W. Reichl, and W. Chiao, "Unified Approach of Incorporating General Features in Decision Tree Based Acoustic Modeling," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 573-576, Phoenix, Arizona, USA, March 1999.

[99] "Speech Recognition Scoring Toolkit (SCTK) Version 1.2c," *http://www.nist.gov/speech/tools/index.htm*, Speech Group, NIST, USA, January 2001.

[100] "Benchmark Tests, Matched Pairs Sentence-Segment Word Error (MAPSSWE)," *http://www.nist.gov/speech/tests/sigtests/mapsswe.htm*, Speech Group, National Institute for Standards and Technology, Gaithersburg, Maryland, USA, January 2001.

[101] R. Sundaram, "Effects of Transcription Errors on Supervised Learning in Speech Recognition," M.S. Dissertation, Institute for Signal and Information Processing, Mississippi State University, August 2003.

[102] P. Price, W. Fisher, J. Bernstein, and D. Pallett, "The DARPA 1000-Word Resource Management Database for continuous speech recognition," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 651-654, 1988.

[103] C. J. Leggetter, *Improved Acoustic Modeling for HMMs using Linear Transformations*, Ph. D. Thesis, Cambridge University, 1996.

[104] V. Valtchev, *Discriminative Methods in HMM-based Speech Recognition*, Ph. D. Thesis, University of Cambridge, UK, 1995.

[105] E. McDermott, *Discriminative Training for Speech Recognition*, Ph. D. dissertation, Waseda University, Japan, 1997.

[106] P. Woodland and D. Povey, "Very Large Scale MMIE Training for Conversational Telephone Speech Recognition," *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, MD, USA, May 2000.

[107] ITU Recommendation P.830, Subjective Performance Assessment Telephone Band Wideband Digital Codecs, February 1996.