APPLICATIONS OF LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION TO FATIGUE DETECTION

By

Sridhar Raghavan

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

August 2006

APPLICATIONS OF LARGE VOCABULARY CONTINUOUS SPEECH

RECOGNITION TO FATIGUE DETECTION

By

Sridhar Raghavan

Approved:

_____	_____
Dr. Joseph Picone Professor of Electrical and Computer Engineering (Major Advisor and Director of Thesis)	Dr. Georgios Lazarou Assistant Professor of Electrical and Computer Engineering (Committee Member)
_____	_____
Dr. Julie Baca Research Professor, Center for Advanced Vehicular Systems (Committee Member)	Dr. Nicholas H.Younan Professor of Electrical and Computer Engineering (Graduate Coordinator)
_____	
Dr. Roger L. King Associate Dean for Research and Graduate Studies	

Name: Sridhar Raghavan

Date of Degree: August 5, 2006

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Joseph Picone

Title of Study: APPLICATIONS OF LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION TO FATIGUE DETECTION

Pages in Study: 68

Candidate for Degree of Master of Science

Applications of speech recognition have evolved in recent years from simple transcription tasks to metadata analysis. This thesis explores the use of speech recognition for automated fatigue detection. The fatigue detection system relies on accurate phonetic alignments from a speech recognition system. The main challenge addressed in this thesis was to make the process of phonetic alignment using speech recognition robust to out of vocabulary words. This requirement was achieved by incorporating confidence measures, which significantly reduce false positives in speech recognition output. This allowed the performance of the fatigue detection system to match the results of other cognitive tests based on the Sleep Onset Latency (SOL) and Sleep, Activity, Fatigue, and Task Effectiveness (SAFTE). Confidence measures reduced the squared error between voice-based fatigue prediction and SAFTE by 20% when 67.1% of the words in the test set were out of vocabulary words.

DEDICATION

I would like to dedicate this work to my parents Kalpana and Raghavan, and to my sister Kavitha.

ACKNOWLEDGEMENTS

First of all I wish to express my gratitude to my major advisor Dr. Joseph Picone, who has been a tremendous mentor and teacher. This thesis would not have been possible without his constant encouragement and support. His constructive criticism and passion for perfection has had a profound impact in my life. I have always enjoyed his way of explaining things in a simple and elegant manner. I have been very fortunate to have worked under him as a graduate assistant.

I was new to the field of speech when I joined Mississippi State University (MS State). With the support and guidance I received from my colleagues I began to enjoy life as a speech researcher. I would like to thank Naveen Parihar and Hualin Gao who introduced me to the art of speech recognition. I would also like to thank Theban Stanley for his continual support. I extend my thanks to Dr. Hal Greeley and his colleagues at Creare Inc. who funded this project. Creare, Inc. provided fatigue data and experimental results that were critical to the success of this thesis.

I would like to thank all my friends in the Intelligent Electronic Systems (IES) program who have made my stay at MS State memorable. Finally, I would like to thank my roommates and friends who made me feel at home and took my mind off work whenever I needed it.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Non-intrusive fatigue assessment systems are crucially needed to successfully monitor the level of alertness of all personnel during critical mission or life-threatening activities. This thesis explored the use of automatic speech recognition (ASR) to detect fatigue from voice. There are numerous challenges which have to be overcome in order to have reliable fatigue detection systems based on voice. However, advances in speech recognition technology have made it possible to obtain good performance even in noisy environments, and hence, the technology has found widespread application in recent years.

1.1 Using Voice to Detect Fatigue

Applications of speech recognition have grown from simple speech to text conversion to other more challenging tasks. A relatively new application using voice is in the field of cognitive analysis [1]. The main goal is to detect the mental preparedness of a worker before critical missions, based on cognitive measures such as fatigue. Speech is one attribute of human behavior that can be used to measure fatigue. People working in stressful environments such as military and aviation are more susceptible to fatigue than

others, and accidents by such workers are often fatal. Complex instrumentation often creates cognitive overload and places a greater demand on the crew to be vigilant [1].

A prescribed remedy for fatigue is sleep. Roher [2] studied the effect of sleep on fatigue and determined that the quality of sleep is more important than the number of hours of sleep. This makes the task of monitoring fatigue even more challenging. There is no well-accepted non-intrusive technique to measure quality of sleep. This thesis explored the potential for using voice to perform real-time fatigue detection.

Several studies [3][4] show that voice is sensitive to fatigue. Typical correlates of fatigue include decreases in fundamental frequency and increases in word duration as the fatigue level increases. The goal of this thesis is to use a speech recognition system to detect such temporal and spectral variations. Greeley, et al. [5] have found that certain phones in human speech show temporal and spectral variations as a function of speaker's level of fatigue. Mel-Frequency Cepstral Coefficients (MFCCs) [6] are a good representation of the temporal and spectral characteristics of the speech signal. By analyzing MFCC features it was found that certain phones show more correlation with fatigue than others.

Greeley, et al. [5] computed correlation coefficients between the active and fatigued features of a single speaker. Experiments were conducted on data that was time stamped in order to asses the variation in the feature vectors as the speakers became increasingly fatigued. The correlation metric computed in this manner was compared with the Sleep Onset Latency (SOL) test which is considered the gold standard for fatigue analysis. An important observation was that the correlation coefficient varied

systematically as the subjects became more and more fatigued, and the correlation metric matched the SOL metric. A more detailed description of these experiments is presented in Chapter II.

1.2 Using Automatic Speech Recognition for Fatigue Analysis

Voice has been successfully used in many applications other than simple speech to text conversion. Applications involving speaker verification [7], deceit detection [8], reading tutors [9], automatic language recognition [10] and translation [9] are actively being developed. These applications share core technology based on a statistical approach to speech recognition. This thesis explores three families of techniques for detecting fatigue as part of a set of pilot experiments. The three approaches are speaker verification, word spotting, and ASR. It was determined that an ASR approach was most promising. Each of these approaches are briefly described below.

Speaker verification is the task of verifying a subject's authenticity based on his or her voice characteristics. This process entails two phases: enrollment and verification. During enrollment a speaker model is built using a subject's speech. During verification, this speech is used as a template to verify the authenticity of the speaker [7][11]. A speaker verification system was used as a primitive change detection system to determine whether systematic variations in the long-term statistics of the speech signal due to fatigue could be modeled using a classic pattern recognition paradigm. The results from this approach were not promising since a clear variation in the likelihood ratio scores between the speakers and the models as a function of fatigue was not found.

A second approach, word spotting, attempts to identify the occurrence of specific word instances in a speech file [12]. A word spotting system is built by training word models corresponding to the required keywords and training a garbage model on all other words. A garbage model is built by labeling all non-keywords in the transcription by the same token. The approach taken in this thesis for decoding used loop grammar; only the words of interest along with the garbage token were present in the grammar. This grammar guided the ASR system to either output the keywords, if present in the utterance, or output only the garbage token. This approach is attractive because there are fewer constraints on the words uttered by the speaker, and the lexicon can contain only the most frequently used words for a particular domain. Result from this approach showed a high number of false alarms, and hence this approach was not pursued.

A third approach entailed use of a large vocabulary continuous speech recognition (LVCSR) system to convert continuous speech to text. An LVCSR system can also cope with pronunciation variations and ambient noise to a certain degree, because it is based on statistical pattern classification algorithms that can learn the pronunciation variations and the noise characteristics from the data. A description of the basic blocks of an LVCSR system is given in Section 1.3. An LVCSR system was used to locate phoneme-like units, referred to as phones, in the incoming audio stream. The fatigue software then extracts the required phone from the decoded output, and in turn extracts the corresponding MFCC vectors from the feature stream. These features are then used by the fatigue analysis system to obtain a fatigue prediction estimate based on the correlation approach described in Chapter II.

Of the three approaches described above, the LVCSR approach was the most promising, and hence was the focus of this thesis. Though word spotting offered a less constrained user interface, the high false alarm rate made the overall system unusable. Performance of the LVCSR approach was improved significantly through the use of confidence measures [13]. This is described in greater detail in 0 .

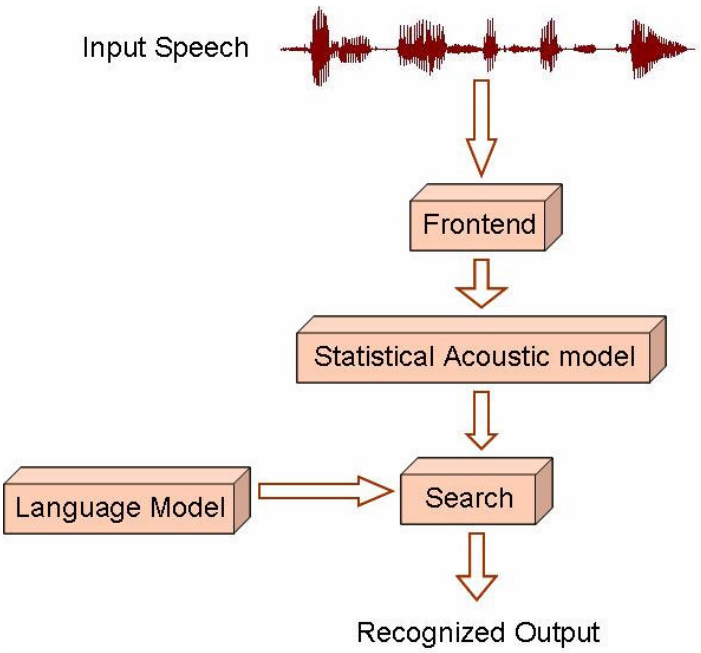

Figure 1 Basic components of a large vocabulary speech recognition system

1.3 Automatic Speech Recognition

The ASR system used in this thesis is a public domain LVCSR system developed by the Intelligent Electronic Systems (IES) program at Mississippi State University [14]. A speech recognition system, shown in Figure 1, consists of the following four blocks: feature extraction, language model, acoustic model and search. Feature extraction

converts the incoming signal to a stream of vectors, and typically uses an MFCC approach [6]. The acoustic model is a statistically trained model that learns the temporal and spectral characteristics of the speech signal. A language model is used to guide the recognizer with some a priori information about the language of interest in the application. The search block typically uses a Viterbi decoding algorithm [15] and finds the best path through the search space using language model and acoustic model probabilities.

The entire speech recognition framework can be represented using Bayes Rule as follows:

$$P(W \mid A) = \frac{P(A \mid W)P(W)}{P(A)} \ , \tag{1}$$

where P(W|A) is the probability of the word sequence given the acoustics. P(A|W) is the probability of the acoustics given the word sequence. P(W) is the probability of the word sequence which is given by the language model. P(A) is called the evidence and is the normalizing term. The evidence term can be neglected in the maximization process since it will remain constant for a particular data set, reducing the equation to the following:

$$\hat{W} = \underset{W}{\arg\max}\, P(A \mid W).P(W) \ . \tag{2}$$

The words can be divided into sub-units called phonemes. There are approximately 46 phones in English language. The acoustic models can model words or phonemes. Each model is represented with an HMM [16] that is implemented using a stochastic finite state automaton [16]. For limited vocabulary tasks, word models achieve high performance [16]. However, for large vocabularies that often consist of more than 100,000 words, word models are not practical [17] and hence phone models or cross-word triphone models are used.

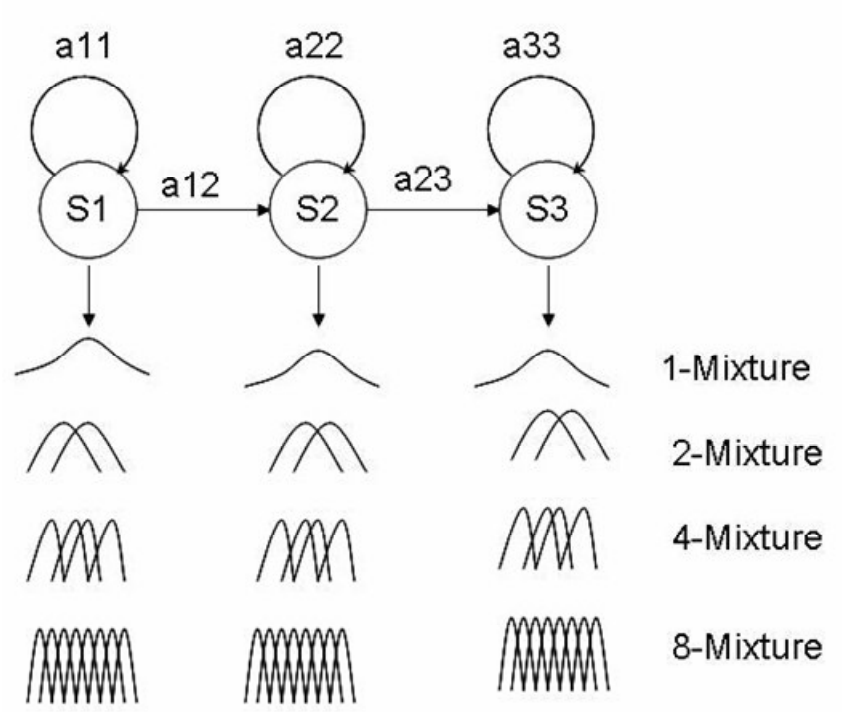

Figure 2 Three-state HMMs used to model phones in an ASR system

Phonetic models, often referred to as phone models, are typically used for large vocabulary systems. The articulation of words in human speech is context dependent, i.e.

the articulation of a particular sound depends on its surrounding sounds. It has been determined [18] that context-dependent phones that model the current phone in the context of the previous and next phone is a reasonable compromise between performance and complexity [17]. Context-independent and context-dependent phone models are often represented by a three-state HMM [16] as shown in Figure 2.

Each state in an HMM can be represented by a Gaussian mixture model (GMM) [15]. Recognition performance generally increases as the number of mixture components in the GMM increases [17] because an increase in the number of mixture components improves the ability of the GMM to model arbitrary distributions. This issue will be explored more extensively in Chapter V.

1.4 Thesis Contribution and Organization

The goal of this thesis is to automate fatigue detection using an ASR system. The concept of performing fatigue analysis on the speech signal using spectral information was developed by Greeley, *et al*. [5]. The fatigue analysis system requires accurate phone alignments to make accurate fatigue prediction. The system should be robust to ambient noise and should also account for out of vocabulary words. Robustness is a challenging issue [19] that is outside the scope of this work.

The main problem addressed in this thesis was detection of selected phones critical to the fatigue detection process with a high degree of confidence. Word posteriors computed from word graphs were used as a confidence estimate [13]. A confidence measure algorithm was implemented and the decoder output was annotated with the

confidence scores. Confidence measures were used to filter out false alarms from the one best output. The effectiveness of a confidence measure was first evaluated by examining the distribution of the measure for out of vocabulary (OOV) words. Models were trained on a selected vocabulary set and then tested on words from both within and outside the training vocabulary set. It was observed that the average confidence measure score for OOVs was 15% less than that for words spoken from within the training set. This was a positive indication that word posteriors could be used for discriminating potential false alarms from the output of an ASR system.

A secondary goal of this thesis was to improve the ability of the fatigue detection system to ignore OOVs in previously unseen data. Robustness to OOVs was achieved by using confidence measure. The relative performance of the baseline system was reduced from a total mean squared error of 0.1535 to 0.1487 with the use of confidence measures. The OOV error rate was 61.7%. It should be noted that only six test epochs were observed in this experiment, which explains why only a marginal improvement was achieved.

The discriminative power of confidence measures was analyzed by using a receiver operating characteristic (ROC) curve. The area under the ROC curve is an indication of the discriminative power of the classifier. An area of 0.5 indicates a random classifier while an area of 1 indicates an ideal classifier. The area under the ROC curve for the system that incorporated a confidence measure was 0.82, which indicates good discrimination. A suitable operating point or threshold had to be determined for

classification. A threshold of -75 was chosen since at that point the probability of false alarms was equal to the probability of true occurrences of words.

Fatigue experiments demonstrated that the Voice Correlation metric [5] matched more closely to the Sleep, Activity, Fatigue, and Task Effectiveness (SAFTE) model than the Sleep Onset Latency (SOL) model [5]. The total squared error between the normalized SOL and Voice Correlation metric was 0.33 while the total squared error between normalized SAFTE and Voice Correlation metric was only 0.029 . After using confidence measures it was determined that the total squared error between the Voice Correlation metric and SAFTE model decreased from 0.15 to 0.12, and this was observed when the test set had an OOV error rate of 61.7%.

This thesis is organized as follows. Chapter II describes the process of extracting MFCC features from a speech signal, and also gives a brief overview of the fatigue prediction technique found by Greeley, *et al*. [5]. Chapter III describes various ways an ASR system could be used for fatigue prediction, and introduces the concept of a confidence measure. 0 provides more details about the implementation of a confidence measure along with an example. Chapter V discusses various experiments run on the fatigue dataset. The thesis concludes with a discussion of future directions for this work in Chapter VI.

CHAPTER II

FATIGUE ANALYSIS USING ACOUSTIC FEATURES

This chapter begins with a description of how fatigue can manifest itself in the spectrum of a speech signal. This thesis refers to these manifestations as the acoustic correlates of fatigue. Research suggests that increases in fatigue correlate with a decrease in the fundamental frequency and an increase in the word duration [3]. Greeley, *et al.* [5] went one step further to analyze the changes in formants due to fatigue. Formant variations can be readily observed in the spectrogram. Development of automated techniques to extract fatigue cues was a major goal of the pioneering work performed by Greeley, *et al.* [5].

The development of such an automated system required annotated, or truth-marked, training data from subjects experiencing a range of fatigue symptoms. An initial data set, referred to as the Phase I data, was developed to further study this problem. After promising results were obtained on this data, a more extensive data set, referred to as the Phase II data, was collected to facilitate the development of ASR technology. The Phase II data was very noisy, and it was not possible to build generalized models using this data. This issue will be dealt with in greater detail in Chapter III. A third fatigue data set, referred to as the FAA data, was collected under clean recording conditions to evaluate the overall system described in this thesis. This chapter reviews the analyses

performed on these data sets and introduces the general approach to automated fatigue detection. Automated fatigue analysis attempts to model the acoustic correlates of fatigue in a speech signal. The first step in building an automated system is to convert the speech signal into a sequence of feature vectors. The most important attribute of a good feature especially for an ASR system is that the features should accurately model distinctions made by the human perceptual system. These are referred as perceptually-meaningful features. This thesis uses mel-frequency cepstral coefficients (MFCCs) [6] as features to capture temporal and spectral variations in the signal. This chapter discusses some results obtained from pilot experiments that were conducted by Greeley, *et al.* [5] on these features.

2.1 Acoustic Correlates of Fatigue in the Speech Signal

Studies on military aircrews operating B1 bombers showed that voice had a similar pattern to that of other cognitive measures of fatigue [3]. An automatic voice-based fatigue detection system should use fatigue cues present in the speech signal. In order to determine these fatigue cues, one needs to understand the changes that occur in human speech as a person becomes fatigued. Literature suggests that there is a spectral and temporal variation in the speech pattern as humans become increasingly fatigued. The spectral variation can be attributed to a change in the human sound production system, while the temporal variation is controlled by the brain and its explanation is beyond the scope of this thesis.

Sounds produced by humans can be represented as a convolution of the excitation signal and the vocal tract characteristics. The excitation can be modeled as either a periodic signal or noise. For voiced speech sounds, the excitation can be modeled as a periodic signal whose fundamental frequency is determined by the vibration of the vocal cords. The vocal cords close temporarily to increase the air pressure generated from the lungs, and they open when the pressure exceeds the resistance of the vocal cords. The vocal cords vibrate due to a combination of factors, including their elasticity, laryngeal muscle tension, and the Bernoulli effect [20]. The opening and closing continues as long as the lungs pump air through the vocal cords and into the oral cavity.

As discussed previously, fatigue affects the fundamental frequency. But fundamental frequency is a speaker-dependent parameter and hence will not be useful if one wants to build a speaker-independent fatigue detection system. On the other hand it is known that the formant frequencies of various phonemes do not vary significantly from person to person. This fact is exploited in most of the state-of-the-art speaker-independent ASR systems.

In the linear acoustics model of speech production [20], the vocal tract can be modeled as a time-varying digital filter. The resonances of this filter are referred to as formants. The frequencies of these formants can be used to identify sounds. These formants are determined by the overall shape, volume and length of the vocal tract. The shape of the filter is a combination of the vocal tract characteristics and the following factors:

1. Yielding walls – The walls of the vocal tract are not rigid, and hence they vibrate due to the variations in air pressure inside the vocal tract. As a result, the cross-sectional area of the vocal tract will change. The effect of yielding walls causes a shift in the formant frequencies. A slight increase in formant frequencies is observed, and this is more pronounced for lower formants [20]. There will also be a slight broadening of the formant bandwidths at the lower end of the spectrum.
2. Viscosity and thermal losses – Energy will be lost due to viscous friction between air and the walls of the vocal tract, and also due to heat conduction inside the vocal tract. The combined effect of these two results in a lowering of the formant frequencies [20]. The effect of this is more pronounced for frequencies beyond 3 to 4 kHz.
3. Lip radiation – The lips act as a short circuit termination to the vocal tract circuitry. In a circuit theory model of speech production, lip radiation is represented by a short circuit. This is not completely true in the case of the human vocal system because this means there will be no change in pressure for a change in volume velocity. A reasonable approximation of lip radiation is a small baffle with infinite extensions at the two ends [20]. The behavior of the lip radiation load affects the wave propagation in the vocal tract [20]. The losses influenced by the lip radiation load cause a slight decrease in the formant frequencies. This is again more pronounced at higher frequencies.

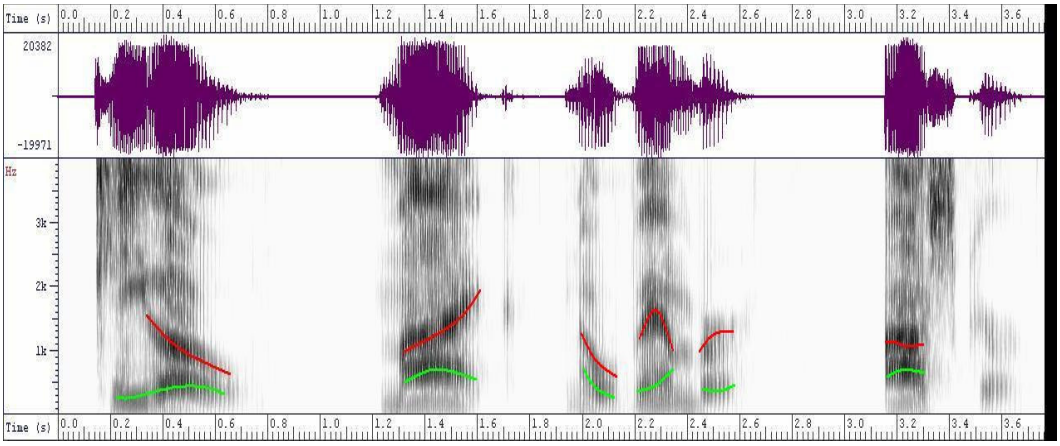

Figure 3.1 A waveform and spectrogram of a non-fatigued subject

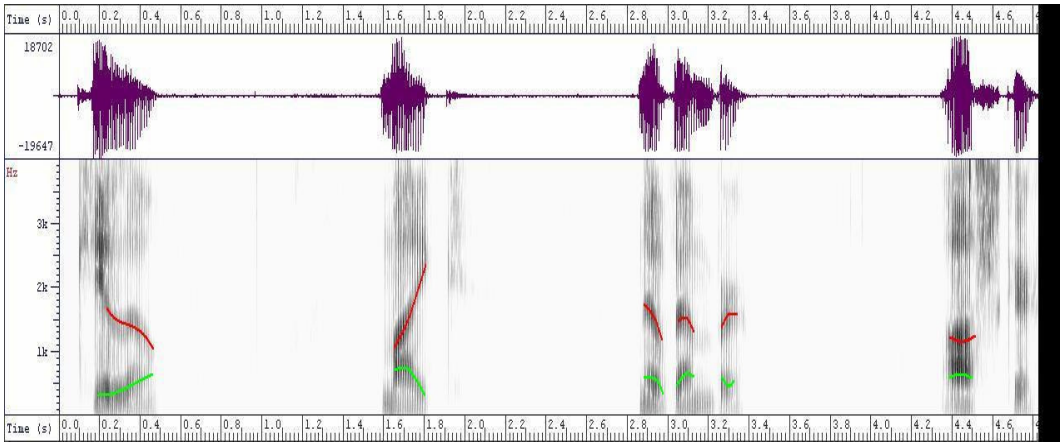

Figure 3.2 A waveform and spectrogram of a fatigued subject

Fatigue can have a direct influence on any of the above described factors, which are responsible for generating the formant frequencies in the vocal tract. Research suggests there is a variation in the pronunciation of phonemes due to artificially induced stress [21], but the study of the effects of fatigue on formants is still in its infancy. Therefore, Greeley, *et al.* [5] laid the foundation for the development of the fatigue detection system described in this study by analyzing the effects of fatigue on formant frequencies.

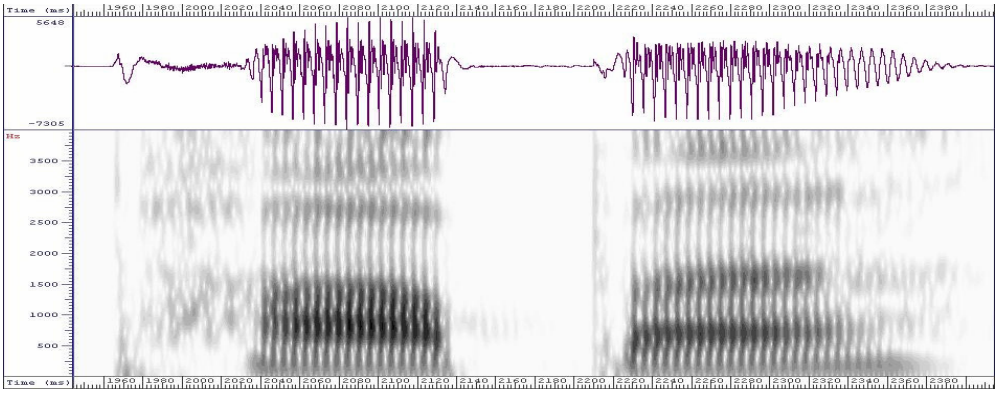

Figure 4.1 A waveform and spectrogram of a non-fatigued subject uttering the word "papa"

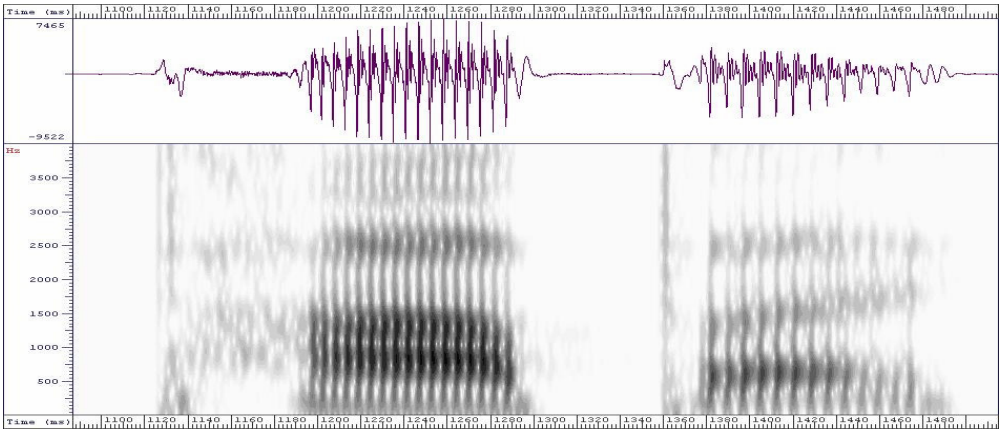

Figure 4.2 A waveform and spectrogram from a fatigued subject uttering the word "papa"

In order to study fatigue in speech, Greeley, *et al.* [5] designed and collected a small database known as the Phase I data. Ten subjects were asked to speak a list of sentences at regular time intervals. Since sleep deprivation is known to induce fatigue, the subjects were deprived of sleep throughout the recording period [2]. Some examples of sentences present in the Phase I list are as follows: "Lucy glanced at her watch and informed me that I have plenty of time to catch the ship." and "I took the journal away with me and read most of it in the car."

Figure 3 shows a comparison of subject's voice before and after fatigue induction. A clear variation in the formants can be observed for some of the sounds. For example, the most notable of the change is that the energy at various formants has decreased. A possible yielding wall effect can also be witnessed in the first and second formants .i.e. there is a slight increase in the formants in this region. This can be attributed to the softening of the vocal tract walls due to fatigue. Figure 4 provides a magnified view of the variations in the formant frequencies. The word "papa" was uttered before and after fatigue induction. The second formant in the fatigued utterance has shifted slightly upwards in frequency near the ends of words. The fourth formant in the fatigued utterance is so weak that one cannot even observe it in the spectrogram.

In order to confidently establish variations in formants as a function of fatigue, one needs to perform a thorough analysis of various sounds present in human speech. Greeley, *et al.* [5] have performed pioneering work in this area by conducting an extensive study that will be described in the next section. In later sections a technique using cepstral analysis [6] to capture the formant variations in the speech signal will be described.

2.2 Analysis from Phase I and Phase II data

Using the Phase I data, Greeley, *et al.* [5] conducted experiments to determine the relationship between formant frequencies of voiced sounds and fatigue. Ten volunteers were asked to speak sentences containing words from a set of 37 words. The recordings were made four times a day, before and after a night of sleep depravation [5]. Reaction

time was measured just before making the recordings, and sleep latency was measured to determine the general level of fatigue.

Reaction time was measured by a simple visual reaction time test. When a light goes on the subject moves his hand as quickly as possible. The hand movements were measured by an optical proximity sensor. Measuring sleep latency involves having the test subject lie on a bed in a quiet, darkened room and telling the subjects to fall asleep. The time that it takes them to fall asleep is measured by an electroencephalogram (EEG) [2].

Approximately 12,000 formant frequencies were analyzed, and 19 of them showed significant correlation with reaction time. Several showed good correlation with the sleep latency tests as shown in Table 1. The results from the table show that the formant frequencies are related to the subject's reaction time.

Table 1 Correlation between formant frequency and performance

Sound	F1	F2	F3	F4
	R (P) slope	R (P) slope	R (P) slope	R (P) slope
[o] clock	0.486 (0.001) +	0.339 (0.010) +	0.710 (0.001) +	0.565 (0.001) +
[^] upper	0.416 (0.001) +	0.352 (0.010) +	0.689 (0.001) +	0.680 (0.001) +
[ay] highly	0.356 (0.001) +	0.359 (0.001) +	0.332 (0.010) +	0.682 (0.001) +
[iy] keep	0.511 (0.001) -	0.241 (0.050) +	0.396 (0.001) +	0.228 (0.050) +
[m] matter	0.574 (0.001) -	0.567 (0.001) -	0.343 (0.010) -	0.118
[o] coughing	0.367 (0.001) +	0.071	0.487 (0.001) +	0.310 (0.010) +
[n] note	0.386 (0.001) -	0.114	0.071	0.000
[n] night	0.389 (0.001) -	0.095	0.095	0.192
[^] fuzzy	0.324 (0.010) +	0.187	0.388 (0.001) +	0.243 (0.050) +
[uw] two	0.360 (0.001) +	0.122	0.205	0.298 (0.010) +
[ae] chatter	0.359 (0.001) +	0.152	0.316 (0.010) +	0.351 (0.001) +
[ay] time	0.326 (0.010) +	0.045	0.326 (0.010) +	0.045
[ae] cabin	0.313 (0.010) +	0.105	0.310 (0.010) +	0.164
[y] yet	0.308 (0.010) -	0.045	0.210	0.152
[U] took	0.055	0.344 (0.01) +	0.705 (0.001) +	0.612 (0.001) +
[iy] serene	0.205	0.623 (0.001) -	0.182	0.071
[n] now	0.164	0.538 (0.001) +	0.576 (0.001) +	0.460 (0.001) +
[r] rather	0.036	0.032	0.310 (0.010) +	0.517 (0.001) +
[o] not	0.045	0.265 (0.050) -	0.164	0.109

Because of the promise shown by Phase I data, a more extensive data collection effort was undertaken, referred to as the Phase II database. The data was collected during a three-day military exercise. The recording instrument used was a Personal Digital Assistant (PDA), and the recordings were made in an outdoor environment. The twenty three participants were allowed sleep only in small fixed time intervals in order to induce fatigue. The subjects were asked to recite eight prewritten phrases and undergo reaction

testing at regular intervals. Reaction time testing was performed to measure the level of fatigue. Various studies on fatigue have identified reaction time as one of the main side effects of fatigue [1]. Reaction time can be measured easily and one such method is described at the beginning of this section. The recorded data was also time-stamped. The Phase II data had some drawbacks when used with an ASR system, and this issue will be dealt extensively in Chapter III.

2.3 Using MFCCs for Fatigue Analysis

Initial Phase I analysis confirmed a dependence between formant frequencies and fatigue. It was found that not all phonemes in human speech were affected equally by fatigue. This was shown in Table 1. It was necessary to analyze the formants of specific phones of interest. An ASR system was used to determine the phone alignments. The ASR system used standard feature vectors [6] for training and decoding. The standard features used with the ASR system captures the formant information present in the speech signal [6]. Therefore, further analysis to detect fatigue was performed in the feature domain.

Mathematically, the speech signal is a convolution of the excitation signal and the filter characteristic function (vocal tract response) in the time domain and a multiplication of the two in the frequency domain. It is possible to extract information about either component mentioned above using conventional signal processing techniques. For example, the spectral characteristics of the speech signal are obtained by taking a Fourier Transform and calculating the logarithm of the resulting amplitudes. This provides a

measure from which excitation and vocal tract response can be separated. Picone [6] covers feature extraction in greater detail.

The log magnitude spectrum is then transformed back to the time domain using a Discrete Fourier Transform. This process results in the calculation of a discrete number of coefficients called cepstral coefficients. Isolation of either the excitation or vocal tract response is accomplished by selection of required cepstral coefficients. With this, the entire human speech production process can be described by only a few cepstral coefficients [6].

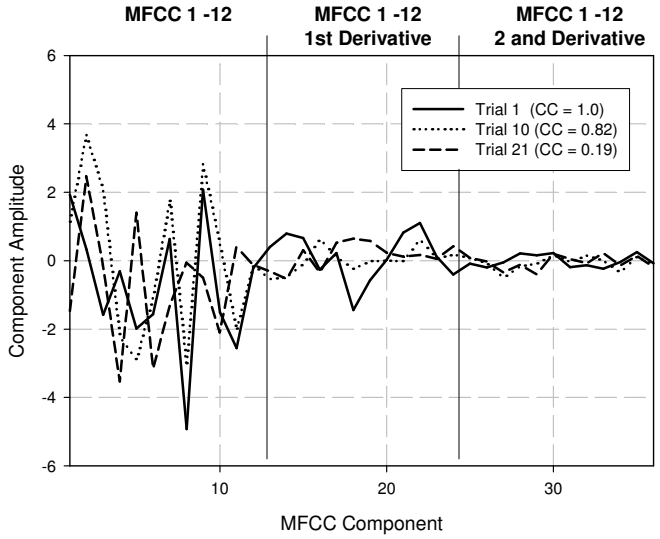

Figure 5 A comparison of three MFCC vectors observed over a four-day period

During the feature extraction process, the linear frequencies are mapped to a Mel frequency scale [6], because it models the perception of the human ear. This is essential if the features are used for ASR applications. For this reason the cepstral coefficients are referred to as the Mel Frequency Cepstral Coefficients (MFCCs). The standard feature

vector is comprised of 12 cepstral coefficients, along with their first and second time derivatives. Hence, the standard feature vector contains 36 coefficients. Also of interest is how the feature vectors changes as a function of the subject's level of fatigue. Figure 5 shows an example of how the MFCC vector changes over a four-day period of sleep restriction.

There is value in analyzing the MFCCs of different phonemes present in human speech, since it is known that formant variations are dependent on phonemes. The MFCCs for various sounds were analyzed, and the sounds that were most affected by fatigue were determined. The analysis was performed on the MFCCs of utterances recorded at different instants of time. The database for this analysis was collected by inducing speakers with fatigue. The process of collecting this data is described in previous sections.

The correlation was calculated between two sets of MFCC vectors. One set was obtained during the initial phase of recording and the other when the subject was fatigued. For example, the variations of the MFCCs were observed for the sound 't' over a four-day period of sleep restriction. The variation of the different components of the MFCC vector can be observed in Figure 5. There is an indication that the MFCC components change as the subject gets increasingly fatigued. A correlation metric can be used as a prediction metric to determine fatigue. The correlation can be computed between the MFCC vectors obtained during initial phase of recording and the MFCC vectors obtained during testing.

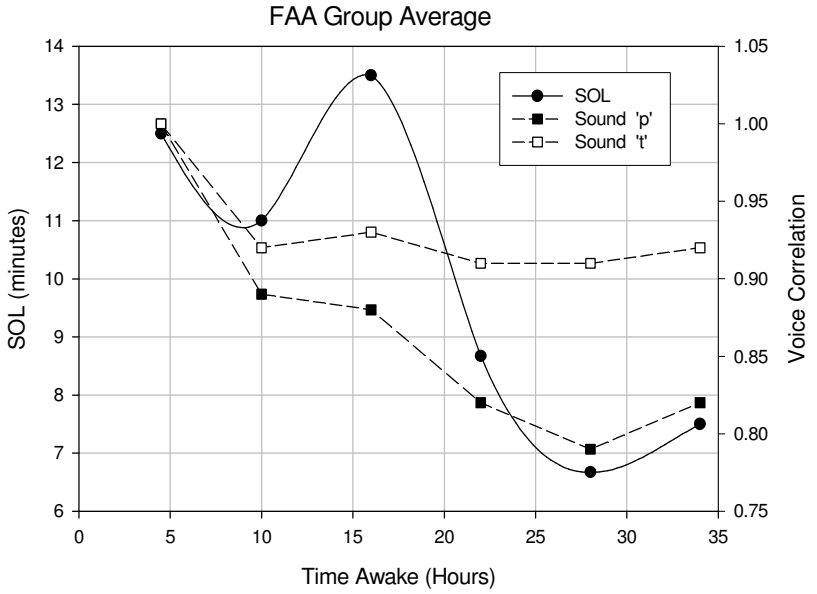

Figure 6 Change in the voice correlation metric for the sound 'p' and 't', along with the sleep onset latency observed at various time instants

As discussed in Chapter V, the ASR system did not give good performance with the Phase II data, as there were challenging issues such as ambient noise and disfluencies, and dealing with such challenges is outside the scope of this thesis. Another data set, known as the FAA data, was collected in a clean studio quality environment. This data set was used for ASR experiments described in the remainder of this thesis.

2.4 Fatigue Analysis Using the FAA data

During a 34-hour period of sleep deprivation, six non-medicated subjects were asked to recite a list of 31 words at six testing times (10:00 AM, 4:00 PM, 10:00 PM, 4:00 AM, 10:00 AM, and 4:00 PM) [22]. These testing times were selected to represent circadian high and low points in performance [2]. Also measured during these testing times was Sleep Onset Latency (SOL), which is the gold standard for sleepiness testing.

This test involves having the test subject lie on a bed in a quiet, darkened room and telling the subjects to fall asleep. The time that it takes them to fall asleep, as measured by an electroencephalogram (EEG), is the sleep onset latency (SOL) [2]. Between tests, subjects were allowed low arousal activities such as reading and watching TV.

Figure 6 shows the group average change in both SOL and our voice correlation metric for the sounds 'p' and 't' over the 34-hour testing period. The correlation coefficient between SOL and time awake is -0.825, and between voice correlation of sounds 'p' and 't' to time awake is -0.89, and -0.67 respectively. It was estimated that time awake accounts for 68%, 79%, and 45% of the variation of SOL, voice correlation of sounds 'p' and voice correlation of 't' respectively [5].

Circadian means "exhibiting periodicity in a 24-hour period." For example, our sleep cycle is considered to have a circadian trend (.i.e., humans sleep better at night than during the day.) All three metrics show a circadian peak at 16 hours. However, the SOL peak is significantly larger than the voice metric peak. Fatigue levels were observed to be higher during normal sleep hours than at regular working hours, which explained the circadian trend. The circadian pattern has been observed in many alertness experiments [23][24][25]. This difference in circadian sensitivity tends to reduce a correlation coefficient-based quantitative comparison.

In order to automate the task of fatigue detection, an approach is described in the following chapters use an ASR system. The ASR system is used to determine phone hypotheses, and these hypotheses are post-processed to predict fatigue. Fatigue prediction was accomplished by using the software developed by Greeley, *et al.* [5]. The fatigue

software extracted the required MFCC vectors from the input feature set. The selected MFCC vectors were correlated with MFCC vectors that were collected when the speaker was less fatigued. The computed correlation metric was used to predict the fatigue level of the speaker.

CHAPTER III

USING AN LVCSR SYSTEM FOR FATIGUE DETECTION

Fatigue detection from speech is one example of a growing application area for speech processing systems known as metadata extraction [26][27]. This chapter discusses various approaches to the task of fatigue detection using an ASR system. Several approaches to fatigue detection were evaluated including speaker verification, word spotting, and LVCSR. Only the LVCSR approach was found to be effective.

The chapter also discusses the various challenges involved in applying an LVCSR system to this task, including how to improve robustness. A confidence measure was used to increase the reliability of the phonetic alignments provided to the fatigue detection system by making the system more robust to OOVs [5]. The confidence measure algorithm will be described in 0.

3.1 Motivation

Using voice to detect fatigue is a challenging task and that requires large amounts of data and sophisticated pattern recognition techniques. An LVCSR system is essentially an application of machine learning that is capable of processing huge data sets that often comprise thousands of hours of speech. There are many ways such a system can be applied to fatigue detection, including the three approaches as discussed in Chapter I.

Through experimentation, it was determined that the phonetic labeling approach was most effective [5]. It was determined that only a small subset of the phonemes could be useful for fatigue detection since the spectral and temporal characteristics of these phones varied significantly as the subject became increasingly fatigued [5].

Chapter II describes the details of the fatigue detection approach used by Greeley, *et al.* [5]. An LVCSR system automates the process by providing phonetic alignments for a subject's utterance. Phonetic alignments can be generated in two ways, and these two techniques will be discussed in the chapter. There are several challenges that have to be overcome to obtain accurate phonetic alignments. For example, obtaining accurate phonetic alignments for unseen speakers under noisy conditions and in the presence of OOVs are both difficult problems even for the most advanced systems [27].

While it is reasonable to focus on a subset of the phones for fatigue detection, one cannot assume these phones will be recognized perfectly. The system described in this chapter includes a metric that specifies the confidence value for each hypothesized phone. This confidence metric can be used to filter out false hypotheses. In this chapter let us look at alternate approaches to this problem that share a common Gaussian mixture model (GMM).

3.2 Applications of Speaker Verification

Various approaches to detect fatigue were analyzed using a public domain ASR toolkit [14]. The first approach was based on a GMM-based speaker verification system. In general, a speaker verification system can use one of two modeling techniques. The

simplest technique is to use a template model [28]. This approach works well for highly constrained applications, but is not as popular for more advanced applications such as text-independent speaker verification. Campbell has presented some good examples of template modeling techniques in his tutorial [28].

The second approach is based on stochastic models, and GMM is one such stochastic approach. During the verification or pattern matching phase, the likelihood of the observation given the speaker model is computed. The observation consists of a sequence of random vectors whose conditional density (speaker model) is estimated during training, and this is achieved by using a set of training vectors. This estimated density is represented by a mixture of Gaussians (GMM). Using the estimated GMM, one can compute the probability of an observation given the claimed model as shown in equation (3).

$$\begin{aligned} \text{utterance score} &= \log P(X \mid \text{speaker model}) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log P(x_i \mid \text{speaker model}) \quad , \end{aligned} \tag{3}$$

N corresponds to the number of observation vectors. A decision to accept or reject a speaker is made based on the overall utterance score. For speaker verification, one can use a better decision making strategy by using impostor models [28]. This thesis did not use impostor models for our fatigue experiments, and hence this technique will not be discussed in this thesis.

A speaker verification system such as the one described above can be used to model the long-term speech characteristics of a speaker. The system builds a model for

each individual speaker (in this case, one speaker) and then computes the likelihood, defined as the conditional probability of the acoustic data given the speaker model, of a test subject's data [7]. By using an empirically determined threshold, it is possible to discriminate between a true speaker and an impostor. The basic structure of a speaker verification system is shown in Figure 7. The system used in this thesis was based on a speech recognition system described in [14].

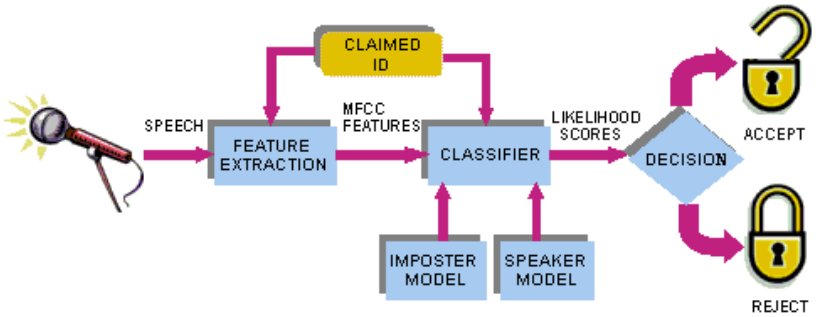

Figure 7 Basic speaker verification system architecture

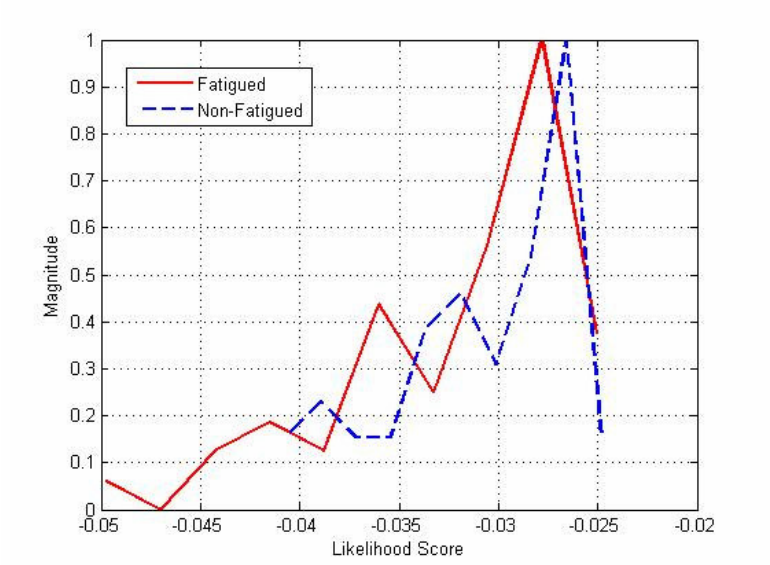

Figure 8 Distribution of the likelihood scores of fatigued and non-fatigued speakers

A speaker verification system was used to build speaker independent fatigue models. Preliminary experiments were conducted using the FAA data [5]. Models were created using data that was obtained during the initial phase of recording. The FAA data consists of six recording times spread over 36 hours. The training data contained a subset of utterances recorded during the first recording phase. The remaining subset of the first recording phase, and the data recorded from Phase 2 through Phase 6 were used for testing. The distributions of the likelihood scores for fatigued and non-fatigued utterances are shown in Figure 8.

There was very little difference in the likelihood scores, and hence setting a threshold to discriminate fatigued and non-fatigued utterance was impossible. A Detection Error Trade-off (DET) curve was also plotted to judge the usefulness of this system. From the DET curve shown in Figure 9 one can see that the system performs

with an EER of 47%, which is unacceptable. The discouraging results could be attributed to two main reasons: 1) data was insufficient for training which led to poor acoustic models; and 2) not all phonemes in human speech are affected by fatigue in the same manner [5].

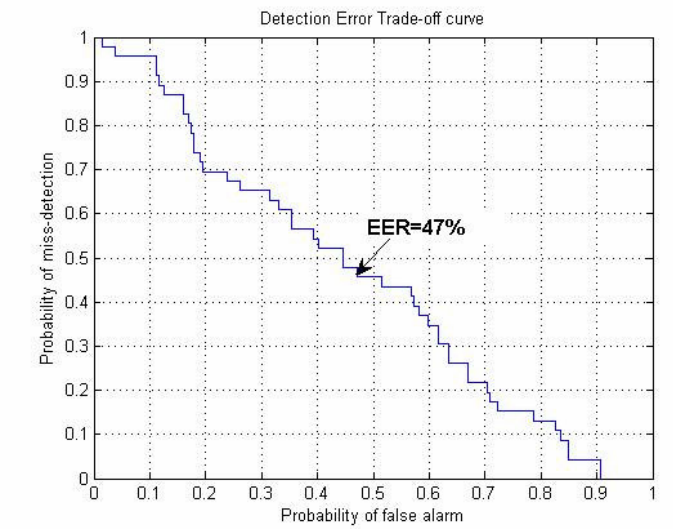

Figure 9 Detection error trade-off curve for a speaker verification-based fatigue detection system

3.3 Word Spotting

Greeley, *et al* [5] found that certain phonemes show more variations in the spectral domain due to fatigue than others. There was a need for a system that could spot these phones of interest from the utterance. The spotting of specific phonemes was made easier by spotting the words that contained the phones of interest. An LVCSR system with a loop grammar [16] was used. A loop grammar allows any word in the lexicon to be followed by any other word. The loop grammar for the word spotter contained the target words of interest defined by the system designer.

The loop grammar also contained a garbage model token. A garbage model is trained by labeling all the words in the transcription by a single token. The garbage model represents the most likely choice in the final hypothesis if none of the other words in the grammar is chosen as the best choice. For example, if one wants to spot the word "tall" in the sentence "The tall women ate my donut," the output from the word spotter can look something like this: "<starting silence> garbage tall garbage garbage garbage garbage <ending silence>". The advantage of using a word spotting system is that one can use a fixed lexicon with words specific to the domain and not worry about OOVs appearing in the hypothesis for the test utterance.

The word spotting system designed in this manner spotted the required words with reasonable accuracy, but also inserted many false alarms in the final hypothesis. This was unacceptable because it forced the fatigue software to perform analysis on data that was labeled incorrectly. For example, an experiment to spot the word "keep" on the FAA data produced a miss-recognition rate of 4%, but the false-alarm rate was as high as 82%. Such a high false alarm rate produces unacceptable performance for the fatigue detection module that post-processes this output.

The acoustic models for this experiment were trained using 80% of the FAA data while the remaining 20% was used for testing. The WER of the system was 12%. Another practical problem with this word spotting system was that retraining was required whenever new words were added to the list of keywords. Due to the drawbacks mentioned above, another approach was investigated and was found to be more robust.

3.4 Large Vocabulary Speech Recognition

An LVCSR system was used to provide phonetic alignments to the fatigue detection system. There are two methods to obtaining alignments for an utterance. The first method is to perform a forced alignment of the reference transcription with the utterance. Forced alignment uses a Viterbi algorithm [16] with transcription data and MFCCs as inputs. Forced alignment is much simpler than conventional decoding since the reference transcription is already known. It is called "forced" because the best path is forced to contain the required reference word sequence. The output of the forced alignment process is time-aligned phonetic labels for the input utterance. Usually, a forced alignment technique is used to retrain models during parameter re-estimation [16]. For fatigue analysis, this approach had a drawback, which is that the subject had to speak predetermined phrases. Hence, this was not practical for operational environments.

The second approach was to perform one-best decoding [16] and obtain phone alignments, but the accuracy of such a system relies heavily on the attributes of the speech data. The recognition performance on the Phase II data was only 50% when a bigram language model [17] was used. On the FAA data, which was relatively noise-free data, the WER was 12%. On closed-loop tests, in which one evaluates on the training data, the WER was 0.1%. Such low error rates are expected on closed-loop tests.

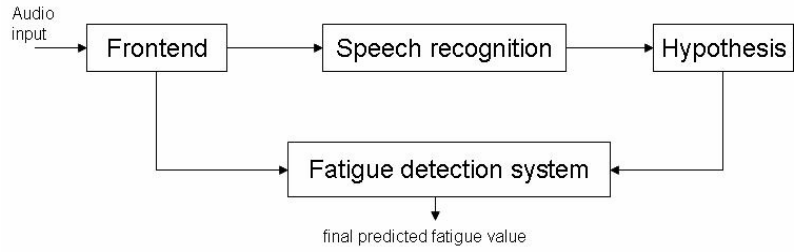

Figure 10 Integration of the fatigue detection system with an ASR system

Initial fatigue experiments were based on the closed-loop experiments. A block diagram showing the integration of the ASR system with the fatigue detection system is shown in Figure 10. There are three main challenges that must be overcome to obtain accurate phonetic alignments using one-best decoding approach. The main obstacles are enumerated in the proceeding subsections.

3.4.1 Out-of-vocabulary Words

The test subject should not be restricted to a specific vocabulary set. The goal of this thesis was to make the system as easy to use as possible, and hence, minimize any constraints on a user's speech. In fact, the test subject should not be aware of the fatigue detection system, as it should run in the background and serve as an automatic alerting system. Manual addition of new words is not a viable option, particularly when N-gram language models are used. This creates additional complexities because language model probabilities need to be carefully balanced to take into account new words, and acoustic models need to be generated automatically.

3.4.2 Unseen Speakers in the Test Data

The system has to work for any speaker, which is a characteristic one refers to as speaker independent. The enrollment time must be minimized and preferably be restricted to establishing a baseline non-fatigued state. Acoustic models should not need to be retrained for each speaker, though some adaptation can be allowed. For example, the fatigue system should be able to detect fatigue levels of different pilots without having any prior information about the voice characteristics of every pilot that flies the aircraft.

3.4.3 Noise and Channel Characteristics

This is the most challenging of the three main obstacles for building a robust ASR system. It is very unlikely that noise conditions will remain same during training and testing. Also the system can be deployed onboard an aircraft, over a telephone network or over the Internet, so it is likely there will be a significant mismatch between the training and testing acoustic environments. For example, it is very difficult for an ASR system to decode an utterance spoken from a cell phone in a subway when the models were trained on utterances that were spoken over a land line phone from a typical business office environment. Though such problems have been heavily researched, this problem remains an active area of research and is beyond the scope of this thesis. This thesis describes a method to combat the first of the three obstacles – OOVs. This was achieved by using confidence measures to prune away less probable hypotheses. The confidence measure algorithm is discussed in 0.

CHAPTER IV

CONFIDENCE MEASURES AND WORD POSTERIORS

The fatigue software developed by Greeley, *et al.* [5] assumes the phonetic alignments to be accurate. As described in Chapter III, this is not a reasonable assumption as one has to deal with three practical problems mentioned in Chapter III. To make the system robust to OOVs, a confidence score was annotated to every word in the output hypothesis. A confidence score is a numeric value that represents the confidence the ASR system has that a particular word hypothesis is correct. This metric will be used by the fatigue detection software to focus on words with high confidence, and hence eliminate false alarms that negatively impact the accuracy of fatigue detection.

4.1 Word Posteriors as a Confidence Measure

The initial approach was to use the likelihood score of the words in the output hypothesis as a confidence measure. After analyzing the likelihood score histograms for correct and false words during a recognition experiment, it was found that the likelihood score from the ASR's output could not be used directly as a confidence measure. The variation in the likelihood score was random, and consequently, a reliable threshold could not be set. Figure 11 shows the likelihood score histogram comparison for output words when the WER was 0% and when there were errors in the hypothesis (i.e. WER

of 42.9%). From Figure 11 it can be observed that the likelihood scores were insensitive to the errors and could not be used as a measure of confidence.

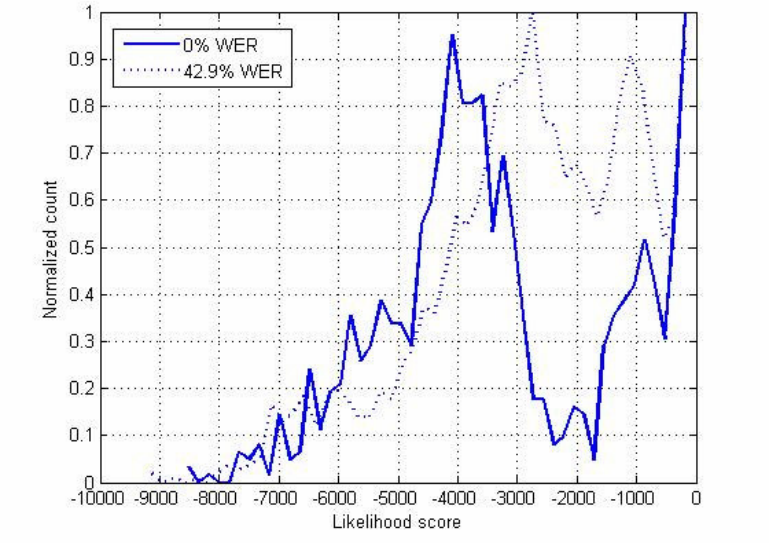

Figure 11 Likelihood score distribution of the words in the final hypothesis

A confidence measure can be computed by taking into account various probable word sequences in the search space. This can be done by either using N-best lists or word graphs [13]. This thesis used a confidence measure computed from word graphs because these have been shown to yield better results compared to the N-best list technique [13]. Specifically, the word posteriors computed from word graphs [13] were used.

Mangu, *et al.*, [13] defines a word posterior as "the sum of the posterior probabilities of all word sequences of which the word is a part." If the WER on the data is poor, then the word posteriors may not be a good confidence estimate [29], because the posteriors are overestimated as the words in the word graph are not the full set of possible words. In the case of a poor WER, the word graph will contain many wrong word

sequences. In such a case, the depth of the word graph becomes a critical factor in determining the effectiveness of using the confidence measure. The depth of the word graph can be adjusted by varying parameters such as MAPMI threshold and search beams [30] during decoding. Before performing a posterior computation, it is important to observe the WER on a particular data set. The recognition performance on FAA data was found to be 12%, which is acceptable and hence the word graph depth will not be a major factor.

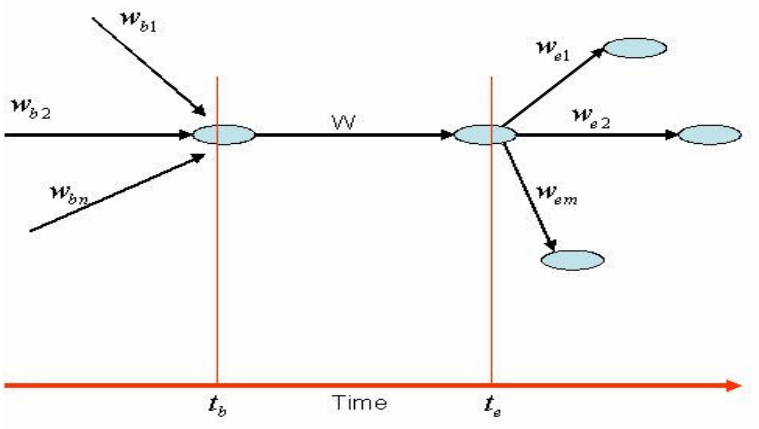

Figure 12 A section of a word graph showing preceding and succeeding nodes

There is an elegant method to compute posterior probabilities from word graphs using a forward-backward type algorithm [29]. The equation to compute word posteriors from a word graph is shown below [29].

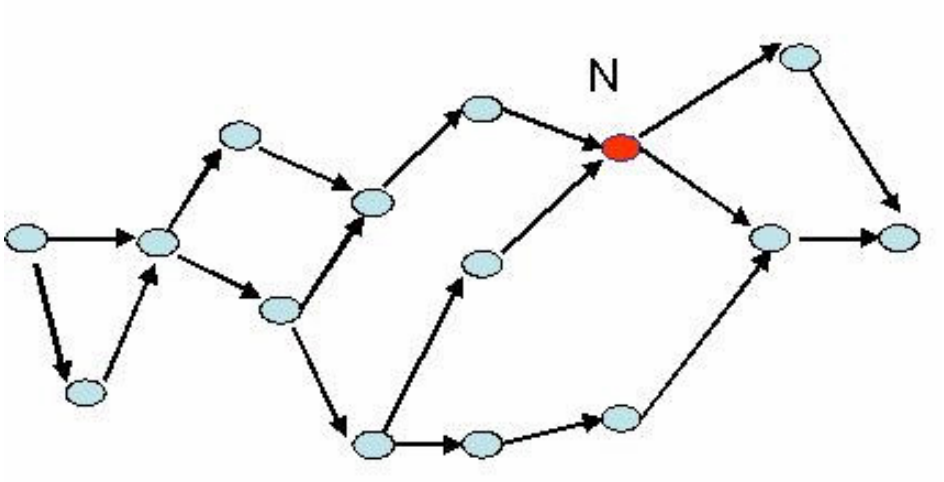

Figure 13 Alternate paths in a word graph entering and exiting a node

$$P(w,t_b,t_e \mid x_1^T) = \sum_{w_b} \sum_{w_e} P(w_b, w, w_e \mid x_1^T)$$

where,

w → single word

t_b, t_e → start and end times of the word

x_1^T → Acoustic vector from time 1 to T.

w_b → Denotes all word sequences preceeding w.

w_e → Denotes all word sequences suceeding w.

, (4)

Equation (4) can be better understood by examining Figure 12. The probability of passing through the link W is calculated by determining the probability of reaching the start node of the word from the preceding nodes and the probability of transitioning from the end node to any of the succeeding nodes. The former is referred to as the forward probability and the latter as the backward probability. A forward-backward type algorithm is used to traverse through the word graph and compute probabilities.

The reason for using forward-backward algorithm can be better understood by examining the example in Figure 13. There are six different ways to react the start node and two different ways to leave node N from the end node. The probability of passing through node N can be obtained by knowing the forward probability and the backward probability of the node. The forward probability is the probability of reaching the node N from the start node, and backward probability is the probability of leaving the node N and reaching the last node. To calculate the probability through a link, one needs to know the forward probability of the start node and the backward probability of the end node.

In equation (4), the right-hand side term cannot be computed directly. Hence, it is decomposed into likelihood and priors using Bayes rule as shown below:

$$\sum_{w_b}\sum_{w_e} P(w_b, w, w_e \mid x_1^{\,T}) = \frac{\sum_{w_a}\sum_{w_e} p(x_1^{\,T} \mid w_a, w, w_e).p(w_a, w, w_e)}{p(x_1^{\,T})}$$

where,

$p(x_1^{\,T} \mid w_a, w, w_e) \quad \rightarrow$ Acousic model probability

$p(w_a, w, w_e) \quad \rightarrow$ Languae model probability

and

$$p(x_1^{\,T}) = \sum_{w}\sum_{w_a}\sum_{w_e} p(x_1^{\,T} \mid w_a, w, w_e).p(w_a, w, w_e)$$

, (5)

The numerator term of equation (5) is calculated by the forward-backward algorithm. The denominator term is the byproduct of the forward-backward computation and is defined as the sum of all paths through the word graph [29]. The purpose of the denominator term is to normalize the posterior values. The posteriors computed in this manner can be used as a confidence measure.

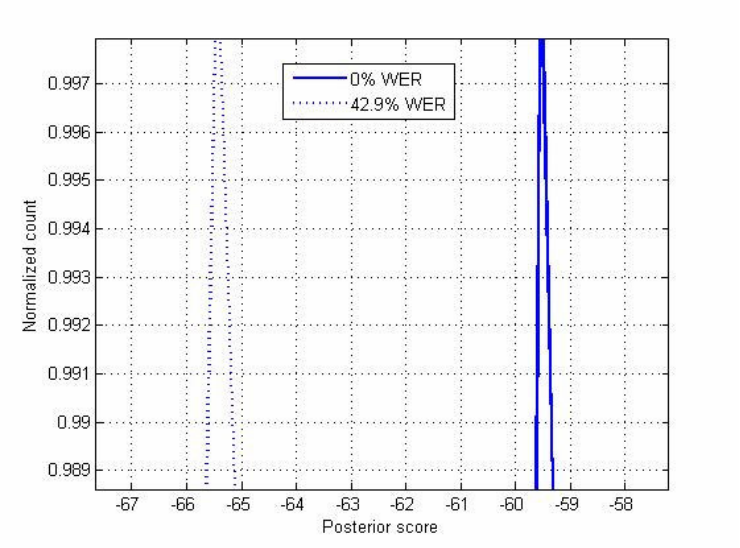

Figure 14 Peaks of posterior distributions for two WERs

To test the effectiveness of word posteriors over likelihood scores, one needs to test the posterior values that were annotated to the same word hypothesis that generated Figure 11. As shown in Figure 14, a distinction in the distribution of the word hypothesis that gave 0% WER and the words that gave 42.9% WER can be seen. This was a promising observation that leads us to believe that word posteriors are better confidence estimators than likelihood scores. Experiments with the fatigue software are discussed in Chapter V.

4.2 An Example Confidence Measure Calculation

A word graph consists of nodes and arcs connected together to represent various alternate hypothesis. The nodes in a word graph are sorted in time and the arcs represent the word hypotheses. The decoder saves the word graph in a text format. The text file contains information about the node indices, node times, arc indices, word labels,

language model probabilities and the acoustic model probabilities. A textual representation of the word graph is shown in Figure 15. The size of the word graph can be controlled by varying search parameters during decoding [30]. The size of the word graph can be judged by observing the number of nodes and arcs in the word graph. The word graphs used in this thesis were generated using a speech recognition system, known as the prototype system, developed at Mississippi State University [14].

```
UTTERANCE=toy_lattice
N=15 L=19
I=0     t=0.00
I=1     t=0.03
I=2     t=0.05
I=3     t=0.09
I=4     t=0.12
I=5     t=0.15
I=6     t=0.15
I=7     t=0.18
I=8     t=0.18
I=9     t=0.18
I=10    t=0.21
I=11    t=0.21
I=12    t=0.25
I=13    t=0.27
I=14    t=0.30
J=0   S=0   E=1   W=<s>       v=0  a=-0.6931  l=0.0000
J=1   S=0   E=2   W=<s>       v=0  a=-0.6931  l=0.0000
J=2   S=1   E=2   W=THIS      v=0  a=-0.6931  l=0.0000
J=3   S=2   E=3   W=IS        v=0  a=-1.0986  l=0.0000
J=4   S=2   E=4   W=THIS      v=0  a=-0.4054  l=0.0000
J=5   S=3   E=5   W=THE       v=0  a=-1.0986  l=0.0000
J=6   S=4   E=5   W=IS        v=0  a=-1.0986  l=0.0000
J=7   S=4   E=6   W=IS        v=0  a=-1.0986  l=0.0000
J=8   S=5   E=7   W=A         v=0  a=-1.7917  l=0.0000
J=9   S=6   E=8   W=THE       v=0  a=-1.7917  l=0.0000
J=10  S=6   E=9   W=A         v=0  a=-0.4054  l=0.0000
J=11  S=7   E=10  W=QUEST     v=0  a=-1.7917  l=0.0000
J=12  S=8   E=10  W=GUEST     v=0  a=-1.7917  l=0.0000
J=13  S=9   E=11  W=TEST      v=0  a=-0.4054  l=0.0000
J=14  S=10  E=13  W=SENSE     v=0  a=-1.7917  l=0.0000
J=15  S=10  E=12  W=SENTENCE  v=0  a=-1.7917  l=0.0000
J=16  S=11  E=12  W=SENTENCE  v=0  a=-0.4054  l=0.0000
J=17  S=13  E=14  W=<s>       v=0  a=-1.7917  l=0.0000
J=18  S=12  E=14  W=<s>       v=0  a=-0.1823  l=0.0000
```

Figure 15 Textual representation of a word graph

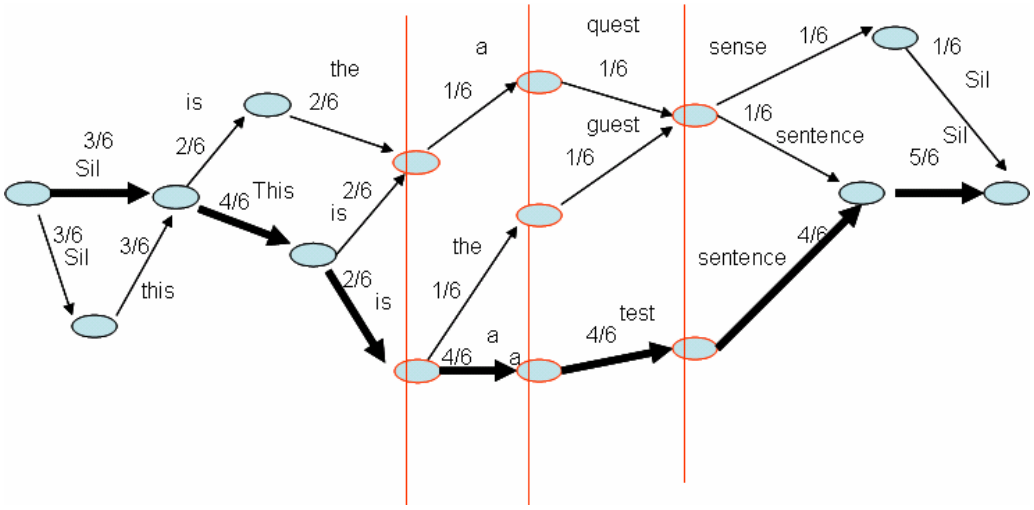

Figure 16 A word graph showing the acoustic likelihood on every arc

The computation of the posterior probability was briefly described in Section 4.1. The posteriors are computed for every link by using a variation of the forward-backward algorithm. The forward-backward algorithm used on a word graph is explained in greater detail in this section using an example. The word graph used in this example is depicted in Figure 16 with the probabilities as shown on the links. The nodes in red signify that they appear at the same time instant. The first step is to compute the forward probabilities (commonly referred to as alphas). This computation is described in the next section.

4.2.1 Computing Alphas

Step 1: Initialization – In a conventional HMM based forward-backward algorithm one would perform the following calculation:

$$\alpha_1(i) = \Pi_i b_i(X_1) \qquad 1 \le i \le N$$

$$\Pi_i \text{ --> Initial prob. of state i}$$
$$b_i(X_1) \text{ --> Emission prob. of the observed data } X_1 \text{ given we are in state i} \quad , \tag{6}$$

The α for the first node is taken as 1:

$$\alpha_1 = 1 \ , \tag{7}$$

Step 2: Induction:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} \right] b_j(X_t) \qquad 2 \le t \le T; 1 \le j \le N$$

$$a_{ij} \text{ --> transition probability}$$
$$b_j(X_t) \text{ --> emmision probability of the observation } X_t \quad , \tag{8}$$

The alpha values computed in the previous step (t) are used to compute the alphas for the succeeding nodes (t+1). Unlike in an HMM, where one moves from left to right at fixed intervals of time, on a word graph one has to move from one node to the next based on node indices which are time aligned.

Let us demonstrate the computation of the alphas from node 2. The alpha for node 1 was initialized as '1' in the previous step.

Node 2:

$$\alpha_2 = 1*(3/6)*1 = 0.5 \ , \tag{9}$$

Node 3:

$$\alpha_3 = (1*(3/6)*1) + (0.5*(3/6)*0.01) = 0.5025 \quad (10)$$

Node 4:

$$\alpha_4 = (0.5025*(2/6)*1) = 1.675\text{E-}03 \quad (11)$$

The alpha calculation continues in this manner for all the remaining nodes. The forward-backward calculation on word graphs is similar to the calculations used on the trellis during Baum-Welch training, but in word graphs the transition matrix is populated by the language model probabilities and the emission probability corresponds to the acoustic score. In this example a constant language model probability of 0.01 was used.

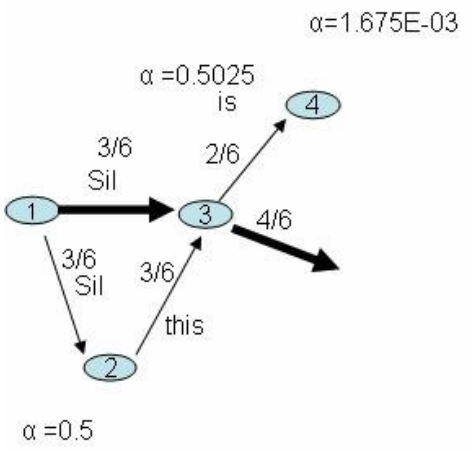

Figure 17 The forward probabilities computed for the first four nodes

4.2.2 Computing Betas

Once the alphas using the forward algorithm are computed, one can begin computation of probabilities using the backward algorithm. This thesis refers to the terms computed in this calculation as betas, since these are related to the alphas computed in the

forward calculation. The backward algorithm is similar to the forward algorithm, but one has to start from the last node and proceed from right to left.

Step 1: Initialization:

$$\beta_T(i) = 1/N \qquad 1 \le i \le \mathrm{N}\ , \tag{12}$$

The number of nodes N at the final instant is 1 and hence β at the final node is 1.

Step 2: Induction:

$$\beta_t(i) = \left[\Sigma_{j=1}^{N} a_{ij} b_j(X_{t+1}) \beta_{t+1}(j)\right] \quad \mathrm{t} = \mathrm{T}-1....1; \quad 1 \le \mathrm{i} \le \mathrm{N}$$

a_{ij} --> Language model score

$b_j(X_{t+1})$ --> The acoustic score

$\beta_{t+1}(j)$ --> The beta value of the nodes just preceeding the current node. , (13)

Let us demonstrate the computation of the beta values from node 14 to node 11:

Node 14:

$$\beta_{14} = (1/6)*1*1 = 0.1667\ , \tag{14}$$

Node 13:

$$\beta_{13} = (5/6)*1*1 = 0.833\ , \tag{15}$$

Node 12:

$$\beta_{12} = (4/6)*0.01*0.833 = 5.555\mathrm{E}-03\ , \tag{16}$$

Node 11:

$$\beta_{11} = ((1/6)*0.01*0.1667) + ((1/6)*0.01*0.833) = 1.666\mathrm{E}-03\ , \tag{17}$$

In a similar manner, one has to obtain the beta values for all the nodes until node 1 is reached. The alpha for the last node should be the same as the beta for the first node.

The posterior probability of a link is computed by simply multiplying the alpha of the start node and the beta of the end node. For example, in Figure 18 the probability of the link between nodes 3 and 4 is obtained by multiplying alpha of node 3 and beta of node 4. The posterior probabilities are normalized by dividing the product of alpha and beta by the sum of all paths through the word graph. The sum of all paths in the word graph is represented by the denominator term in equation (5), $p(x_1^T)$. The value of the sum of all paths through the word graph is a by-product of the forward-backward calculations. The denominator tern of equation (5) is either the alpha on the last node or the beta on the first node. Chapter V will discuss experimental results obtained using word posteriors as a confidence measure.

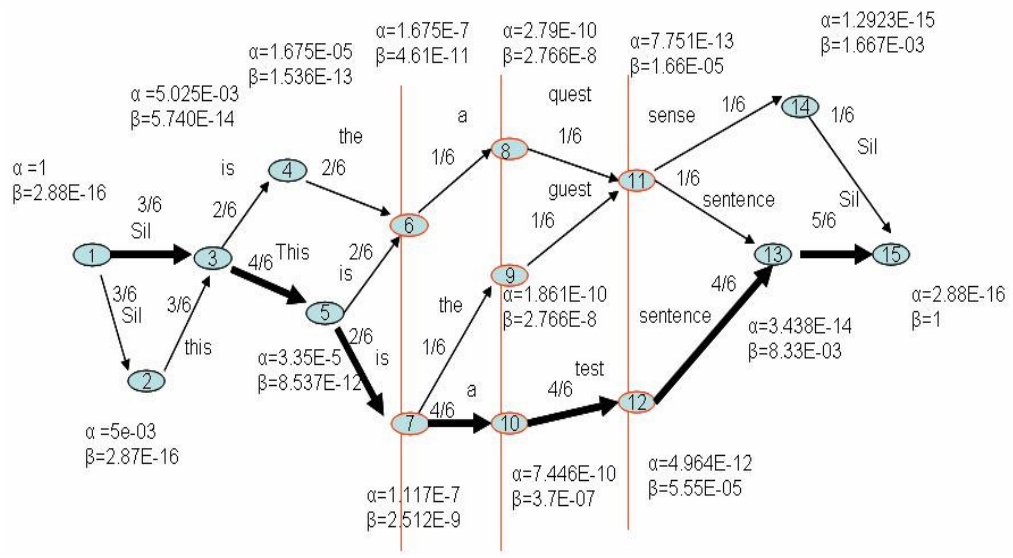

Figure 18 Word graph with alphas and betas computed for every node

CHAPTER V

EXPERIMENTAL RESULTS AND ANALYSIS

This chapter discusses various experiments that were run to assess the impact of a confidence measure on fatigue detection. Experiments were run on three different data sets. The first of the three data sets, the Phase II data, was collected during a military exercise [31]. This data was primarily collected for use with an ASR system. As described in Chapter II, the Phase II data had some issues regarding quality of recording. ASR experiments run using the Phase II gave a word error rate of 50%. A detailed tabulation of results on Phase II data is provided in this chapter.

Another data called as the FAA data [31], was used with the ASR system. The WER on this data was 12%. The FAA database contained 31 words spoken in a studio environment. The speakers were subjected to controlled sleep cycles, as this was necessary to induce fatigue. The data recorded from this experiment contained very long utterances, and therefore had to be carefully chopped for better acoustic model training and faster decoding [32]. The results from recognition experiments run on the FAA data are also discussed in this chapter.

A third data set similar to the FAA data, which was called the Bravo data, was used for fatigue prediction experiments by Greeley, *et al.* [31]. Fatigue prediction using

voice was compared with other standard fatigue metrics such as SOL [25] and SAFTE [33]. Fatigue prediction from voice was also tested with and without a confidence measure. It was found that the confidence measures helped in bringing the voice based fatigue prediction closer to other metrics of fatigue detection such as SOL and SAFTE.

5.1 Recognition Experiments on Phase II Data

During the data recording process, there was constant monitoring of each speaker's level of fatigue followed by a recording of the subject's speech. The subjects were asked to recite eight fixed phrases and one spontaneous phrase. Phonetic alignment of the data was needed to analyze signs of fatigue, and these alignments were obtained using an ASR system. The Phase II data required segmentation in order to make it suitable for use with an ASR system [31]. The original set of utterances had variable durations and contained significant amounts of silence and noise between words. The average duration of the original utterance was 45 seconds. The segmented utterances had lengths of approximately 5 seconds each.

The segmented utterances were used to build an ASR system. The feature extraction block produced MFCC [6] features from the raw speech data. The features were extracted every 10 msec using a 25 msec analysis window. The experiments in this thesis used standard 39-dimensional MFCC features [6].

The acoustic modeling block uses a GMM model with a standard left to right HMM topology [16]. Only 50% of the data was used for training, and the remaining was used for testing. The number of mixtures used for building the GMMs was varied from one to sixteen. A cross-word triphone model was used as the fundamental acoustic model [30]. It was found that an 8-mixture GMM model was optimum for this task. Experimental results on optimization of this parameter are shown in Table 2.

Table 2 WER as a function of the number of mixtures

No. of Mixtures	WER %
1	10.1
2	5.3
4	1.1
8	0.0
16	0.0

The grammar used for this experiment consisted of a loop grammar, with the sentence transcription embedded into a single node. During decoding, the ASR system requires a language model. The language model helps in constricting the search space [30]. The ASR system was run using both a loop grammar [16] and a bigram language model [34]. The Phase II data set contained eight fixed phrases and one spontaneous phrase from each speaker. The initial experiment used a loop grammar with the sentence transcription embedded into the grammar (only the fixed phrases were used for this experiment). The grammar contained loops of eight sentence sequences, with each sentence representing a node in the grammar. With this grammar the ASR system gave 100% accuracy.

The grammar had to be changed when the test data included spontaneous phrases. New sentence sequences were added to the grammar and the ASR system was run with the updated test set that included spontaneous utterances. This time, the WER of the system increased to 34% compared to 0% for fixed phrases. This increase was because the words in the spontaneous phrases did not occur as often in the training data as the words in the fixed phrases.

One way to improve performance of an ASR system is by strengthening the language model. In order to strengthen the language model, an interpolated bigram model [34] was used, instead of a loop grammar. New words were added to the lexicon and the bigram language model was interpolated [35] from a Switchboard bigram language model [34]. The perplexity [34] of the interpolated language model was kept close to the perplexity of the Phase II language model. Language model interpolation was performed using the SRILM toolkit [36].

Experiments with a bigram language model gave a WER of 52.4% on unseen fixed phrases. When spontaneous phrases were included in the test set, the WER increased to 74.5%. The WERs were much worse than that found using a loop grammar. But note that in this case the system did not use the sentence transcription in the grammar, so it is a more generic system than the loop grammar system. Experiments were run in order to tune the parameters for optimizing the WER. The results of these experiments are shown in Table 3.

Table 3 Experimental results on the Phase 2 data using a 16-mixture cross-word triphone system

Grammar Type	WER %
Sentence level grammar (Fixed phrases)	0.0
Sentence level grammar (Fixed+ Spontaneous phrases)	34.0
Word level grammar (Fixed phrases)	60.0
Word level grammar (Fixed + Spontaneous phrases)	82.0
Bigram model (Fixed phrases)	52.4
Bigram model (Fixed + Spontaneous phrases)	74.5

The goal of running these experiments was to obtain robust ASR models that could be used for obtaining phonetic alignments for previously unseen utterances (open-loop testing). Apparently, this was not possible because the size of the database was small, and also the quality of the data was very poor. The challenge to make the system robust to noise conditions is beyond the scope of this thesis. The Phase II data could not be used for building an automated fatigue detection system because of its high WER, and hence no further analysis was pursued using this data.

5.2 Recognition Experiments on FAA Data

For the experiments on the FAA data, the ASR system used a loop grammar for the language model and used cross-word triphones for acoustic model. The first set of experiments was conducted to determine the optimum model type. The WERs for word, monophone, and cross-word models are shown in Table 4. All the states in the model were represented by a single-mixture Gaussian model. These experiments were

conducted in a closed-loop framework, .i.e. the training and the testing was performed on same data.

Table 4 WER as a function of the model type

Model Type	WER %
Word	63.9
Monophone	54.3
Cross-word triphone	47.3

From the results shown in Table 4, it can be observed that the cross-word triphone model gave the lowest WER. The WER for cross-word models was further improved by using a larger number of mixture components. The improvement in WER as a function of the number of mixtures is shown in Table 5. It was found that an 8-mixture model was optimum for this data. The WER was still lowered by adjusting the state-tying parameters. The state-tying parameters have a significant effect on the WER [17].

Table 5 WER as a function of the number of mixtures for cross-word models on the FAA data

No. of Mixtures	WER %
1	47.3
2	36.3
4	23.6
8	11.3
16	11.3

Table 6 Effect of state-tying parameters on the WER

Split threshold	Merge threshold	Occupancy threshold	No. of states	WER
650	650	1400	20	11.3
165	165	840	37	8.5
150	150	900	34	8.4
125	125	750	41	5.7
110	110	660	47	4.8
100	100	600	56	3.8
75	75	550	57	3.6
50	50	500	58	4.0
25	25	450	62	3.0
10	10	250	94	1.1
10	10	100	118	0.5
10	10	50	126	0.1

The changes in the state-tying parameters affect the number of tied states in the model [17]. Table 6 shows the WER as a function of the number of tied states in the model. A minimum WER of 0.1% was obtained on the closed-loop setup. Increasing the number of tied states poses the problem of over fitting the training data. To further analyze the effect of state-tying, this thesis tested the models using a cross-validation scenario in which performance was measured on held-out data. The unseen data consisted of 20% of the original data which was separated and was not used for training. The results of the experiments are shown in Table 7.

It can be observed that the minimum WER is obtained when the number of tied states is 56. Increasing the number of tied states beyond 56 causes the model to overfit the training data and hence leads to increase in WER. The general idea behind state tying is to generalize the acoustic model by sharing parameters; however, if not done correctly it will yield negative results.

For fatigue analysis, this thesis used a closed loop system because the phonetic alignments were more accurate. This was an acceptable approach for initial pilot experiments. To build a real system, one needs the ASR to work on unseen data. One can broadly classify unseen .data into three categories: (a) seen speaker with OOVs, (b) unseen speaker with no OOVs, and (c) unseen speaker with OOVs. Discussion and analysis on noise and channel characteristics of the unseen data is beyond the scope of this thesis. This thesis addressed the problem described in category (a) since our goal was to determine the feasibility of the fatigue detection approach.

Table 7 Effect of state-tying parameters on WER with unseen FAA data included in the test set

Split threshold	Merge threshold	Occupancy threshold	Number of states	WER %
650	650	1400	20	11.5
165	165	840	31	17.3
150	150	900	29	19.8
125	125	750	38	25.2
110	110	660	42	14.8
100	100	600	54	12.2
75	75	550	56	9.8
50	50	500	57	7.1
25	25	450	56	3.9
10	10	250	94	7.3
10	10	100	119	12.5
10	10	50	122	11.4
10	0	0	131	25.3

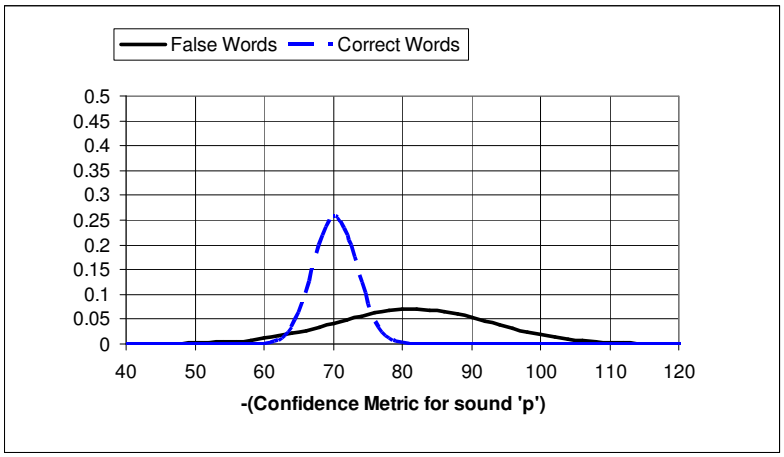

Figure 19 Distribution of confidence scores for false and correct words

Whenever there were OOVs in the test data, the ASR system generated many false hypotheses. To overcome this problem, a confidence measure algorithm was implemented and annotated each word hypothesis with a confidence score. The fatigue analysis software used the confidence measure to filter out false hypotheses. The effectiveness of confidence measures can be observed in Figure 19 which shows the distribution of correct and false word confidence scores. A clear difference in their mean values can be observed, and as a result one can use a threshold to select only the true words.

5.3 Fatigue Detection Experiments

The effectiveness of the confidence scores was evaluated by examining the effect of OOVs on the test set. Analysis was done on the voice data from two test subjects who underwent a night of sleep deprivation. At six test epochs, separated by 6 hours, these subjects each recited from two word lists. One of the lists contained words from the

training data set and the second list contained words not seen during training. This thesis refers to the latter as the foreign list. Both the subjects were part of the training speaker set.

Table 8 An analysis of the confidence metric

	Subject 6	
	Training	Foreign
Average CM	-72.22	-81.51
CM Standard Deviation	3.10	11.34
	Subject 8	
	Training	Foreign
Average CM	-70.24	-82.41
CM Standard Deviation	3.50	15.16

The ASR system was trained to recognize words from the training list. During fatigue analysis, the speech recognition system was presented words from both the training list and the foreign list which contained words not seen during training. For both subjects, the confidence score observed when the speakers recited from the first list had a higher average value and smaller standard deviation than that observed when the speaker recited from the foreign list. Table 8 presents these results. It was observed that the average confidence measure score for falsely recognized words was 15% less than that for words spoken from within the training set. This was a positive indication, and hence one could use these confidence measures to filter out false alarms from an ASR system's output.

The discriminative power of confidence measures was analyzed by using a receiver operating characteristic (ROC) curve. The area under the ROC curve is an indication of the discriminative power of the classifier. An area of 0.5 indicates a random classifier while an area of 1.0 indicates an ideal classifier. The area under the ROC curve for the system that incorporated a confidence measure was 0.85, which indicates good discrimination. The ROC plot is shown in Figure 20. A suitable operating point or threshold had to be determined for classification. A threshold of -75 was chosen because at that point the probability of false alarms was equal to the probability of true occurrences of words. This point is also called as the Equal Error Rate (EER) point [37].

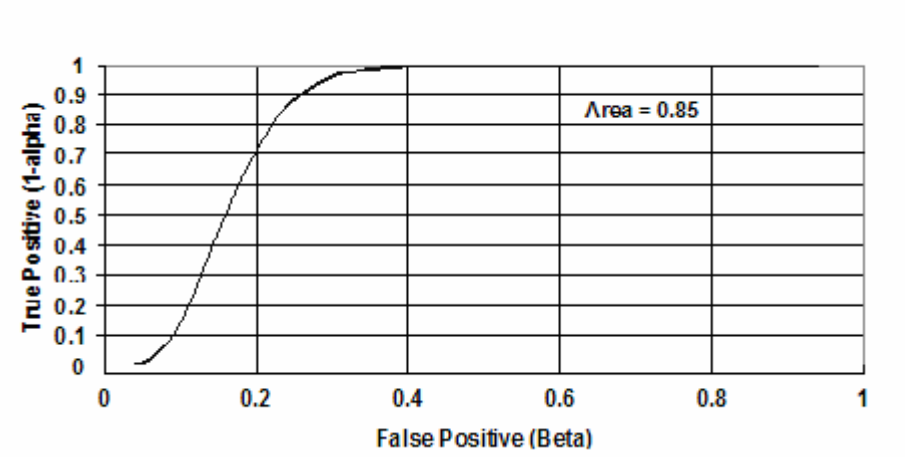

Figure 20 Receiver Operating Characteristic (ROC)

A Detection Error Tradeoff (DET) curve would be most suited for judging the performance of the fatigue detection system, but this requires large amounts of data. Greeley, *et al.* [5] are currently working on such a task. This thesis observed the effectiveness of confidence measures based on the error difference between standardized fatigue detection metrics and voice-based fatigue prediction metrics. Comparisons of various metrics are provided in Figure 21 and Figure 22.

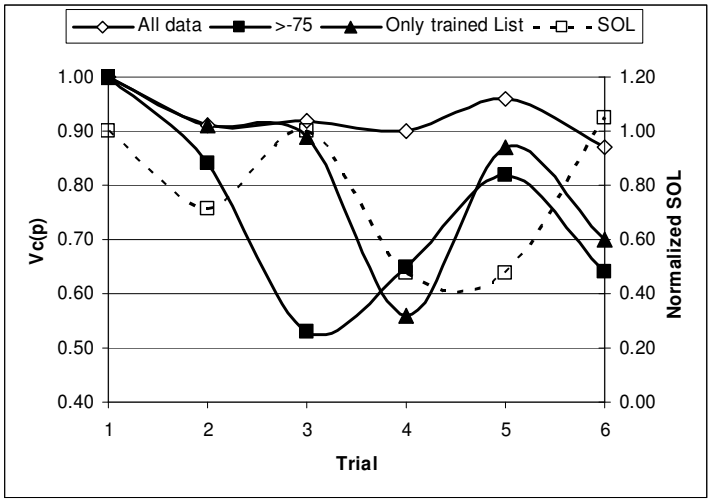

Figure 21 Comparison of the trend between SOL and voice correlation for the sound 'p' with and without a confidence metric

Figure 21 demonstrates a comparison between a subject's normalized sleep onset latency (SOL) [3] and the voice-based fatigue prediction for the sound 'p' (Vc(p)). These metrics were obtained for each of the six trials. It can be observed that the error between the "SOL" metric and the closed loop (no OOVs) voice-based fatigue prediction metric is smallest. For analysis the SOL metric and the voice based fatigue metric were normalized to the same scale. The differences in the metrics at various test epochs were computed

and used that as a metric to judge the performance of the voice-based fatigue prediction system. The average error between the six test epochs for the closed loop voice based metric was 0.20.

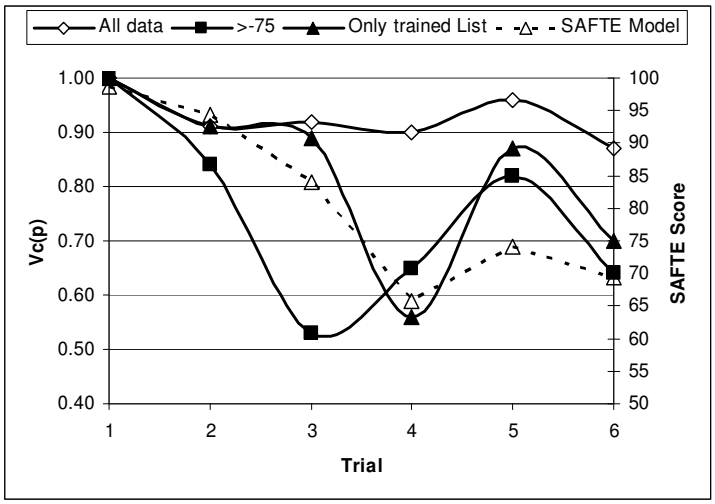

Figure 22 Comparison of the trend between SAFTE and voice correlation for the sound 'p' with and without confidence metric

Using the voice input containing both training and foreign set words, with no confidence metric, the error between the metrics was found to be 0.33. However, by using the confidence measure, with a threshold setting of -75, the error rate decreased to 0.30, which represents a 9% improvement. A much more significant improvement was observed when the voice-based fatigue prediction was compared with Sleep, Activity, Fatigue, and Task Effectiveness (SAFTE) model [33], which is depicted in Figure 22.

The cyclic pattern observed in Figure 21 and Figure 22 is due to circadian rhythms [25]. Over the 30 hours between trial number 1 and trial number 6, a full circadian cycle has elapsed. The SOL reflects the circadian influence of an individual's need to sleep [3]. A more direct way to match a speaker's overall performance and

circadian influences is to use the speaker's body temperature and his or her time without sleep. This is accomplished using the SAFTE model [33].

Figure 22 shows speaker's SAFTE score and the voice-based fatigue prediction for the sound 'p' (Vc(p)) at each of the six trials. As was the case with the SOL, the SAFTE model and the closed loop (no OOV) voice-based fatigue prediction had the lowest difference in metrics after normalization. The error difference was found to be 0.10 considering all six test epochs. Using voice input containing mixed words (with OOVs), the confidence measure-based system provides a significant improvement over the use of a system without confidence measures. The error difference between the SAFTE metric and the voice-based fatigue detection metric decreased from 0.15 to 0.12 with the use of confidence measures, which represents a 20% improvement. This was observed when the test set had an OOV rate of 61.7%.

CHAPTER VI

CONCLUSION AND FUTURE WORK

Non-intrusive fatigue assessment systems are needed to successfully monitor the level of alertness of all personnel during critical mission or life-threatening activities. This thesis explores the first attempt at detecting fatigue from voice using an ASR system. Various approaches such as speaker verification, word spotting and LVCSR techniques were analyzed in this thesis, and the LVCSR approach was found to be superior for this particular task. The LVCSR approach did not require fatigue-dependent data for training and it used a fixed grammar. LVCSR approach was relatively more effective when dealing with OOVs, as compared to the word spotting approach. The OOVs caused insertion and substitution errors in the final output. The fatigue detection system treated the insertions and deletions generated by the LVCSR system as false alarms. The problem of false alarms was tackled by implementing a confidence measure algorithm. The LVCSR system output was annotated with a word posterior-based confidence measure. The confidence measure was used to filter out false alarms. Use of the confidence measure improved the robustness of the fatigue detection system to OOVs by 20%.

6.1 Thesis Contribution

This thesis explored the use of an LVCSR system to automate the task of voice-based fatigue detection. As discussed in Chapter II, Greeley, *et al.* [5] found that certain phonemes in human speech are more affected by fatigue than others. Hence, there was a need to obtain accurate phonetic alignments for those particular phonemes on test data and use those alignments for fatigue detection. Obtaining accurate phonetic alignments on a data set with OOVs is a challenge for any ASR system. In the case of fatigue detection, the greatest problem was to counter false alarms caused by insertions and substitutions in the LVCSR's output. The false alarms caused errors in the fatigue detection system.

This thesis explored a technique by which one can generate robust phonetic alignment for fatigue detection even when the test data contained OOVs. The improvement in robustness was achieved by using a word posterior-based confidence measure. The confidence measure algorithm computed word posteriors from a word graph. The word posteriors were computed using a forward backward type algorithm as described in Chapter III. The confidence measure algorithm was embedded into the core ASR system. This upgraded version of the ASR system was used by Creare, Inc. [5] to perform fatigue detection. Fatigue detection performance improved by 20% when 61.7% of the words in the test set were OOVs.

6.2 Future Work

Though every effort was made to generalize the acoustic models built by the LVCSR system, it would definitely be better to use larger data sets for acoustic model training. Running the experiments on fatigue dataset with larger number and variety of speakers would enhance its value. The overall architecture of the system is currently suited only for laboratory analysis. The system could be made to perform in near real time if a one-pass strategy to compute confidence measures were employed. Also, computing confidence measures on the fly, rather than generating word graphs, would further enhance the usability of the system. The ASR system for fatigue detection used a loop grammar, in which case the word posteriors are entirely dependent on the acoustic score, and hence the scores are slightly biased towards the training data. By using a statistical language model word posterior scores that are more effective could be obtained.

REFERENCES

[1] P. Colquhoun, "Psychological and Psycho Physiological Aspects of Work and Fatigue," *Activitas Nervosa Superior*, vol. 18, no. 4, pp. 257-263, March 1976.

[2] T. Roehrs, V. Timms, A.Z. Doorenbos, and T. Roth, "Sleep Extension in Sleepy and Alert Normals," *Sleep*, vol. 12, no. 5, pp. 449-457, October 1989.

[3] J. Whitmore and S. Fisher, "Speech During Sustained Operations," *Speech Communication*, vol. 20, no. 2, pp. 55-70, November 1996.

[4] I. Saito, O. Fujiwara, N. Utsuki, C. Mizumoto, and T. Arimori, "Hypoxia-Induced Fatal Aircraft Accident Revealed by Voice Analysis," *Aviation, Space, and Environmental Medicine*, vol. 51, no. 4, pp. 402-406, April 1980.

[5] H. Greeley, J. Berg, E. Friets, J. Wilson, G. Greenough, J. Picone and J. Whitmore, "Fatigue Prediction Using Voice Analysis," submitted to *Behavorial Research Methods*, February 2006.

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

[7] W.M. Campbell, J.R. Campbell, D.A. Reynolds, D.A. Jones, T.R. Leek, "High-level Speaker Verification with Support Vector Machines," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 73-76, Montreal, Quebec, Canada, May 2004.

[8] M.G. Frank and P. Ekman, "The Ability to Detect Deceit Generalized Across Different Types of High-Stake Lies," *Journal of Personality and Social Psychology*, vol. 72, no. 6, pp. 1429-1439, January 1997.

[9] J.E. Beck, K. Chang, J. Mostow, and A. Corbett, "Using a Student Model to Improve a Computer Tutor's Speech Recognition," *Proceedings of International Conference on Artificial Intelligence in Education*, vol. 7, pp. 2-11, Las Vegas, Nevada, USA, June 2005.

[10] M.A. Zissman, "Automatic Language Identification Using Gaussian Mixture and Hidden Markov Models," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 399-402, Minneapolis, Minnesota, USA, April 1993.

[11] S.S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman and J. Zheng, "SRI's 2004 NIST Speaker Recognition Evaluation System," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 173-176, Philadelphia, Pennsylvania, USA, March 2005.

[12] J.G. Wilpon, L.R. Rabiner, C.H. Lee, and E.R. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1870-1878, Albuquerque, New Mexico, USA, November 1990.

[13] L. Mangu, E. Brill and A. Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks," *Computer, Speech and Language*, vol. 14, no. 4, pp. 373-400, October 2000.

[14] "Automatic Speech Recognition," *http://www.cavs.msstate.edu/hse/ies/projects/speech/index.html*, Intelligent Electronic Systems, Mississippi State University, Mississippi State, Mississippi, USA, May 2006.

[15] J. Picone, "Continuous Speech Recognition Using Hidden Markov Models," *IEEE ASSP Magazine*, vol. 7, no. 3, pp. 26-41, July 1990.

[16] I. Alphonso, *Network Training for Continuous Speech Recognition*, M.S. Thesis, Department of Electrical and Computer Engineering, Mississippi State University, August 2003.

[17] N. Parihar, *Performance Analysis of Advanced Front Ends on the Aurora Large Vocabulary Evaluation*, M.S. Thesis, Department of Electrical and Computer Engineering, Mississippi State University, November 2003.

[18] L.R. Bahl, P.V. De Souza, P.S. Gopalakrishnan, D. Nahamoo and M.A. Pichney, "Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees," *Proceedings of the DARPA Speech and Natural Language Processing Workshop*, pp. 264-270, San Mateo, California, USA, February 1991.

[19] L. Deng, A. Acero, M. Plumpe and X. Huang, "Large-Vocabulary Speech Recognition under Adverse Acoustic Environments," *Proceedings of the International Conference on Spoken Language Processing*, vol. 3, pp. 806-809, Beijing, China, October 2000.

[20] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Upper Saddle River, New Jersey, USA, 1978.

[21] J.H.L. Hansen, "Analysis and Compensation of Speech under Stress and Noise for Environmental Robustness in Speech Recognition," *Speech Communication – Special Issue on Speech Under Stress*, vol. 20, no. 2, pp. 151-170, November 1996.

[22] J.M. Gregory, X. Xie, and S.A. Megel, "SLEEP (Sleep Loss Effects on Everyday Performance) Model," *Aviation, Space, and Environmental Medicine,* vol. 75, no. 1, pp. 125-133, December 2004.

[23] M.W. Geisler and J. Polich, "P300 and Time of Day: Circadian Rhythms, Food Intake, and Body Temperature," *Biological Psychology*, vol. 31, no. 2, pp. 117-136, July 1990.

[24] T. Roth, T.A. Roehrs, M.A. Carskadon, and W.C. Dement, *Daytime Sleepiness and Alertness in Principles and Practice of Sleep Medicine*, W.B. Saunders Company, Philadelphia, Pennsylvania, USA, 1989.

[25] T. Roehrs and T. Roth, "Multiple Sleep Latency Test: Technical Aspects and Normal Values," *The Journal of Neurophysiology*, vol. 9, no. 1, pp. 63-67, October 1992.

[26] D. Palmer, J. Burger, and M. Ostendorf, "Information Extraction from Broadcast News Speech Data," *Proceedings of the DARPA Broadcast News Workshop*, pp. 41-46, Herndon, Virginia, USA, February 1999.

[27] J. Fiscus, A. Le, and G. Sanders, "MDE Tasks and Results," presented at the Effective, Affordable, Reusable Speech-to-Text (EARS): The Rich Transcription Evaluation Workshop, Palisades, New York, USA, November 2004.

[28] J.P. Campbell, "Speaker Recognition: A Tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437-1462, September 1997.

[29] F. Wessel, R. Schlüter, K. Macherey and H. Ney, "Confidence Measures for Large Vocabulary Continuous Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288-298, November 2001.

[30] N. Deshmukh, A. Ganapathiraju and J. Picone, "Hierarchical Search for Large Vocabulary Conversational Speech Recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 84-107, September 1999.

[31] H.P. Greeley, J. Berg, E. Friets, J.P. Wilson, S. Raghavan and J. Picone, "Detecting Fatigue from Voice Using Speech Recognition," to be presented at the IEEE International Symposium on Signal Processing and Information Technology, Vancouver, Canada, August 2006.

[32] N. Deshmukh, A. Ganapathiraju, A. Gleeson, J. Hamaker and J. Picone, "Resegmentation of Switchboard," *Proceedings of the International Conference on Spoken Language Processing*, pp. 1543-1546, Sydney, Australia, November 1998.

[33] S.R. Hursh, D.P. Redmond, M.L. Johnson, D.R. Thorne, G. Belenky, T.J. Balkin, W.F. Storm, J.C. Miller and D.R. Eddy, "Fatigue Models for Applied Research in War Fighting," *Aviation, Space, and Environmental Medicine,* vol. 75, no. 3, pp. 44-53, December 2004.

[34] D. Jurafsky and J.H. Martin, *An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice-Hall, Upper Saddle River, New Jersey, USA, 2000.

[35] X. Huang, A. Acero, and H.W. Hon, *Spoken Language Processing – A Guide to Theory, Algorithm, and System Development*, Prentice-Hall, Upper Saddle River, New Jersey, USA, 2001.

[36] A. Stolcke, "SRILM An Extensible Language Modeling Toolkit," *Proceedings of International Conference on Spoken Language Processing*, vol. 2, pp. 901-904, Denver, Colorado, USA, September 2002.

[37] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET Curve in Assessment of Detection Task Performance," *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH)*, vol. 4, pp. 1895-1898, Rhodes, Greece, September 1997.