Internal Use Only (非公開)

TR－SLT－0045

Relating Phonetic Feature Stream Reliability to Noise Robust Speech Recognition

Alex Park

2003.08.28

This report describes an attempt to improve robustness in automatic speech recognition by optimizing the extraction of phonetic feature streams independently of the recognition process. The eventual goal is to use a bank of feature extraction modules which use specialized signal processing and statistical techniques to reliably extract speech relevant features from the acoustic signal. In this work, we demonstrate the viability of using a sparse set of feature streams for a simple connected digit recognition task. We then illustrate some techniques for improving the reliability of the voicing feature module and evaluate several alternatives modules using both clean and noisy data. Finally, we relate the reliability of extraction for the individual voicing module to the overall performance of the recognizer by performing recognition experiments on the Aurora 2 database.

(株) 国際電気通信基礎技術研究所
音声言語コミュニケーション研究所
〒619－0288「けいはんな学研都市」光台二丁目2番地2 TEL：0774－95－1301

Advanced Telecommunication Research Institute International
Spoken Language Translation Research Laboratories
2-2-2 Hikaridai "Keihanna Science City" 619-0288,Japan
Telephone:+81-774-95-1301
Fax :+81-774-95-1308

Contents

List of Figures

List of Tables

1

Introduction

One of the main challenges facing automatic speech recognition research is the issue of robustness to noise and other environmental corruption. While speech recognition performance by humans tends to degrade gracefully in the presence of noise, even state-of-the-art automatic speech recognition (ASR) systems tend to fail drastically when confronted with corrupted speech.

1.1 The problem of mismatch

At the heart of the robustness dilemma lies the problem of mismatch between training and testing data. Modern ASR systems represent the speech signal as a time series of spectral feature vectors, $\mathbf{v}_t$, which are computed over windowed frames of the time domain signal at a fixed frame rate. During training, feature vectors belonging to the same class, C_i (at the word, syllable, phone, etc. level), are used to estimate the parameters, θ_i of the statistical acoustic model for that class. This phase produces, for each class C_i, a compact probability distribution representation, $f_i(\mathbf{v};\theta_i)$, which captures the aggregate properties of its training data.

$$\mathbf{V}_i = \{\mathbf{v}|\mathbf{v} \in C_i\} \longrightarrow \theta_i$$

During recognition, the test utterance is also converted to a time series of spectral feature vectors, $\mathbf{y}_t$. The likelihood that a feature vector belongs to a particular class C_i, a critical measure used in all subsequent stages of recognition when determining likelihoods of words and phrases, is computed by calculating the probability of occurence from the statistical model for that class, $f_i(\mathbf{y}_t;\theta_i)$. The key assumption in this framework is that the distribution of feature vectors in a class will be the same in testing as they were observed in training.

However, there are many reasons why this assumption might fail to hold in practice. The test environment could include additive noise or reverberation that was not present in during training. The recording microphone or transmission channel might be also be different from what was used during training. Even speaker variation and differences in speaking rate, though unrelated to environmental conditions, are sources of mismatch which often cause significant difficulties for most ASR systems.

1.2 Dealing with mismatch

In general, most approaches to dealing with mismatch fall under one of three categories, which are described in the following sections.

1.2.1 Multi-condition training

This approach attempts to reduce instances of mismatch by including more varied examples of training data during training. Thus, mismatch due to noise is reduced by training on data collected in a variety of noisy environments and mismatch due to speaker variation is reduced by training on data collected from a wide catalogue of speakers.

In some ways, this strategy is attractive because it is not knowledge intensive, yet improves performance on noisy data. However, this approach is not scalable, since training can never be exhaustive, which means that unseen testing conditions will always exist. Moreover, data collection is often time consuming and expensive, and multi-condition training typically worsens performance on clean data.

1.2.2 Model adaptation

Model adaptation strategies attempt to change the parameters of the learned acoustic model to better match the characteristics of the incoming speech. Speaker adaptation falls under this category, as do many robustness techniques including Parallel Model Combination [1]. Model adaptation is performed by first estimating the test conditions, then adapting the model parameters to those conditions. The main drawbacks to model adaptation strategies are heavy computation cost and degradation of performance in situations where the adaptation parameters are not accurately estimated.

1.2.3 Feature domain compensation

Feature based approaches attempt to address the mismatch problem either by using features which are discriminative for speech and largely invariant to other factors, or by

cleaning the features prior to presenting them to the recognizer. The motivation for the former approach stems from the observation that frame-level Mel scale Cepstral Coefficient (MFCC) vectors, which are widely used in ASR systems, are typically not consistently extracted in the presence of noise or other corruption. Some proposed alternative feature representations that have been demonstrated to be more robust than MFCCs are the modulation spectrogram [2] and RASTA features [3]. In addition, delta MFCC are frequently used in addition to standard MFCC feature sets to improve noise robustness. Feature cleaning techniques include methods like Vector Taylor Series [4] and Algonquin [5].

A related feature domain approach to robustness is the application of missing feature theory, where the reliability of front end feature components are determined by the estimated local signal to noise ratio. Those components of the feature vector that fall below a certain SNR are deemed missing, and can then be either reconstructed [6] or marginalized out when computing likelihoods during recognition [7]. The main difficulty in utilizing missing feature theory in practice remains determining the correct SNR mask. If, during recognition, the system marks the wrong components as missing, then serious degradations in accuracy can result from relying on the incorrect portions of the signal.

1.3 Recognition with independent feature stream modules

The framework advocated in this work is aligned most closely with the the feature domain approaches described in the previous section. Like previous feature based approaches, we believe in presenting the acoustic modelling layer with a set of features that are robustly and consistently extracted in all conditions. However, unlike RASTA and the modulation spectrogram, we ground our choice of features to phonetic attributes or distinctive features. One advantage of these types of features is that we can evaluate the quality of the feature detectors under different conditions without relying on global word error rate as the sole criteria for optimality. An additional advantage is that multiple techniques and inputs can be used calculate each possible feature stream. Finally, the use of phonetic features provides a way to meaningfully diagnose recognizer failure in new environments. As it stands now, the feature vectors provided as input to the acoustic models have little diagnostic information because the values do not correspond to any intuitive perceptual attribute of the speech signal. Our goal is to decouple the evaluation of features from the rest of the recognition process. Eventually, we plan to join the feature extraction and acoustic modeling stages using a layered paradigm which relies on the following principles:

(1) Abstraction. The modelling layer should not affect the features computed. Just as in networking, the client layer is not concerned with the route that a packet takes in the network, neither should the modelling layer be concerned with how a particular feature stream is computed.

(2) Consistency. For a given underlying speech state, the goal of the feature extraction layer should be to present the modelling layer with the same value under any condition.

(3) Redundancy. Multiple processing strategies can and should be used in the feature extraction layer to provide the most robust estimate of a target feature. For example, if video input is present in addition to audio, then lip motion should be used in addition to, formant motion to get a more accurate estimate of labializtion.

(4) Integrity. Not only should the feature extraction layer attempt to provide the most consistent and accurate estimate of the target feature, it should also provide information about the reliability of such an estimate. For example, if the estimated SNR of the input utterance becomes very low, then certain feature streams such as frication should be given a low reliability score. When integrated with appropriate higher layers, the use of reliability scores will allow recognizers to focus on more reliable acoustic cues as the signal degrades, and in the worst case, output no hypothesis in the absence of any reliable streams. The hope is that such a framework will eliminate instances of random noise inputs generating nonsensical word hypotheses from the recognizer

1.4 Previous work in feature based recognition

Recognition strategies using feature-based modeling have recently gained popularity in the research community because of their ability to explain articulatory phenomena that occur in conversational speech. For example, Deng *et al.* [8] use a hidden state variable to model and constrain the evolution of articulatory features. Livescu [9] has proposed an extension to this approach which uses multiple hidden states in a dynamic Bayesian network to model the relationships between hidden articulatory feature streams. An alternate strategy proposed by Kirchoff [10] makes estimates of the articulatory features themselves, then treats these estimates as observed variables for recognition. Similarily, Metze *et al.* [11] have demonstrated improvements on read speech using a multi-stream hidden Markov model (HMM) architecture supported by a set of articulatory feature detectors.

In the area of distinctive features, Hermansky *et al.* [12] have described the use of a connectionist approach, where traditional input feature vectors are first passed through

a neural network to generate posterior probabilities for subword phonetic units, which are then used as the base features for a conventional recognition system. This is closely related to work by Lee *et al.* [13], [14] who demonstrated a similar approach using neural networks to compute streams of distinctive feature probabilities for use as the input to an HMM recognizer.

While all of the works mentioned in this section share the common goal of using speech-relevant feature streams for improving recognition, the focus has been exclusively on lowering word error rate while making little attempt at ensuring the fidelity of the actual features which are computed. At the same time, there have been several efforts by researchers to develop detectors optimized for specific phonetic classes such as nasals [15], sonorants [16], and stops [17]. Though these works demonstrate detection performance, they stop short of producing results on actual speech recognition tasks. A logical next step would be to combine these specialized detectors with the stream-based recognizers discussed above.

1.5 Overview

The organization of the rest of this paper is as follows. Section 2 details the building and training of a preliminary recognizer which uses a bank of phonetic feature modules to compute input feature vectors. In Section 3, we describe alternatives to the initially proposed voicing module and compare the performance and consistency of these modules at the feature level. Next, we present experimental results comparing recognition performance using the different voicing modules in Section 4. Finally, Section 5 contains conclusions and directions for future work.

2

Feature-based recognizer

In order to investigate the benefits of an approach where individual feature detectors can be independently improved, it was first necessary to build and train a recognizer which makes use of the outputs of such feature modules rather than traditional spectral feature vectors. This section details the procedure we followed to build such a recognizer for the Aurora TI digits task.

2.1 Training individual feature detectors

For this preliminary study, a set of six phonetic features was selected, and individual feature stream modules were created by training Gaussian mixture model (GMM) classifiers. The phonetic feature modules used are shown in Table 2.1. The input to each feature module is a sequence of MFCC derived feature vectors, $\mathbf{x}_t$, computed every 10 ms. The output of each feature stream module is the posterior probability of that particular feature being present at time t.

Feature	*Example TIMIT Labels*
Frication	s, sh, z, ...
Rounding	w, ow, uw, ...
Nasal	n, m, ng, ...
Liquid/Glide	el, l, uw, ...
Burst	g, k, p, ...
Voice	aa, ae, ah, ...

Table 2.1: Phonetic feature streams and corresponding phone labels used for feature-based Aurora recognizer

The feature modules were trained using 410 sentences spoken by 40 speakers taken from the TIMIT speech database. The speech data was first downsampled to 8 kHz, then segmented into 10 ms frames. For each feature **F**, each frame was labeled as either **+F** or **-F** depending on the identity of the time aligned phonetic label for that segment of speech. For example, all frames belonging to an /s/ segment were labeled as **+fricative**, whereas all frames belonging to an /uw/ segment were labeled as **-fricative**.

In total, approximately 126,000 frames were used for training. For each feature **F**, these binary labeled frames were then used to train two GMMs, $p(\mathbf{x}_t|+\mathrm{F})$ and $p(\mathbf{x}_t|-\mathrm{F})$. The input vectors $\mathbf{x}_t$ were computed by first concatenating a set of MFCC averages collected from regions, as far as 35 ms away, surrounding the time t, then reducing the resulting 80-dimension vector to 50 dimensions using a principal component analysis (PCA) rotation matrix.

During recognition or testing, the posterior probability that feature **F** is expressed in an input frame $\mathbf{x}_t$ can be computed using Bayes rule. Assuming equal priors,

$$p(+\mathrm{F}|\mathbf{x}_t) = \frac{p(\mathbf{x}_t|+\mathrm{F})}{p(\mathbf{x}_t|+\mathrm{F}) + p(\mathbf{x}_t|-\mathrm{F})}$$

Figure 2.1 shows an example of an Aurora utterance together with the feature streams computed by the MFCC GMM modules. One can see that some modules, such as **voicing** and **frication** appear to produce very credible output streams that correspond well to what is observed in the speech signal. Others, such as the **burst** and **nasal** modules, appear to be more erratic, contributing high probabilities even in regions where the feature is likely not present. At this stage, no attempt was made to improve the performance of these initial feature modules.

2.2 Integration of feature streams

The next step was to use the outputs of the feature stream module to build a digit recognizer suitable for the Aurora task. First, all speech data in the clean training set for the Aurora corpus was parameterized as feature stream probabilities using the feature detection modules. At each 10 ms time step, a six-dimension feature vector was created by concatenating the outputs of the six feature modules at that time. This six-dimension vector was then used as the input feature vector to train the default recognizer included in the Aurora corpus release. Using the HMM Toolkit (HTK) training template provided with the Aurora database, eleven whole word HMM models were trained: the digits "zero" through "nine" and "oh". Each whole word model consisted of an 18 state HMM, with each state containing a six-dimension, three-mixture GMM which modeled the state emission probability.

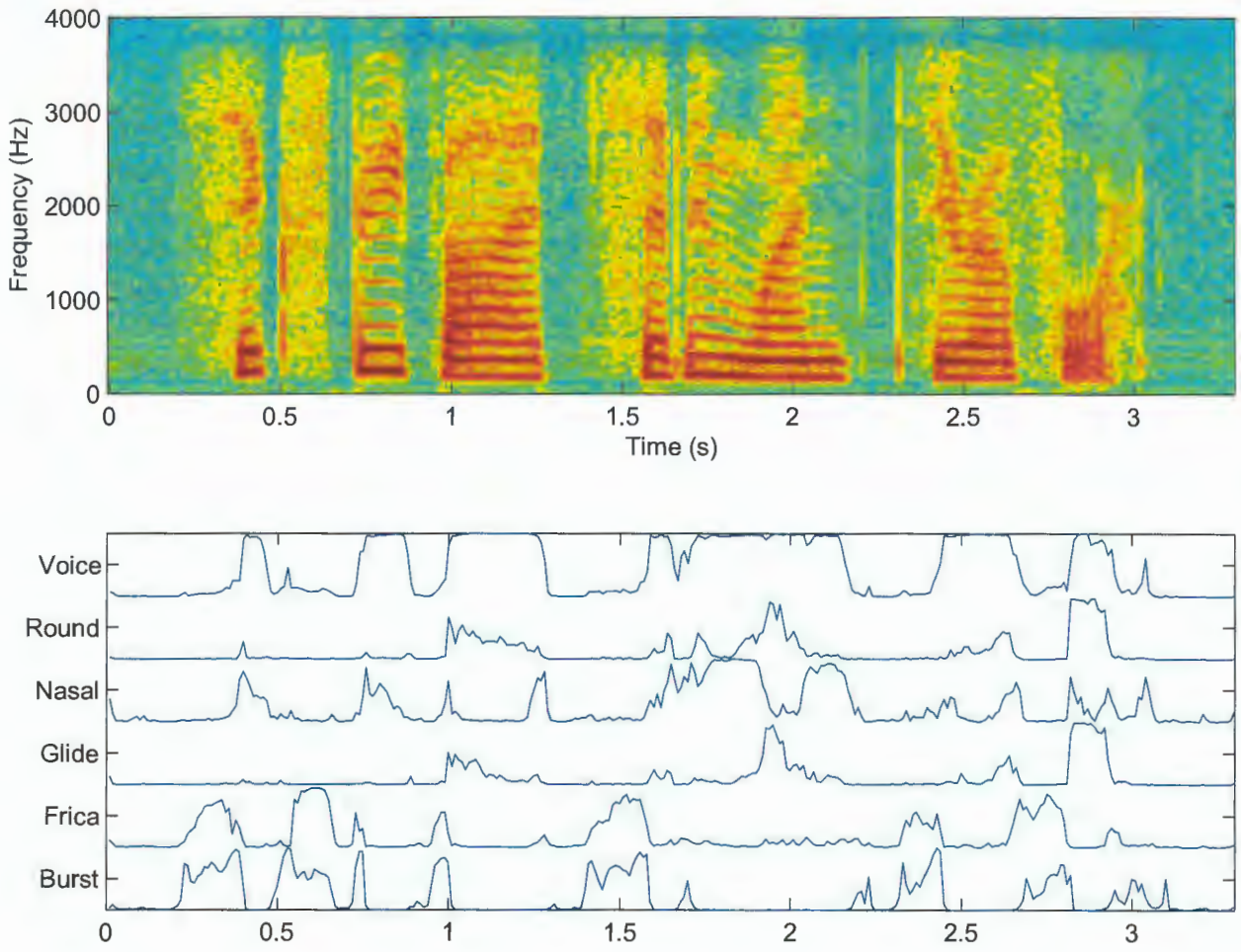

Figure 2.1: Spectrogram and sample feaure module outputs using the initial MFCC GMM feature modules for the Aurora utterance "six three five seven one zero four" with no additive noise

It should be noted here that the recognizer architecture used for these experiments is probably not optimal for the type of inputs being used. Ideally, a more knowledge-based system would be able to make better use of the relationships between the different phonetic features to determine appropriate segmentations and word identities. The choice to use an HMM recognizer in this work was mainly due to time constraints.

2.3 Preliminary recognition results

For testing, we used the 'testa' set from the Aurora corpus, which consists of 1001 spoken digit utterances presented under four different noise conditions (subway, babble, car, and exhibition hall) and eight different SNR levels (Clean, 20dB, 15dB, 10dB, 5dB, 0dB and -5dB). Despite the suboptimality of the recognizer architecture and the extreme sparsity of the feature vectors used as inputs to the HMM system, we were surprised to see that the word accuracy rate was fairly high ($\approx$ 87%) when tested on the clean data. Figure 2.2 shows the recognition accuracy rate of the feature stream recognizer averaged across 4 different noise conditions. Also shown is the word accuracy rate for the baseline Aurora recognizer included with the data set.

Although the baseline recognizer clearly outperforms the stream-based recognizer at all SNR levels, we were not overly discouraged by the performance gap for several reasons. First, no attempt was made to optimize individual stream processing modules, and the example stream outputs displayed in Figure 2.1 indicated that improvement is clearly needed for some of the modules. Second, compared to the Aurora baseline recognizer, which uses 39-dimension input vectors that include delta and delta-delta measurements, the feature-based recognizer uses extremely sparse inputs with no acoustic context from frame to frame. In order to provide a more balanced comparison, we plan to evaluate the performance of the baseline Aurora recognizer using input MFCC feature vectors that do not include delta and delta-delta coefficients. Another factor that favours the baseline Aurora recognizer is the use of diagonal variance GMMs in the HMM states, which are well suited for modeling the uncorrelated dimensions of the MFCC feature vectors. In contrast, the feature stream inputs, when observed as frame level vectors, have correlated dimensions due to the redundancy of some phonetic features.

We believe there is much room for improvement in the matched performance of the preliminary phonetic feature recognizer described. Some immediate possibilities include using asynchronous transition states, adding more phonetic feature streams, and improving the individual feature stream modules. This last direction is the focus of the remainder of this work.

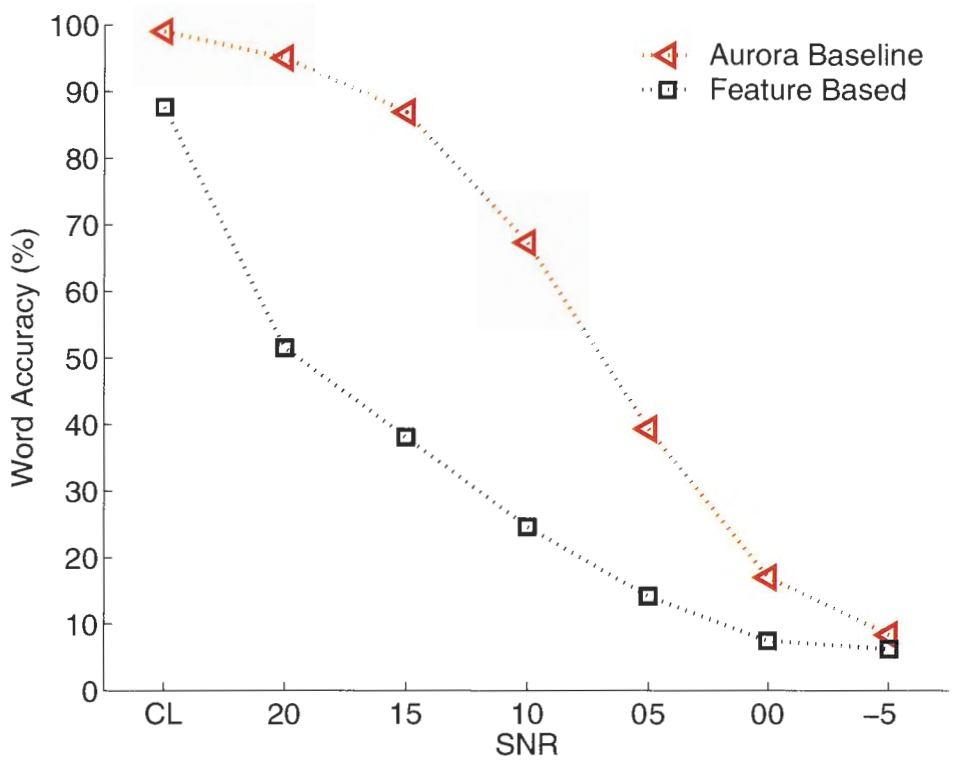

Figure 2.2: Comparison of recognition accuracy for Aurora baseline recognizer and preliminary feature-based recognizer

3

Feature module improvement

As mentioned in the previous section, one pathway to improving the performance of the feature-based recognizer is to improve the quality of the feature stream modules. In this work, we chose to focus on the voicing feature module, as voicing happens to be, for humans, one of the more robust cues extracted from the speech signal even under significant corruption due to additive noise.

Voicing is associated with periodicity in time, and harmonic spectral structure in frequency. Because of these special signal attributes, there have been many attempts to produce reliable voicing detectors using purely signal processing based approaches such as harmonic product spectrum, autocorrelation analysis, and cepstrum analysis [18]. In addition to these techniques, there have also been some approaches which make use statistical learning algorithms such as the multiband Bayesian network approach proposed by Saul *et al.* [16].

In this section, we detail efforts to improve the voicing feature module by utilizing two temporal signal processing measurements as well as a variation on our initial statistical GMM module which incorporates of non-MFCC measurements.

3.1 Alternative voicing feature modules

3.1.1 Autocorrelation

Autocorrelation is a measure of the similarity of a signal with time shifts of itself. Voiced signals exhibit strong autocorrelation values at time shifts that are multiples of the fundamental period of voicing. For a continuous time signal, $x(t)$, the normalized autocorrelation function is

$$R_{xx}[k] = \frac{\sum_{n=-\infty}^{\infty} x[n]x[n-k]}{\sum_{n=-\infty}^{\infty} x[n]^2}$$

If $x[n]$ is periodic with period N, $R_{xx}[k]$ will have peaks at $k = mN$. The normalizing term in the denominator of $R_{xx}[k]$ ensures that all values fall between between 0 and 1.

To create an autocorrelation-based voicing feature module, we computed the short-time normalized autocorrelation of the signal, $R_{xx}[k,n]$, using a sliding window of 64 ms, shifted every 10 ms. The output of the autocorrelation-based voicing module was then taken as the height of the maximum peak in the autocorrelation function for that time window.

$$v_{autocorr}[n] = \max_{k \neq 0} R_{xx}[k,n]$$

3.1.2 Sinusoid Uncertainty

Recently, Saul *et al.* proposed a new method for real-time pitch extraction and voicing detection using a novel sinusoid fitting approach [19]. First the signal is half-wave rectified to concentrate energy at the fundamental frequency for periodic regions. The signal is then low pass filtered and passed through a bank of octave spaced filterbanks to obtain a set of band-limited signals, $x_j[n]$. The premise of this approach is as follows: if $x_j[n]$ are discrete samples of a sinusoid with frequency ω_j, then they will obey the difference equation

$$x_j[n] = \frac{1}{\cos\omega_j}\left[\frac{x_j[n-1]+x_j[n+1]}{2}\right]$$

Using this difference equation, we can determine the minimum mean square error (MMSE) estimate, ω_j^*, of ω_j for the windowed segment of $x_j[n]$ between $[N-k,N+k]$ by minimizing the error function, $E_j(\alpha)$

$$\begin{aligned} E_j(\alpha) &= \sum_{n=N-K}^{N+K}\left[x_j[n] - \alpha\left(\frac{x_j[n-1]+x_j[n+1]}{2}\right)\right]^2 \\ \Rightarrow \alpha_j^* &= \frac{2\sum_n x_j[n](x_j[n-1]+x_j[n+1])}{\sum_n (x_j[n-1]+x_j[n+1])^2} \\ \Rightarrow \omega_j^* &= \cos^{-1}(1/\alpha_j^*) \end{aligned}$$

Thus, at each time step, each band j generates a pitch estimate ω_j^*. The value of ω_j^* with the smallest error is chosen as the pitch estimate, and the sharpness of the least square errors fit, $\Delta\mu_j^*$, is used as a measure of the uncertainty of the pitch estimate.

$$\Delta\mu_j^* = \frac{1}{\omega_j^*}\left(\frac{\cos^2\omega_j^*}{\sin\omega_j^*}\right)\left[\left(\frac{1}{E_j}\frac{\partial^2 E_j}{\partial\alpha^2}\right)\Big|_{\alpha=\alpha_j^*}\right]$$

The uncertainty measure is a dimensionless quantity that ranges from 0 to infinity, with smaller values indicating higher confidence in voicing. We scaled this quantity to range between 0 and 1 by taking the exponential of the negative of the uncertainty measure.

$$v_{sinun}[N] = \max_j \exp(-\Delta\mu_j^*)$$

Because the original uncertainty measure was negatively correlated with periodicity, this processing step ensures that $v_{sinun}[N]$ will be positively correlated with voicing.

3.1.3 Alternative Feature GMM

The benefits of the approaches described in the sections 3.1.1 and 3.1.2 are that they generate measurements which are directly relevant to the acoustic qualities of the feature stream they are trying to estimate. One disadvantage of the above approaches alone is that they do not make explicit use of information from neighbouring frames. Another difficulty is that these purely signal processing based approaches make no use of available training data, so the quantities computed, while scaled between 0 and 1, do not represent actual probabilities.

To address these drawbacks, we used the same training method used to train the MFCC GMM voicing feature detector, but replaced the MFCC inputs with features derived from the signal autocorrelation and sinusoid uncertainty. At every frame step, t, the maximum normalized autocorrelation and sinusoid uncertainty measurements from frames surrounding t were concatenated to form a 28-dimension vector which was reduced to six dimensions using a PCA rotation matrix. These six dimension vectors were then used to train the *alternative feature* GMMs (AF GMM), $p(\mathbf{x}_t|+\mathbf{voice})$ and $p(\mathbf{x}_t|-\mathbf{voice})$. Using these two GMMs, the posterior probability of voicing can then calculated as for the MFCC feature modules.

3.2 Evaluating feature modules using distortion

Initially, we attempted to evaluate the performance of the different voicing modules by calculating the detection error tradeoff characteristic for each module on a small TIMIT test set of 100 sentences. The voiced/unvoiced reference was determined using the phonetic transcription, and the equal error rate (error rate where the voiced to unvoiced error rate equals the unvoiced to voiced error rate) was used as a metric for comparison. The DET curves for the feature detectors evaluated on clean data are shown in Figure 3.1 and the associated equal error rates are shown in Table 3.1. Based on the observed detection performance, we could have concluded that the MFCC GMM module was the

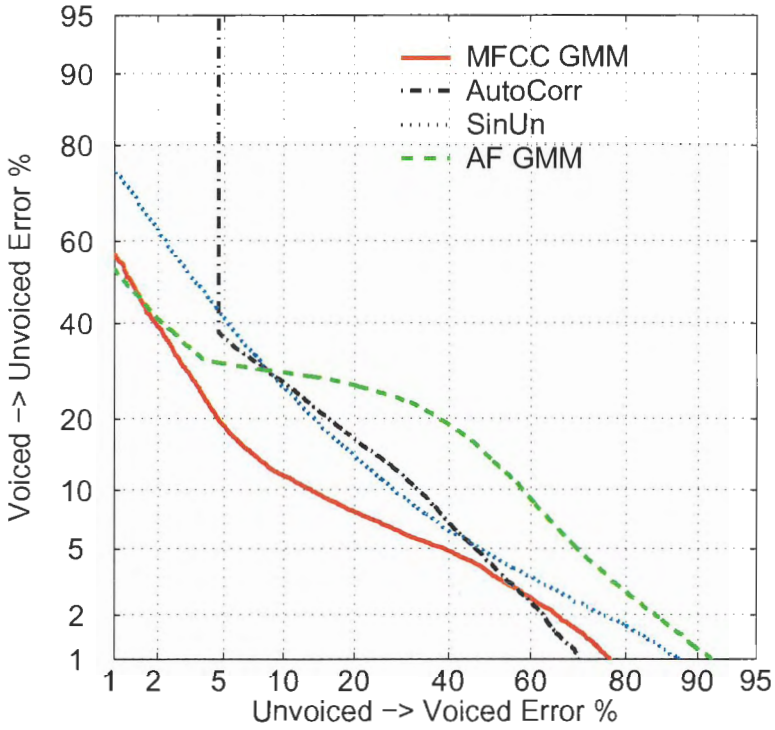

Figure 3.1: Detection Error Tradeoff curves comparing voicing detection performance of the different voicing modules

Voicing Feature module	*Equal Error Rate (%)*
MFCC-based GMM (MFCC GMM)	11.14
Autocorrelation (AutoCorr)	16.78
Sinusoid Uncertainty (SinUn)	18.11
Alternative feature GMM (AF GMM)	24.84

Table 3.1: Equal error rates for the four voicing modules evaluated on a small set of clean TIMIT data

most accurate followed by the Autocorrelation, Sinusoid Uncertainty, and finally the AF GMM. However, we had several concerns about using the detection performance as a criterion for evaluating the overall quality of the feature modules. First, the phonetic transcriptions used for the voicing reference only indicate the segment boundaries at the phone level and don't correspond to ground truth where voicing is concerned. Second, using a binary +/- voicing label quantizes the measurement we are actually seeking, which is a continuous measure of the voicing likelihood. Finally, unless the equal error rate is very low, global detection error rate does not give any sense of how closely the output for one utterance matches the output for the same utterance in noise.

As a second attempt at performing a meaningful evaluation of the different feature modules, we used a simple distortion measure to determine the consistency of extraction for a feature stream output under different noise conditions. Because Aurora data is essentially clean data with artificially added noise at different SNR levels, it is possible to compare the feature module outputs for the same utterance under different noise conditions. Figure 3.2 shows the output of the MFCC GMM voicing module for a particular utterance at different noise levels. Figure 3.3 shows the outputs of the AF GMM module for the same utterance. The second voicing module appears to have less discriminative power in separating the voiced and unvoiced frames, but appears to be more consistent across a range of noise conditions when compared to the MFCC GMM. This obser-

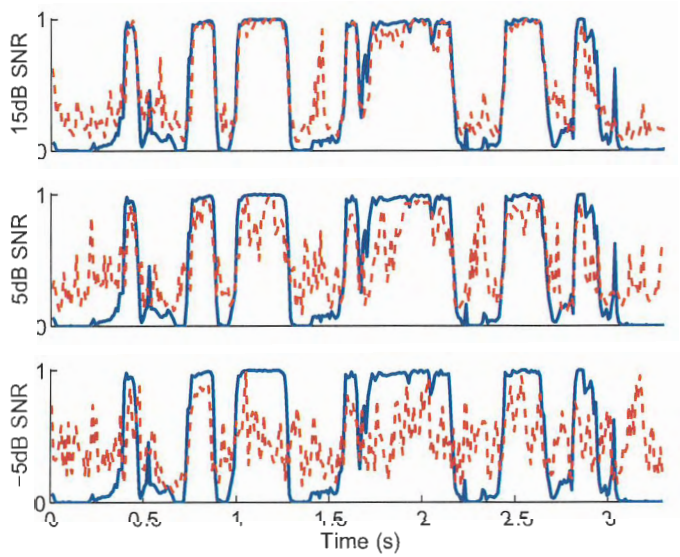

Figure 3.2: Comparison of MFCC GMM voicing module output on clean data vs. noisy data

vation can be quantified by using the absolute difference between frames for clean and noisy versions of a particular utterance. For voicing module i, if the output voicing

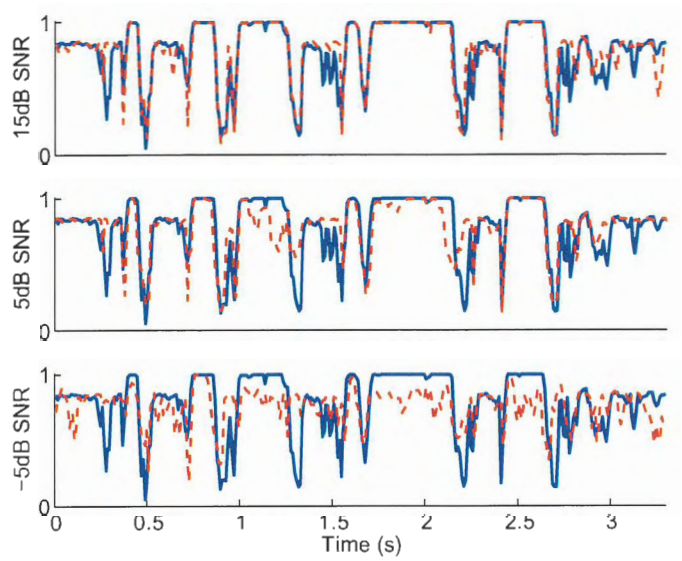

Figure 3.3: Comparison of AF GMM voicing module output on clean data vs. noisy data

stream on a clean waveform is $v_{i,c}(t)$, and the output voicing stream on the noisy version of the same waveform is $v_{i,n}(t)$, then the distortion at time t is given by

$$D(t) = |x_{i,c}(t) - x_{i,n}(t)|$$

Two distortion quantities were measured. First, we computed the average frame distortion for frames which had distortion ≤ 0.2. Second, we computed the percentage of overall frames which had distortion > 0.2. These frames were labeled as gross errors. The average frame distortion and frame gross error rate for the four voicing modules are illustrated in Figures 3.4 and 3.5.

Comparing the frame by frame similarity of the different modules under different noise levels revealed important differences about the behaviour of the modules under noisy conditions. Even at a relatively modest noise level of 20 dB SNR, the output of the MFCC GMM voicing module outputs exhibited significant distortion. Over 25% of frames were labeled as gross errors, and for the remaining frames, the average frame distortion was approximately 6.6%. As the noise level increased, both the gross error rate and frame distortion increased monotonically. At -5 dB SNR, the gross error rate was 76% and the average frame distortion was 11.2%.

In contrast, the AF GMM voicing module outputs exhibited much lower distortion and gross error rates across all six noise levels. While the sinusoid uncertainty module had the lowest distortion figures at high SNR levels, the AF GMM module had more consistently lower distortion across all noise levels. At 20 dB SNR, 10% of frames were

labeled as gross errors, and for the remaining frames, the average frame distortion was 3.4%. At -5 dB SNR, the gross error rate was 30% and the average frame distortion was 8.4%. Both of these figures significantly improved compared to the MFCC GMM module under the same conditions. The distortion results reveal significant differences

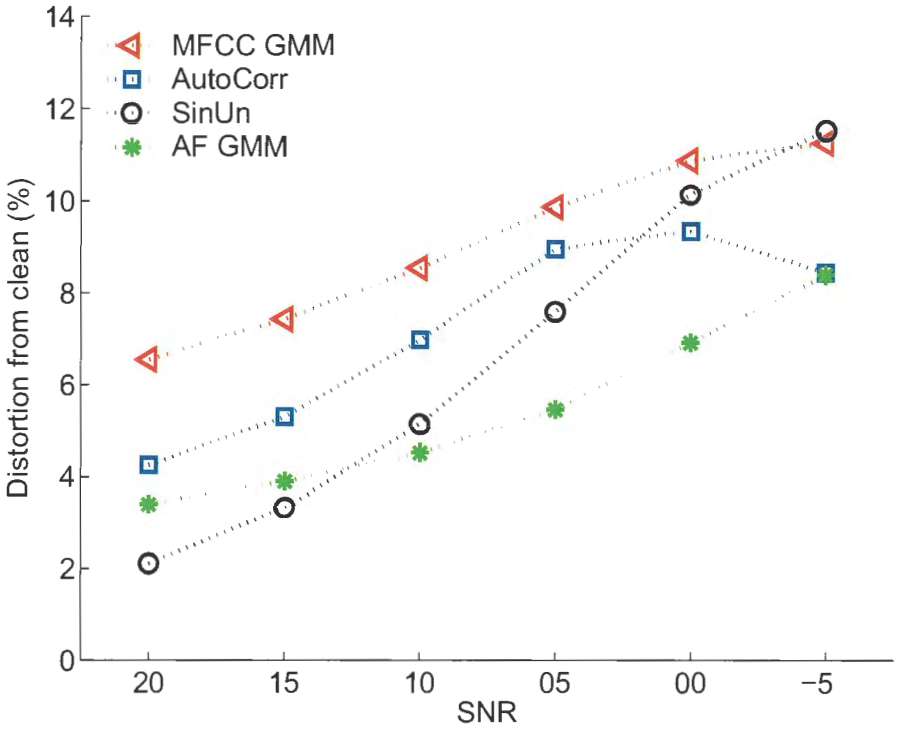

Figure 3.4: Average frame distortion across all noise conditions

in the characteristics of the different voicing module that weren't highlighted by the voicing detection task. Despite the lower detection performance of the AF GMM module, it is clear that the output of this module is considerably less erratic than that of the more accurate MFCC GMM module. The next section examines whether this improved consistency translates to better robustness in recognition.

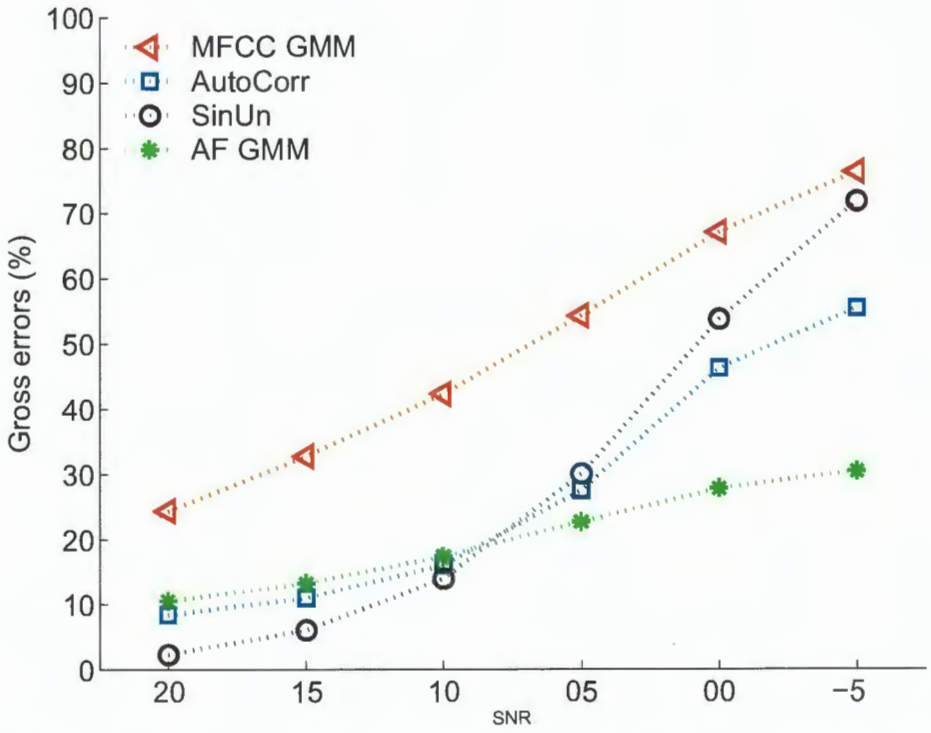

Figure 3.5: Percentage of frames labeled as gross errors across all noise conditions

4

Recognition Results

After using distortion to characterize the outputs of the different voicing feature modules, we performed recognition experiments to identify how the consistency and quality of the different voicing modules affected recognition performance.

4.1 Normal scenario results

Following the approach described in Section 2, we trained a separate HMM recognizer for each voicing module. The structure and training data for each the four recognizers was the same, with the only difference being the voicing module used to compute the voicing stream features.

We performed recognition experiments on the 'testa' set of the Aurora database using each of the four recognizers. For this first experiment, we simply used the feature stream outputs computed on the input test utterance as the input to the corresponding HMM recognizer. The recognition results for this normal scenario were averaged over the four noise conditions are shown in Figure 4.1.

Overall, no significant difference in accuracy were observed, although the recognizer using the MFCC GMM voicing module appeared to have the lowest performance in all conditions except clean. Based on these results alone, it was impossible to conclusively indicate the benefit of one voicing module over another. The similar performance of all four systems may be due to the effect of the other feature stream modules, which also experience heavy corruption due to noise. We performed additional oracle experiments in order to isolate the effect of the voicing module on the recognition task.

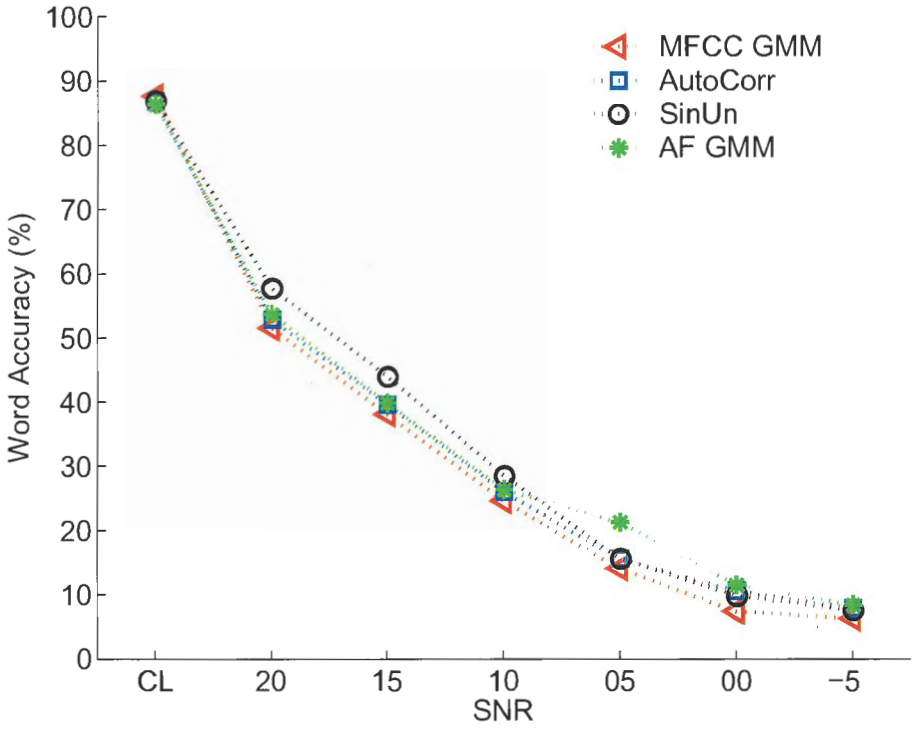

Figure 4.1: Word accuracy for normal scenario experiment

4.2 Oracle scenario results

In this experiment, we attempted to control the output of the voicing module as the noise level varied to see if the quality of the voicing module plays a role in improving recognition accuracy. As observed in Section 3, the MFCC GMM voicing module had the best voicing detection performance in clean data. To see if this detection performance translated to better recognition accuracy, we assumed perfect consistency for each voicing module by using the clean voicing feature stream for recognition at each noise level. The framework for this experiment is illustrated in Figure 4.2, and the resulting recognition accuracies are shown in Figure 4.3. On examining these results, as in the previous recognition experiment we found it difficult to make any kind of meaningful comparison between the four systems due to the similar recognition performance at all noise levels. This result is likely due to the influence of the non-voicing module feature streams, which undergo significant corruption due to noise, but are presented identically to the four recognition systems. Thus the overall effect of these streams likely overpowers any improvement that might arise from the clean estimation of the voicing stream.

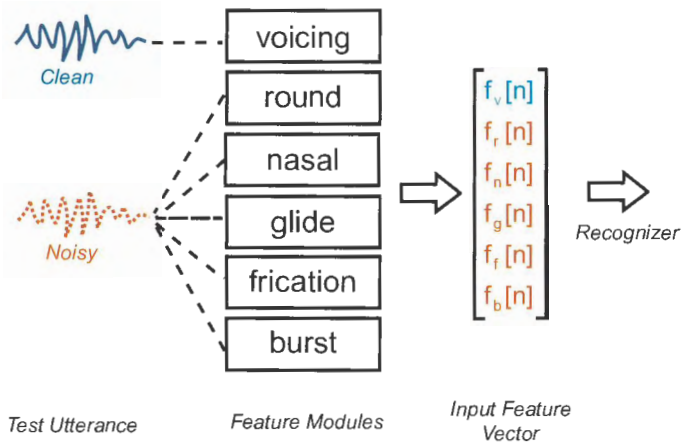

Figure 4.2: Setup for oracle recognition experiment

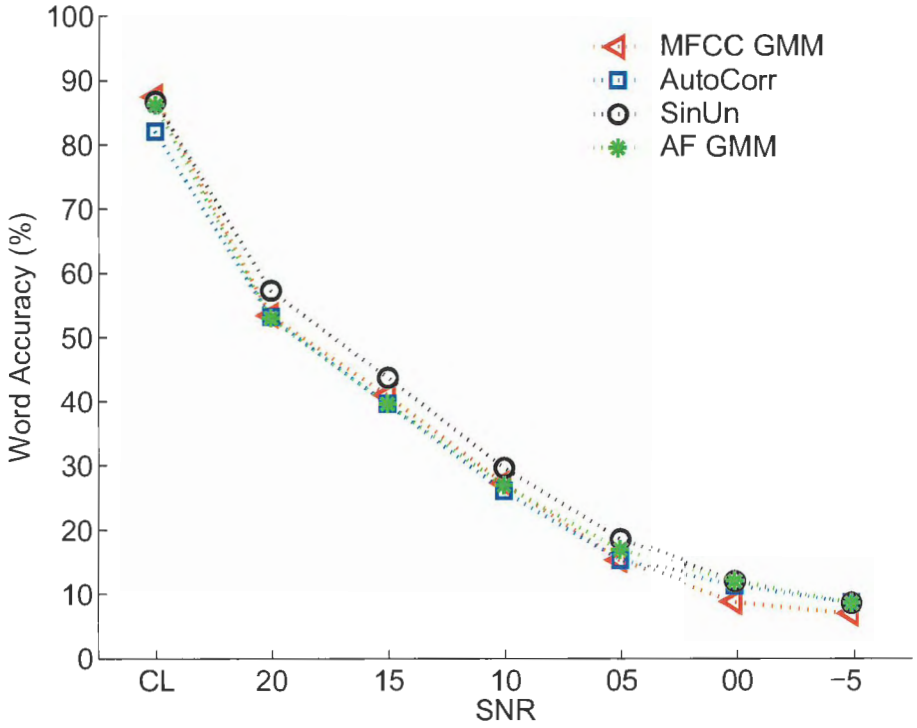

Figure 4.3: Word accuracy for oracle scenario experiment

4.3 Inverse oracle scenario results

In the previous experiment, as in the normal scenario, we failed to observe significant differences in recognizer performance despite controlling for the consistency of the voicing module. We hypothesized that a major reason for the indistinguishability of recognition results was due to significant corruption in the other streams. To control for this factor, we performed an *inverse* oracle experiment by feeding clean feature stream data for all modules except the voicing module. The framework for this experiment is shown in Figure 4.4. The net effect of this idealized scenario is to control for the recognition degradation due to corruption in the non-voicing streams. The recognition results for this experiment are shown in Figure 4.5.

In this experiment, we observed a marked difference in performance between the four systems. For the system using the MFCC GMM voicing module, we observed significant degradation in recognition performance as the SNR level decreased. This degradation occured despite the fact that only the voicing stream was affected by the noise. We can conclude that for this experiment, the decrease in recognition accuracy from 87.6% (clean) down to 21.8% (-5 dB SNR) is directly attributable to the inconsistent extraction of the voicing feature for the MFCC GMM module in increasing noise levels. By contrast, the systems making use of the other three voicing modules are much more resistant to noise, exhibiting no significant degradation in accuracy for SNR levels above 10 dB. Overall, the system employing the Alternative GMM voicing module has the least degradation due to noise, going from 86.3% accuracy in clean conditions to 66.6% accuracy in -5 dB. This represents a significant improvement when compared with the performance of the MFCC GMM module. We noted that the recognition results are closely correlated with the distortion results obtained in Section 3. This observation indicates that evaluating reliability at the feature level may be useful for improving overall recognition performance.

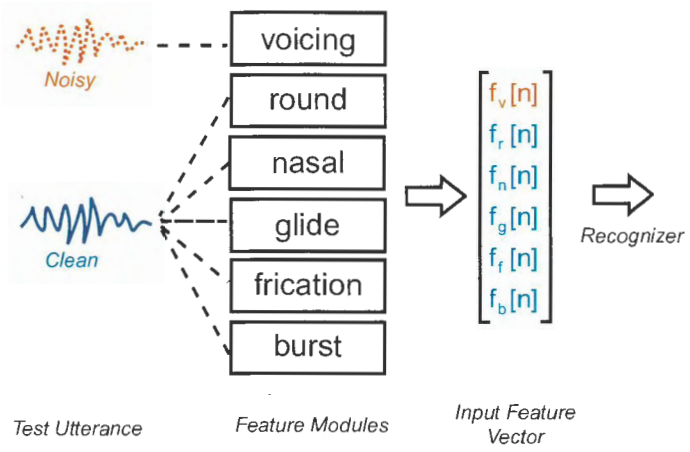

Figure 4.4: Setup for inverse oracle recognition experiment

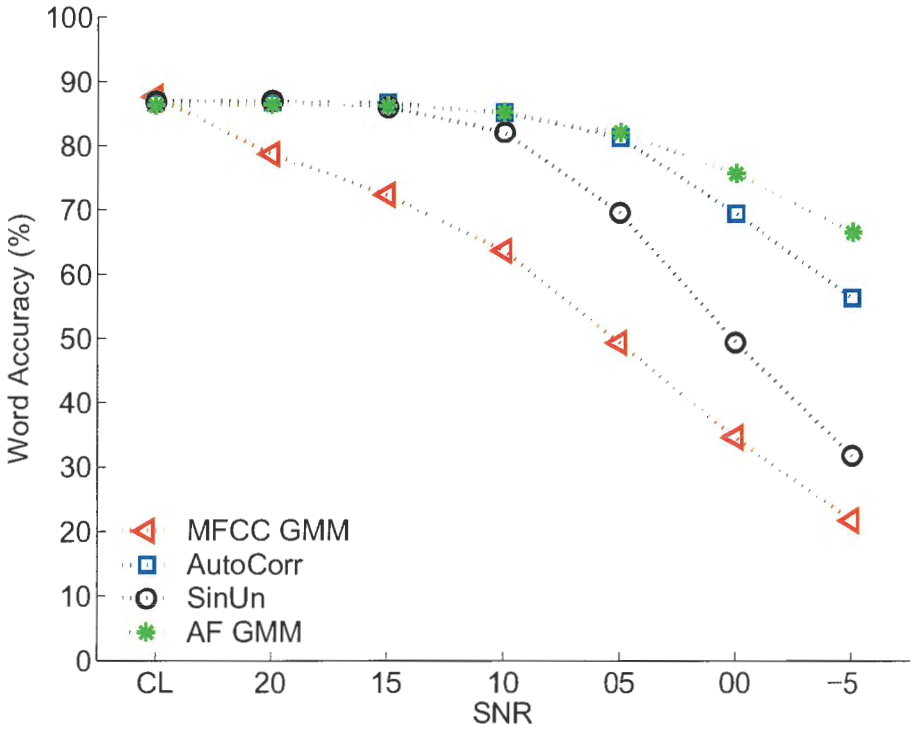

Figure 4.5: Word accuracy for inverse oracle scenario experiment

5

Conclusions and Future Work

Several conclusions can be obtained from this work. First, in building the preliminary feature-based recognizer, we determined that using a small set of phonetic feature streams as input to an HMM system can result in surprisingly high ($\approx$ 87%) recognition accuracy for a constrained digit task in non-noisy environments.

Based on our attempts to optimize the voicing feature module, we found that combining statistical training with feature-specific measurements can improve consistency for feature stream extraction. By evaluating the voicing detection performance, we also noted that consistency does not necessarily correspond to good detection. Another reason to avoid depending exclusively on detection accuracy is that the voicing references used to determine these detection rates are based on phonetic transcriptions and not the actual voicing state. Thus, the "ground truth" used for both training and testing in this case is only approximately correct.

The recognition experiments and oracle experiments performed in Section 4 indicated that feature stream corruption can lead to drastic degradations in recognition accuracy, even when the voicing stream is perfectly consistent. Furthermore, by performing inverse oracle experiments, we observed that the inconsistency of the MFCC GMM voicing module output leads to significant performance degradation in noise, even when the other streams are computed from clean data. The AF GMM voicing module, which had the lowest distortion metric across the different noise conditions, also had the least degradation in recognition accuracy in this condition. This result is encouraging as it indicates that we may be able to improve the overall performance of the recognizer by separately improving the other feature detectors in a similar manner.

5.1 Future Work

Much future work remains to be done. Although the voicing modules investigated were more consistent than the MFCC GMM module, they were not designed with any kind of noise resistance in mind. To that end, feature modules explicitly designed for noise robustness, like the Bayesian network sonorant detector described in [19], should be pursued.

Moving beyond the voicing feature module, we believe that it is critical to work on improving the other feature detectors in the same manner as demonstrated here - by using signal measurements that are directly relevant to the type of feature being computed and combining these measurements with statistical training to improve generality and reliability.

We have mentioned that the HMM framework used for these experiments is likely not optimal for the features streams being used. Improved recognition results may be obtained by integrating these feature streams with more appropriate temporal modeling structures such as feature based graphical models or DBNs. The phonetic feature streams may also provide valuable discriminative measurements that could be used in a segment-based recognition framework.

We feel that an important extension for any feature stream recognition framework is to pair the stream outputs with a reliability measure of some sort. As demonstrated in the oracle experiments, corrupted feature streams can severely degrade recognition accuracy, so it is useful to know if and when a feature stream is no longer reliable. For example, a running estimate of SNR can be used to discount the reliability of certain noise-vulnerable features as the SNR level goes down. Another possibility is to monitor the modulation rate of the feature module outputs to ensure that the rate of change is consistent with speech patterns. For example, if a burst detection module is continually triggered, then the contribution from this module should be discounted or ignored completely.

Bibliography

[1] M. Gales and S. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 352–359, September 1996.

[2] S. Greenberg and B.E.D. Kingsbury, "The modulation spectrogram: In pursuit of an invariant representation of speech," in *Proc. ICASSP*, Munich, Germany, May 1997, pp. 1647–1650.

[3] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, October 1994.

[4] P. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 1996.

[5] B.J. Frey, L. Deng, A. Acero, and T. Kristjansson, "Algonquin: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition," in *Proc. Eurospeech*, Aalborg, Denmark, September 2001.

[6] B. Raj, *Reconstruction of Incomplete Spectrograms for Robust Speech Recognition*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, April 2000.

[7] M. Cooke, P. Green, and M. Crawford, "Handling missing data in speech recognition," in *Proc. ICSLP*, Yokohama, Japan, 1994, pp. 1555–1558.

[8] L. Deng, G. Ramsay, and D. Sun, "Production models as a structural basis for automatic speech recognition," *Speech Communication*, vol. 33, no. 2-3, pp. 93–111, August 1997.

[9] K. Livescu, J.R. Glass, and J. Bilmes, "Hidden feature models for speech recognition using dynamic Bayesian networks," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

[10] K. Kirchoff, "Syllable-level desynchronisation of phonetic features for speech recognition," in *Proc. ICSLP*, Philadelphia, PA, 1996.

[11] F. Metze and A. Waibel, "A flexible stream architecture for ASR using articulatory features," in *Proc. ICSLP*, Denver, CO, 2002.

[12] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," in *Proc. ICASSP*, Istanbul, Turkey, 2000.

[13] C.-H. Lee, "On automatic speech recognition at the dawn of the 21st century," *IEICE Transactions on Information and Systems*, vol. E86-D, no. 3, pp. 377–397, March 2003.

[14] B. Launay, O. Siohan, A.C. Surendran, and C.-H.Lee, "Towards knowledge-based features for HMM based large vocabulary automatic speech recognition," in *Proc. ICASSP*, Orlando, FL, 2002.

[15] J.R. Glass and V.W. Zue, "Detection and recognition of nasal consonants in American English," in *Proc. ICASSP*, Tokyo, Japan, 1986.

[16] L. K. Saul, M. G. Rahim, and J. B. Allen, "Learning from examples in critical bands of speech," in *Proc. of IEEE ASRU Workshop*, Keystone, CO, 1999.

[17] P. Niyogi and M.M. Sondhi, "Detecting stop consonants in continuous speech," *J. Acoust. Soc. Am.*, vol. 111, no. 2, pp. 1063, February 2002.

[18] P.C. Bagshaw, S.M. Hiller, and M.A. Jack, "Enhanced pitch tracking and the processing of F0 contours for computer aided intonation teaching," in *Proc. Eurospeech*, Berlin, Germany, September 1993, vol. 2, pp. 1003–1006.

[19] L. K. Saul, D. D. Lee, C. L. Isbell, and Y. LeCun, "Real time voice processing with audiovisual feedback: Toward autonomous agents with perfect pitch," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds., vol. 15. MIT Press, Cambridge, MA, 2003.