IMPLEMENTATION AND ANALYSIS OF SPEECH RECOGNITION FRONT-ENDS

V. Mantha, R. Duncan, Y. Wu, and J. Zhao

Signal Group
Department of Electrical and Computer Engineering
Mississippi State University, Mississippi State, Mississippi, 39762
{mantha,duncan,wu,zhao}@isip.msstate.edu

ABSTRACT

We have developed a standard comprehensive front-end module for a speech recognition system. Several standard front-ends have been implemented, including mel cepstra, perceptual linear prediction, filter bank amplitudes, and delta features. The framework for this system was carefully designed to ensure simple integration with the speech recognition system. The modular design of the software along with an intuitive GUI allow a student of speech processing to easily interchange algorithms and vary every aspect of each model parameter, providing a powerful tutorial. The code itself is written in tutorial fashion, with a direct correlation between algorithmic lines of code and equations in the technical paper. The effectiveness of the different front-end algorithms has been evaluated on a common set of speech data.

1. INTRODUCTION

Before a computer can recognize human speech with current technology, the speech must first be processed into observation vectors representing events in the probability space [14]. This process, known as signal modeling, is the function of the front-end module. Using these acoustic observation vectors and some language constraints, a network search algorithm (performed by a decoder) finds the most probable sequence of events to hypothesize the textual content of the audio signal [14].

This paper describes the development and evaluation of a standard comprehensive front-end module for a speech recognition system. Several standard front-ends have been implemented, including mel cepstral, perceptual linear prediction, filter bank amplitudes, and delta features. The framework for this system was carefully designed to ensure simple integration with the speech recognition system [5]. The code itself is written in tutorial fashion, with a direct correlation between algorithmic lines of code and equations in this technical paper. This report aims to describe the signal processing algorithms used in the ISIP front-end.

2. HISTORICAL PERSPECTIVE

In order for the front-end to model useful observation vectors for speech recognition, it must extract important features from the speech waveform that are relatively insensitive to the talker and channel variability which is unrelated to the message content [10]. The algorithms used by the front-end are composed almost entirely of standard signal processing techniques, such as digital filter banks, linear predictive coding, and homomorphic analysis. These algorithms are successful because they model the speech signal consistently with the human auditory perceptual system —in the frequency domain [2]. Specifically, the short time spectral envelope is needed since speech is a time-variant signal [16]. Furthermore, the addition of physiological knowledge of the vocal articulatory system can be applied to the problem in order to increase recognition performance [10].

There are advantages and disadvantages to each algorithm described in this paper. For example, while linear prediction (LP) coefficients can generally be computed with fewer resources, the compressive nature of the transformation makes the model less robust to noise. Most current state of the art systems use one energy coefficient, twelve Fourier transform-derived cepstral coefficients, and delta and delta-delta derivatives of the first thirteen coefficients.

3. OVERVIEW

This report is broken into two sections. First, an overview of the general system structure is discussed. This section focuses mainly on the pre- and post-processing, with only a cursory scan of the modeling algorithms. This section also describes how the front-end is interfaced to the full speech recognition system. The second part of the report provides an in depth look at the algorithms which form the heart of the system, a description of the graphical user interface designed for this project, and the evaluation of these algorithms.

4. SYSTEM STRUCTURE

The modular design of the front-end is shown in Figure 1. After pre-processing (windowing and pre-emphasis are not shown on the diagram), three basic operations can be performed on the speech signal. These general algorithms are filter bank amplitudes (FBA), the Fourier transform (FFT), and linear prediction (LP) [16]. From the digital filter bank a power estimation may be directly computed. Perceptual linear prediction (PLP) is a post-processing step for LP coefficients, acting as a cascaded filter. The FT, LP, and PLP algorithms compute the spectrum of the signal, which is then processed into usable spectral parameters in one of two ways. The first method is filter bank amplitudes, similar to the general FBA algorithm which operated on the original signal. It computes a reduced number of averaged sample values from the spectrum. Computing the cepstrum is an alternate method of processing this spectrum. The details of these algorithms are further described in the next section.

4.1. Windowing and I/O

In order to extract short-term frequency information from a time-varying speech signal, a window function must be applied. The simplest window function is rectangular in shape; however, oftentimes more complicated shapes produce a more desirable windowed signal [17]. For speech processing, the Hamming window is used almost exclusively [14]. The Hamming window is a special form of the general Hanning window, shown in equation (1), with $\alpha_w = 0.54$.

$$w(n) = \frac{\alpha_w - (1-\alpha_w)\cos(2\pi n/(N_s-1))}{\beta_w} \quad (1)$$

The user can vary the window duration, window type, and frame duration. A physiological investigation into the human auditory system reveals the quickest movements of the vocal articulators are on the order of 10 ms. This means if the speech signal is averaged and evaluated (framed) every 10 ms, almost no information will be lost. Since the window duration is longer than the frame duration, efficient buffering algorithms reduce the I/O complexity of the task by only reading in a single frame of data at each time step. Compared to the decoding phase of speech recognition, a front-end's computational cost is negligible [14]. Nevertheless, poorly written code at any stage in the process can bog down a production system run in real-time.

4.2. Coefficient Concatenation

All coefficients from the various algorithms are concatenated into a single observation vector for each frame. To interpret the meaning of a number from its position, sequentially add up the number of each specified coefficient. For example, if energy and twelve FFT-derived cepstral coefficients are specified, the first number output is the energy, the fifth number is the fourth cepstral coefficient, etc. This is an efficient method for passing parameters to the network search algorithm because it decouples the signal modeling information into a vector of pure numbers for pattern recognition. The decoder need only be trained on the same coefficients as the test data.

4.3. Vector Post-Processing

Higher order time derivatives of the signal measurements can be added to better characterize temporal variations in the signal. Since the measurements previously described operate on a single window of data, they are considered zeroth order derivatives. First and second derivatives are now commonly used in speech recognition systems.

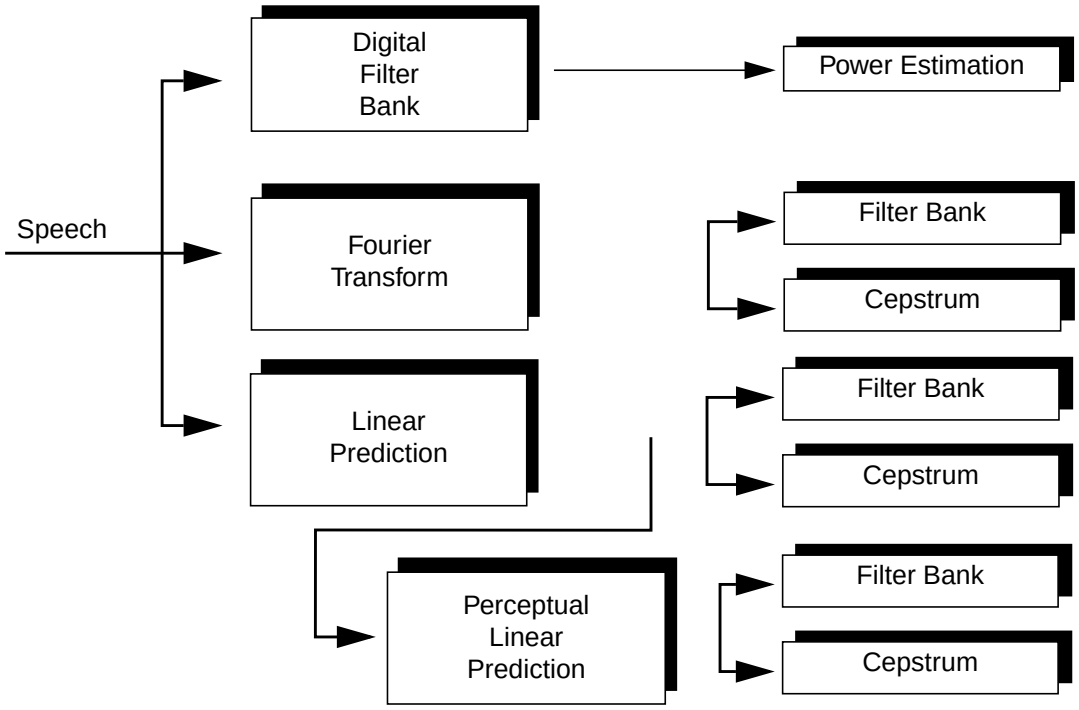

Figure 1. System block diagram

5. SIGNAL MODELING ALGORITHMS

The algorithms described in this section (with the exception of delta features) operate on a single window of speech data. The code itself is written in clear and simple form, referencing blocks of code directly to the equations described in this section where applicable. New signal modeling algorithms are added at this point in the structure.

5.1. Filter Bank Amplitudes

The digital filter bank is one of the most fundamental concepts in speech processing. A filter bank can be regarded as a crude model of the initial stages of transduction in the human auditory system. Each filter in the digital filter bank is usually implemented as a linear phase filter. The filter equations for a linear phase filter implementation can be summarized as follows:

$$s_i(n) = \sum_{j=\left(-\left(N_{FB_i}-1\right)\right)/2}^{\left(N_{FB_i}-1\right)/2} a_{FB_i}(j)s(n+j), \qquad (2)$$

where $a_{FB_i}(j)$ denotes the j^{th} coefficient for the i^{th} critical band filter. The number of filter banks normally is odd when implementing linear phase filters. The basic merit of the algorithm is that certain filter outputs can be correlated with certain classes of speech sounds.

The output of filter bank analysis is a vector of power values for each frame of speech data. Usually these values are combined with other parameters, such as mean energy, to form the final signal measurement vector. Since the analysis is based entirely on linear processing, the technique is generally robust to ambient noise.

5.1.1.Fourier Transform-Derived Coefficients

Simple Fourier transform-based filter banks designed for front-ends obtain the desired frequency resolution on a mel-scale (the mel-scale is described on page 4). To implement this filter bank, the window of speech data is transformed into the frequency domain by the Fourier transform. The magnitude of the spectral coefficients are then binned through correlation with triangular filters equally spaced on the mel-scale [19]. As defined here, binning means that each spectral

coefficient is multiplied by the corresponding filter gain; the bin value is the accumulation of every such product. Thus, each filter bank coefficient represents the average spectral magnitude in the filter channel,

$$S_{avg}(f) = \frac{1}{N_s}\sum_{n=0}^{N_s} w_{FB}(n)|S(f)|, \qquad (3)$$

where N_s represents the number of samples used to obtain the averaged value, $w_{FB}(n)$ represents a weighting function (filter gain), and $S(f)$ is the magnitude of the frequency response computed by the FFT.

5.1.2. Linear Prediction-Derived Coefficients

Linear predictive (LP) analysis is an estimate of the autoregressive all-pole model $A(w)$ of the short-term power spectrum of speech $P(w)$. Alternately, LP analysis is a means for obtaining the smoothed spectral envelope of $P(w)$. The major disadvantage of the LP model in speech analysis is that $A(w)$ approximates $P(w)$ equally well at all frequencies of the analysis band. This property is inconsistent with human hearing, which tends to be nonlinear above 800 Hz. Consequently, LP analysis does not preserve or discard the spectral details of $P(w)$ according to auditory prominence. The perceptual linear prediction algorithm, described in section [5.3], improves the basic LP model.

The spectrum is computed through application of the Fourier transform to the linear prediction coefficients. Since there are fewer points in the LP model, this approach is more efficient. From this LP-derived spectrum, filter banks are applied in exactly the same way as for the FT-derived spectrum. These coefficients are known as LP-derived filter bank amplitudes. A comparison of the LP-derived spectrum with the Fourier spectrum is given in Figure 2. From this we can observe that the LP-derived spectrum is not very robust to noise and is unable to model the second peak (also called the second formant) in the presence of noise.

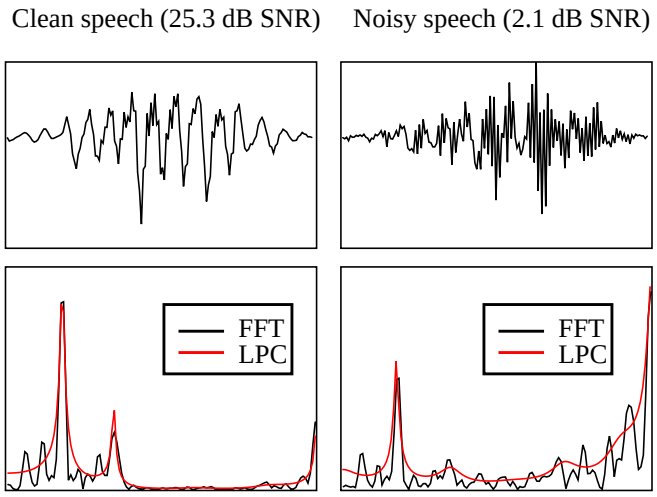

Figure 2. LP vs Fourier spectrum

5.2. Mel Frequency Cepstral Coefficients

A mel is a psychoacoustic unit of measure for the perceived pitch of a tone, rather than the physical frequency. The correlation of the mel to the physical frequency is not linear, as the human auditory system is a nonlinear system. A mapping between the mel scale and real frequencies was empirically determined by Stevens and Volkman in 1940 [14]. The scale is roughly linear below 1000 Hz, then decays logarithmically. It is described mathematically as:

$$Mel(f) = 2595\log_{10}(1 + f/700). \qquad (4)$$

This nonlinear scale is invaluable to speech coding in that it reduces the sample space with minimal perceptual loss. In practice, filters banks are evenly spaced along the mel scale. An overlay of the highest six triangular filters on the spectrum of a speech segment is shown in Figure 3. The bars below this figure represent the filter bank amplitudes [19].

A homomorphic system is useful for speech processing because it offers a methodology for separating the excitation signal from the vocal tract shape [14]. One space which offers this property is the cepstrum, computed as the inverse discrete Fourier transform (IDFT) of the log energy [3]. This signal is by definition minimum phase, another useful property. Cepstral coefficients are computed by the following equation:

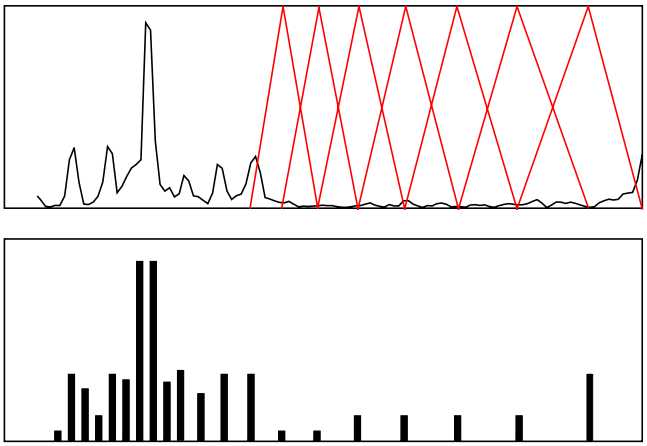

Figure 3. Mel-frequency spaced triangular filters

$$c(n) = \frac{1}{N_s}\sum_{k=0}^{N_s} \log\left|S_{avg}(k)\right| e^{j\frac{2\pi}{N_s}kn} \qquad 0 \le n \le N_s - 1 \,, \quad (5)$$

where $S_{avg}(k)$ is the average signal value in the k^{th} filter channel. In practice,fsfs the discrete cosine transform may be used in lieu of the IDFT for computational efficiency.

A critical analysis of the cepstral variability across different speakers and channel conditions leads to a more robust acoustic model for automatic speech recognition. The higher order cepstral coefficients are more influenced by algorithmic artifacts of the LPC analysis (the all-pole constraint, for instance). Alternately, the low cepstral coefficients vary primarily due to variations in transmission, speaker characteristics, and vocal efforts [16]. A liftering procedure,

$$w(n) = \begin{cases} 1 + \left(\frac{L}{2}\right)\sin\frac{n\pi}{L} & n = 1, 2, \ldots, L \\ 0 & n \le 0, n > L \end{cases}, \quad (6)$$

is used to weight the cepstrum and control the non-information bearing variabilities. For telephone bandwidth speech, typically L is set to 24 [19].

Most state-of-the-art speech recognition systems use a front-end comprising of 12 Fourier transform-derived mel-frequency cepstral coefficients and mean energy as a first order model of the signal.

5.3. Perceptual Linear Prediction

Perceptual linear predictive (PLP) analysis is a relatively new method for the analysis of speech signals. It is an improvement over the widely used LP (Linear Predictive) analysis. In PLP analysis, the all-pole modeling is applied to an auditory spectrum derived by (a) convolving $P(w)$ with a critical band masking pattern, followed by (b) resampling the critical band spectrum at approximately l Bark intervals, (c) pre-emphasis by a simulated fixed equal loudness curve, and finally (d) compression of the resampled and pre-emphasized spectrum through the cubic root non-linearity, simulating the intensity-loudness power law. The low order all-pole model of such an auditory spectrum has been found to be consistent with several phenomena observed in speech perception [9]. The block diagram of PLP Analysis is shown in Figure 4.

After windowing, the real and imaginary components of the short-term speech spectrum are squared and added to get the power spectrum,

$$P(w) = Re[S(w)]^2 + Im[S(w)]^2 . \quad (7)$$

The spectrum $P(w)$ is warped along its frequency axis into the Bark frequency Ω by

$$\Omega(w) = 6\ln\left\{(w/(1200\pi)) + [w/(1200\pi)^2 + 1]^{0.5}\right\} \quad (8)$$

where w is the angular frequency in rad/s. The resulting warped power spectrum is then convolved

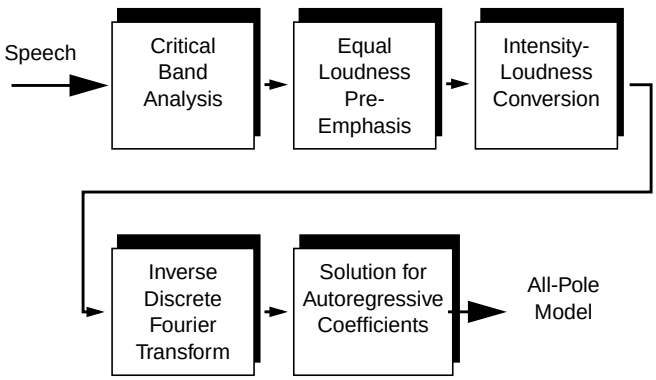

Figure 4.Block Diagram for PLP Analysis

with the power spectrum of the simulated critical band masking curve $\psi(\Omega)$,

$$\psi(\Omega) = \begin{cases} 0 & ,\Omega < -1.3 \\ 10^{2.5(\Omega + 0.5)} & ,-1.3 \le \Omega \le -0.5 \\ 1 & ,-0.5 \le \Omega \le 0.5 \\ 10^{-1.0(\Omega - 0.5)} & ,0.5 \le \Omega \le 2.5 \\ 0 & ,2.5 < \Omega \end{cases} . \qquad (9)$$

The discrete convolution of $\psi(\Omega)$ with (the even symmetric and periodic function) $P(w)$ yields samples of the critical band power spectrum,

$$\theta(\Omega_i) = \sum_{i = -1.3}^{2.5} P(\Omega - \Omega_i) \cdot \psi(\Omega) . \qquad (10)$$

This convolution significantly reduces the spectral resolution of $\theta(\Omega)$ in comparison with the original $P(w)$. This also allows for down sampling.

The sampled $\theta(\Omega(w))$ is pre-emphasized by a simulated equal loudness curve,

$$\Xi[\Omega(w)] = E(w) \bullet \dot{\theta}[\Omega(w)], \qquad (11)$$

where $E(w)$ is an approximation to the nonequal sensitivity of human hearing at different frequencies and simulates the sensitivity of human hearing at about the 40 dB level. The particular approximation is given by:

$$E(w) = \frac{(w^2 + k_1)w^4}{(w^2 + k_2)^2(w^2 + k_3)}, \qquad (12)$$

where $k_1 = 56.8 \times 10^6$, $k_2 = 6.3 \times 10^6$, and $k_3 = 0.38 \times 10^9$. This pre-emphasized function is then amplitude compressed using cubic root amplitude compression.

In practice, the convolution and preemphasis are carried out for each sample of $\Xi(\Omega_k)$ in the $P(w)$ domain by one weighted spectral summation per spectral sample $\Xi(\Omega_i)$. Thus the spectral sample $\Xi[\Omega(w_i)]$ is then given as

$$\Xi[\Omega(w_i)] = \sum_{w = w_{il}}^{w_{ih}} \omega_i(w) P(w) \qquad (13)$$

The limits in the summation and the weighting functions ω_i are computed from Equations (9), (11), and (14) using the inverse of (8), which is given by

$$w = 1200\pi \sinh\left(\frac{\Omega}{6}\right) \qquad (14)$$

The final operation of PLP analysis is the approximation of $\theta(\Omega)$ by the spectrum of an all-pole model using the autocorrelation method of all-pole spectral modeling [12]. The principle is to apply the inverse discrete Fourier transform (IDFT) to $\theta(\Omega)$ and find the dual of its autocorrelation function. The IDFT is the better choice here than the inverse FFT, since only a few autocorrelation values are needed. The first $(M + 1)$ autocorrelation values are used to solve the Yule-Walker equations for the autoregressive coefficients of the Mth-order all-pole model. These PLP coefficients can be processed through the same methods as standard LP coefficients to extract observation vectors.

The PLP-derived spectrum is more robust to noise compared to the LP-derived spectrum. This is illustrated in Figure 5. It may be observed that the PLP-derived spectrum is able to model the second formant in regardless of noise whereas the LP-derived

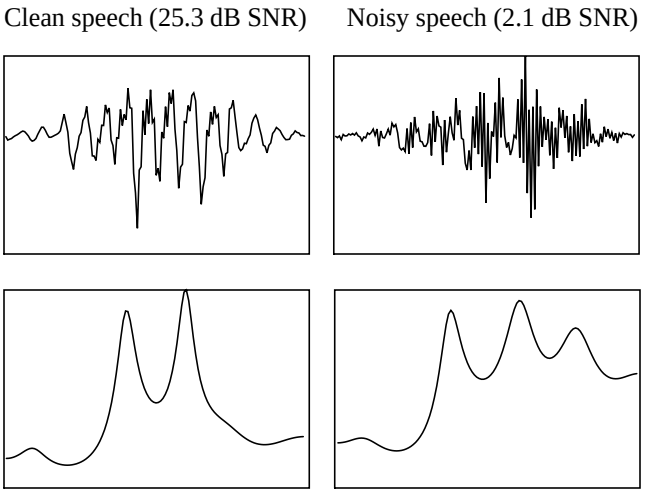

Figure 5.PLP-derived spectrum

spectrum was unable to as shown previously in Figure 2.. The comparison between the PLP-derived spectrum and the FFT derived spectrum is not shown in this figure because the PLP frequency axis is warped to the Bark scale.

5.4. Delta Features

The performance of a speech recognition system is enhanced greatly by adding time derivatives to the basic static parameters. The first-order derivatives are referred to as delta features; the second-order derivatives are referred to as delta-delta features.

In digital signal processing, there are several ways to approximate the first order time derivative of signal.

$$s^*(n) = \frac{\partial}{\partial t}s(n) = s(n) - s(n-1) \qquad (15)$$

$$s^*(n) = \frac{\partial}{\partial t}(n) = s(n+1) - s(n) \qquad (16)$$

$$s^*(n) = \frac{\partial}{\partial t}s(n) = \sum_{w=-N}^{N} w s(n+w) \qquad (17)$$

Equations (15) and (16) are known as backward and forward differences, respectively. Equation (17) is often referred to as regression analysis. Similarly, the second-order time derivatives are approximated by reapplying these equations to the output of the first-order differentiator [19].

Since differentiation is inherently a noisy process, computing derivatives of smoothed parameters is desirable. The regression analysis as shown in equation (17) is a popular way to achieve this result. Since this equation computes differences symmetrically placed around the sample at time n, it uses a combination of N previous samples in each direction to compute the current value. Hence some measure of smoothing is inherent.

Regression analysis is used in this front-end to compute delta features. The first formulation is simply a weighted version of equation (17):

$$d_n = \frac{\sum_{w=1}^{dw} w(c_{n+w} - c_{n-w})}{2\sum_{w=1}^{dw} w^2}, \qquad (18)$$

where d_n is a delta coefficient at frame n, c_{n-w} and c_{n+w} are static parameters before and next to the current frame coefficient c_n, and dw is the delta window size. Since the regression formula depends on past and future speech parameter values, some modifications are required for the beginning and end of the speech data. The formulas shown in (19) account for these conditions.

$$d_n = \frac{\sum_{w=1}^{dw} w\left(c_{n+w} - c_0\right)}{2\sum_{w=1}^{dw} w^2}, n < dw,$$

$$d_n = \frac{\sum_{w=1}^{dw} w\left(c_{dw} - c_{n-w}\right)}{2\sum_{w=1}^{dw} w^2}, n > dw \qquad (19)$$

6. GRAPHICAL USER INTERFACE

While the front-end is capable of producing output models consistent with other state of the art systems, it can also be used to study the differences between the different algorithms. A Tcl-Tk based graphical user interface (GUI) is available to facilitate this user interaction. This utility inherits the signal display routine from the SWITCHBOARD Segmenter [4]. A snap short of the GUI is shown in Figure 6.

The user can vary different parameters for each algorithm and study its effect on the output feature vector. The option to run two or more algorithms at the same time is also available, enabling the user to compare the performance of different algorithms with respect to any parameter of interest. Of course audio capabilities are present, either the entire utterance or

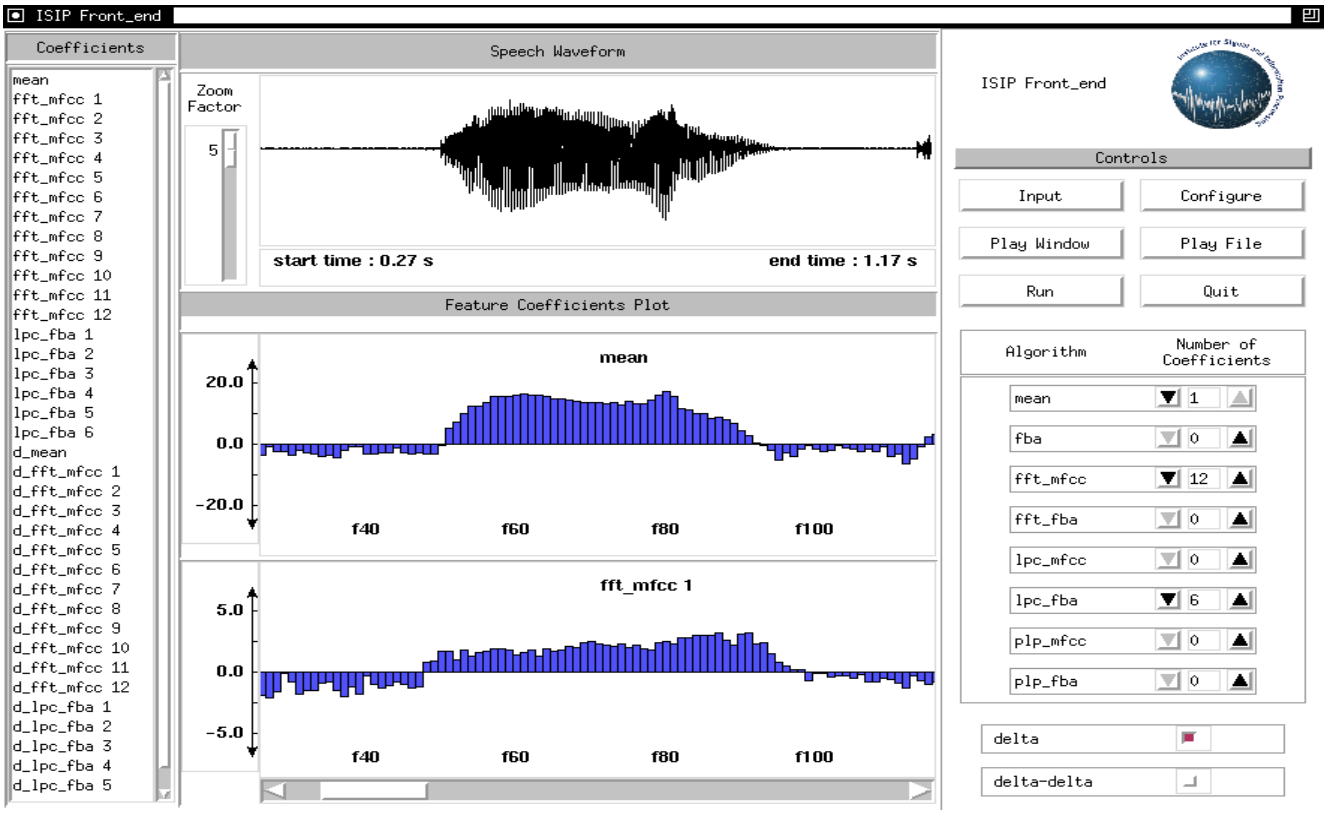

Figure 6. Screen capture of the graphical user

the currently displayed window can be played. All algorithmic parameters (window type, LP order, etc.) can be varied in the configuration window, shown in Figure 7.

7. EVALUATION

While the preferred method of evaluation would have been to study the effects of the front-end algorithms on the overall speech-to-text word error rate (WER), this is not a plausible course of action due to the current state of development of the ISIP recognition system. The ISIP recognizer does not currently support a training mode, which means it cannot use feature vectors generated by our front-end to train the acoustic models. The only acoustic models available to the ISIP recognizer are based on external software, so any WER experiments would suffer greatly from mismatched acoustic information. Instead, a state-of-the-art phone classification system is used to evaluate the effectiveness of each feature extraction algorithm. The data used is a subset of the OGI Alphadigits Corpus [1]. The Alphadigit Corpus was chosen because it has similar acoustic conditions to SWITCHBOARD [8], yet the Alphadigit task is significantly easier and forced alignments will be more accurate.

A frame duration of 10 ms. and a window duration of 25 ms. is used for data generation. The coefficient vectors generated include 12 mel scaled cepstral coefficients, mean energy, and 24 filter bank amplitudes. These features are generated for each algorithm, namely FFT, LPC and PLP. A LP order of 14 and PLP order of 5 is used.

The first classification technique employed is a Support Vector Machine (SVM) [7]. The core component in this paradigm is SVMLite, an SVM toolkit which is available as freeware. This SVM package can be applied to large datasets and is capable of handling classification tasks with tens of thousands of support vectors.

Since SVMs are a recent addition to the suite of tools commonly used by speech researchers, we leveraged

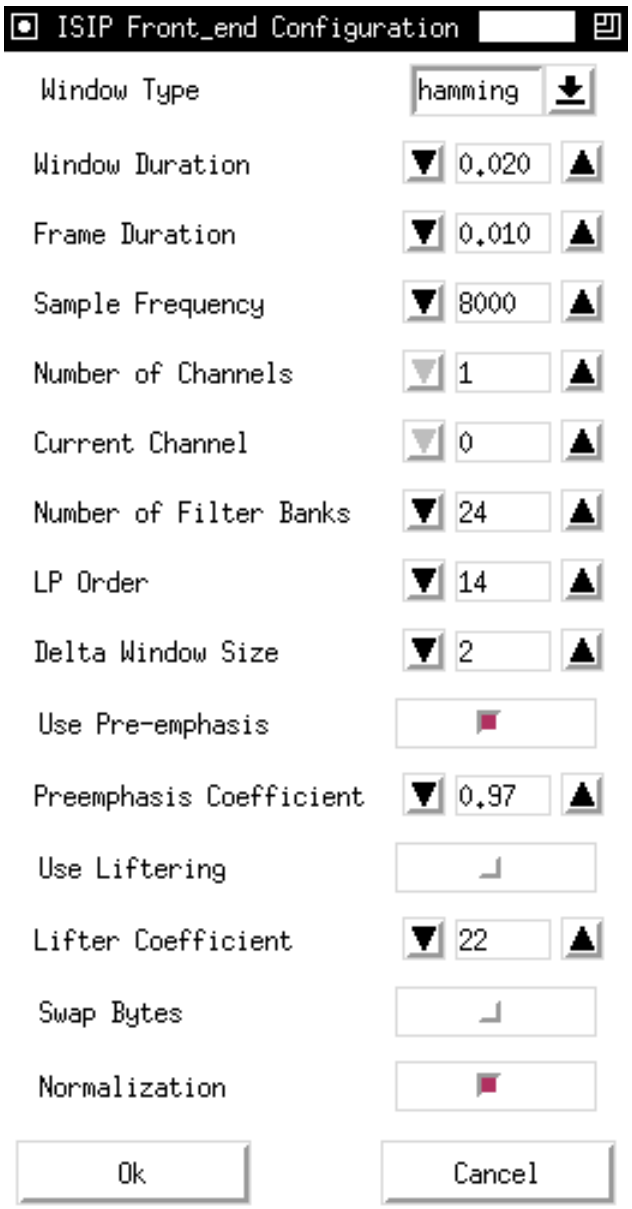

Figure 7.Configuration Window for the GUI

indigenous expertise in decision trees [11] to verify the results of one experiment. A Bayesian decision tree was trained and evaluated on the same data as the SVM for the FFT_MFCC features.

By comparing the output of this classifier and the reference information, which can be obtained by the state-level forced alignments of the input speech data, we evaluated the performance of each algorithm.

8. RESULTS AND CONCLUSIONS

The training for the classification techniques described above was performed on 200 frames per phone and the testing used 80 frames per phone. The results from the classification techniques are given in Table 1.

Algorithm	Classification Error	
	SVM	DT
FBA	96.3	
FFT_FBA	92.7	
FFT_MFCC	80.7	78.8
LPC_FBA	95.1	
LPC_MFCC	77.8	
PLP_FBA	91.9	
PLP_MFCC	91.5	

Table 1: Classification Errors

Nothing useful may be concluded from these numbers except for the fact that our evaluation was severely flawed.

The first thought upon seeing these numbers is that a better understanding of the SVM toolkit is needed to improve performance. To test this hypothesis, a decision tree experiment was run for one case. Unfortunately, the decision tree confirmed the poor results of the SVM experiment.

However the features do obey theoretical trends and are comparable with those of HTK, a state-of-the-art commercial recognizer. A comparison of the first cepstral coefficient is give in Figure 8. The plot indicates a nearly constant difference between the two due to pre-scaling of the data by the HTK front-end. A true evaluation of the front-end module involving full recognition experiments will be necessary to verify the validity of its output.

As neither classification technique seemed able to properly discriminate between the phones, yet the visual inspection of the coefficients suggest validity,

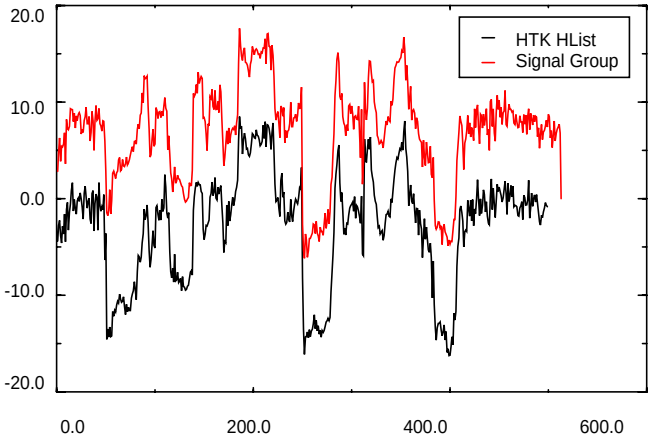

Figure 8.Comparison of the second FFT_MFCC

the fault must be either in the data preparation or the experimental paradigm itself. Statistical inspection of the training and testing data showed huge within class variance, generally an order of magnitude larger than the distance between the means of other classes.

The most likely factor is improper correlation between the forced alignment time markers and our features. Visual inspection of the audio file and forced alignments show a variable skew between 5 and 15 frames (0.15 seconds). In other words, the frame numbers shown in the alignments do not match the absolute time within the audio signal. The HTK recognizer prunes away data from the beginning and end of the utterance, so time-marks are not relative to the first sample in the file. Also, these alignments were obtained using error prone monophone models (as opposed to more state-of-the-art crossword triphones).

Hypothesizing that HTK generated features would match the HTK generated alignments, a decision tree classification experiment was run with 300 FFT-MFCC features for each phone. This experiment produced an open loop error of 95%. Visual and statistical inspection of the HTK generated data again showed extremely high variance and misalignment. Further inspection into the HTK recognizer is necessary to ascertain the absolute time of the alignment markers before meaningful classification results may be conducted.

9. SUMMARY

The processing of speech data into observation vectors which represent events in the probability space is performed by the front-end module. Frequency domain signal analysis techniques tend to be more insensitive to talker and channel variability than time domain approaches, thus extracting more useful information for speech-to-text systems. The standard algorithms employed are mean energy, digital filter banks, the Fourier transform, linear prediction, the cepstrum, and difference equations. Physiological knowledge of the human auditory and vocal articulatory systems is applied (the mel and Bark scales, perceptual linear prediction, frame duration, etc.) to the standard signal processing techniques to better model speech and increase recognition performance.

All software for this front-end module was developed in C++ using the public-domain GNU compiler. Our software is comprehensive, allowing the user complete control over all aspects of the signal modeling process. This includes algorithm selection, frame and window duration, and internal parameters. A Tcl-Tk based graphical user interface (GUI) is also available to facilitate user interaction with the numerous parameters. The GUI allows the user to vary different modeling parameters and study the effect on the output observations. It also assists in the comparison of different algorithms on the same data.

While the classification experiments are inconclusive, visual comparison of the first order coefficients to a reference system suggest the validity of our features. The most likely problems in the evaluation are improper phone alignment markers, stemming either from lack of understanding of HTK recognizer output or poor performance of the recognizer in forced alignment mode due to the use of monophone HMMs.

The front-end module described in this paper interfaces directly with the ISIP speech recognition system. A public domain implementation of all algorithms examined is available [6].

10. FUTURE WORK

The most obvious continuance of the work described in this paper is to run speech recognition experiments! A Viterbi training mode will be available to the ISIP recognizer by the end of the year. No longer encumbered by this limitation, real experiments may

be run to compare signal modeling parameters on recognizer accuracy.

Classification results would be helpful as an alternate means for empirically optimizing the front-end. The source of failure in the current experiments is most likely the forced alignment output, either an offset is present or the decoder does a poor job with monophone models. More powerful cross-word triphone models should be run for more accurate alignment information. If an offset still exists after running these models, its source (and inverse) must be uncovered.

The software itself leaves room for optimization. The current interface to the speech recognition system is text based, highly inefficient. Also, the final details of standardization to strict software engineering guidelines have yet to be applied.

Once the performance of this software is empirically verified and the code is standardized, this front-end module will take its place as an integral part of the public domain speech recognition system.

11. ACKNOWLEDGEMENTS

Aravind Ganapathiraju and his immense knowledge of Support Vector Machines has been instrumental in the evaluation. Dr. Joe Picone, Jon Hamaker, and Neeraj Deshmukh have provided considerable algorithmic help in developing this software. Audrey Le guided us through the use of her decision tree software on our phone classification problem.

12. REFERENCES

[1] “Alphadigit v1.0,” http://cslu.cse.ogi.edu/corpora/alphadigit/, Center for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology, USA, 1997.

[2] S.B. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, no. 4, pp. 357-366, 1980.

[3] J. Deller, J. Proakis and J. Hansen, *Discrete-Time Processing of Speech Signals*, MacMillan Publishing Company, New York, New York, USA, 1993.

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

[5] N. Deshmukh, A. Ganapathiraju, J. Hamaker and J. Picone, “An Efficient Public Domain LVCSR Decoder,” *Proceedings of the Hub-5 Conversational Speech Recognition (LVCSR) Workshop*, National Institute for Standards and Technology, Linthicum Heights, Maryland, September 1998.

[6] R. J. Duncan, V. Mantha, Y. Wu and J. Zhao, “Implementation and Analysis of Speech Recognition Front-Ends,” http://www.isip.msstate.edu/resources/ece_4773/projects/1998/group_signal, *Institute for Signal and Information Processing*, Mississippi State University, USA, November 1998.

[7] A. Ganapathiraju, J. Hamaker and J. Picone, "Support Vector Machines for Speech Recognition," *Proceedings of the International Conference on Spoken Language Processing*, Sydney, Australia, November 1998.

[8] J. Godfrey, E. Holliman and J. McDaniel, "SWITCHBOARD: Telephone Speech Corpus for Research and Development", *Proceedings of the IEEE ICASSP*, vol. 1, pp. 517-520, San Francisco, CA, USA, March 1992.

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

[10] C.R. Jankowski, H. Hoang-Doan and L.P. Lippmann, “A Comparison of Signal Processing Front Ends for Automatic Word Recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 286-292, July 1995.

[11] A. Le, "Bayesian Decision Tree for Classifica-

tion of Nonlinear Signal Processing Problems," Master of Science Special Project Presentation, http://www.isip.msstate.edu/resources/seminars/masters_oral/1998/decision_tree_bayes," *Institute for Signal and Information Processing*, Mississippi State University, November 1998.

[12] J. Makhoul, "Spectral Linear Prediction: Properties and Applications," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-23, pp. 283-296, 1975.

[13] D. O'Shaughnessy, "Speech Technology," *Applied Speech Technology*, Ed. A. Syrdal, R. Bennett, and S. Greenspan, CRC Press, Boca Raton, pp. 47-98, 1995.

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

[15] J. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1996.

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

[17] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1978.

[18] A.J. Robinson, "Dynamic Error Propagation Networks," Ph.D. Thesis, Cambridge University Engineering Department, Cambridge, England, February 1989.

[19] S. Young, *The HTK Book: for HTK Version 2.0*, Cambridge University Press, Cambridge, England, 1995.