FRONT END PARAMETERIZATION: LINEAR PREDICTION BASED MEASUREMENTS

Jim Trimble III

EE 8993 Course Project
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, Mississippi 39762
trimble@isip.msstate.edu

ABSTRACT

The front-end of a speech recognition system is the signal processing component of the whole system. Its function is to take a block of the speech signal at a time and from each, derive a smooth spectral estimate. This can be accomplished through various means (linear prediction, fourier analysis, filter banks); however, this paper focuses on linear prediction based measurements. Time derivatives and regression features of these measurements will also be investigated.

The output of the front-end is a set of feature vectors that represents the signal. Linear prediction provides an efficient and simple means of computing these feature vectors. This paper describes four different measurements: linear prediction, linear cepstral, mel cepstral, and perceptual linear prediction analysis.

1. INTRODUCTION

The front-end of a speech recognition system converts the speech signal into a sequence of feature or acoustic vectors. These vectors represent the frequency spectrum of the speech waveform over a time period of typically 10ms. Recognition of input sounds is achieved by computing a distance measure between input parameter vectors and corresponding reference vectors of the vocabulary words[2]. Therefore, using parameterization techniques that represent the speech signal as accurately as possible have a big impact on recognition rate. The techniques that may be employed to generate acoustic vectors include linear prediction, Fourier transforms, digital filter banks, and various transformations.

This paper focuses on linear prediction based measurements (linear prediction, linear cepstral, mel cepstral, and perceptual linear prediction analysis). Time derivatives and regression features of the above techniques will also be investigated. Both are used to represent the dynamic changes within the signal's frequency spectrum.

The implementation of the front-end resulted in a system that generates an arbitrary number of linear prediction and linear cepstral coefficients and twenty mel cepstral coefficients with its first and second time derivatives. The first and second order regression features of the mel cepstral coefficients computed over three frames were also generated. The different measurements will be related to the system implementation as they are discussed.

2. LINEAR PREDICTION

Linear prediction provides an efficient and simple means of calculating the static coefficients of the feature vector. Its computation is based on an all-pole model of speech. This model can be derived from a concatenation of lossless acoustic tubes to represent the human vocal tract. Each tube has a different cross-sectional area pertaining to the varying cross-sectional area of the vocal tract. However, in the study of modeling speech production, one can find compelling arguments for the inclusion of zeros in the speech production model [8]. It is well known that the spectrum of phonemes, especially vowels, have format frequencies that are approximately modeled by all-pole structures. On the other hand, vocal tract characteristics such as the glottal pulse waveform and lip radiation and articulations involving nasals and fricatives introduce zeros into the system function. Thus a zero-pole system would seem to be the better choice.

The all-pole model of speech production is a minimum phase representation of the speech signal. Because all information within speech is contained solely in the signal's magnitude, the absence of phase does not hinder the perception of the human ear nor does it hinder the recognizer. A linear prediction representation of the speech model is not only simple but works well.

Linear prediction tries to predict a signal, *s(n)*, by a linear combination of its past values. This linear combination includes weights, *a(i)*'s, which form the predictor equation coefficients. Thus, the term *linear perediciton* is used.

Consider the predicted values of the signal defined by

$$\hat{s}(n) = s(n) - as(n-1) \tag{1}$$

The error between the signal and its predicted value based on its previous value is given by:

$$e(n) = s(n) - \hat{s}(n) = s(n) - as(n-1) \tag{2}$$

The total squared error is defined by:

$$E = \sum_n e^2(n) = \sum_n \{s(n) - as(n-1)\}^2 \tag{3}$$

$$= \sum_n (s^2(n) - 2as(n)s(n-1) + a^2 s^2(n-1))$$

The error is minimized with respect to a by differentiating E and setting the result equal to zero

$$\frac{\partial E}{\partial a} = 0 = \sum_n -2s(n)s(n-1) + 2as^2(n-1) \tag{4}$$

$$= \sum_n s(n)s(n-1) = a\sum_n s^2(n-1)$$

A more general form which includes multiple pass values:

$$\frac{\partial E}{\partial a} = \sum_n s(n)s(n-l) = a_k \sum_n s(n-k)s(n-l) \tag{5}$$

or

$$c(l, 0) = \sum_{k=1}^{p} a_k c(k, l) \tag{6}$$

This equation gives the covariance method of calculating the a's or predictor coefficients.

$$\bar{c} = \underline{C}\bar{a} \tag{7}$$

Inversion of the covariance matrix is needed to solve for the predictor coefficients:

$$\bar{a} = \frac{\underline{C}^{-1}}{\bar{c}} \tag{8}$$

By limiting the data to within each frame, the autocorrelation method can be used to compute the predictor coefficients:

$$\bar{a} = \underline{R}^{-1}\bar{r} \tag{9}$$

This method, of course, requires the inversion of the autocorrelation matrix. A simple and efficient algorithm that circumvents the need for matrix inversion is the Levinson-Durbin algorithm.

2.1. Levinson-Durbin Recursion

$E_0 = r(0)$

for i = 1, 2,.,p

$$k_i = \left(r(i) - \sum_{i=1}^{i-1} a_{i-1}(j) r(i-j) \right) / E_{i-1}$$

$a_i(i) = k_i$

for j = 1, 2,...,i-1

$a_i(j) = a_{i-1}(j) - k_i a_{i-1}(i-j)$

$E_i = (1 - k^2)E_{i-1}$

The autocorrelation values are used to compute intermediate values (k_i's) called reflector coefficients. These values in turn are used to compute the predictor coefficients. The error denoted by E decreases within the iterative process. The pth value of E which is the difference between the signal and the predicted signal is used for gain matching. The gain

$$gain = 10 \cdot \log|E| \tag{10}$$

is multiplied by the model spectrum to match the magnitude of the signal spectrum.

Let us make a few comments about the reflector coefficients. The reflector coefficients in the algorithm above have an absolute value of less than one. Absolute values of one implies a harmonic process (poles on the unit circle), and absolute values greater than one implies an instability (poles outside the unit circle). Now, we have insight on how to determine the LP order during the calculations. As the reflector coefficients approach one, we approach our LP order. The reflector coefficients are orthogonal in that the pth order model contains the p coefficients for the p+1 order model.

The linear prediction process can be viewed as a filter by taking the z-transform of the error:

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{11}$$

$$E(z) = S(z)A(z) \tag{12}$$

where

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \quad (13)$$

and $1/A(z)$ is our all-pole model.

3. CEPSTRAL ANALYSIS

Cepstral analysis is designed for problems centered around "voiced" speech. Speech is composed of a glottal excitation sequence convolved with the impulse response of the vocal tract. Glottal excitation is a unique characteristic of each individual. Eliminating this characteristic from the speech model facilitates better speaker-independent recognition. Because the individual parts are not linearly combined, the customary linear techniques (e.g. Fourier analysis) provide no help. Cepstral analysis, on the other hand, provides a means for separating convolved signals. It's cepstrum (like the spectrum) represents a transformation on the speech signal with two properties[7,4]:

1. The representation of the component signals will be separated in the cepstrum.

2. The representation of the component signals will be linearly combined in the cepstrum.

In talking about the cepstrum, one can refer to what is called the "real cepstrum" (RC) and the "complex cepstrum" (CC). The RC is equivalent to the even part of the CC; it discards the phase information of the signal.

One of the most important applications of cepstral analysis in contemporary speech processing is the representation of an LP model by cepstral parameters [8]. On that note, we shall only concern ourselves with the "real cepstrum."

The computation of the RC is shown in block diagram in Figure 1.

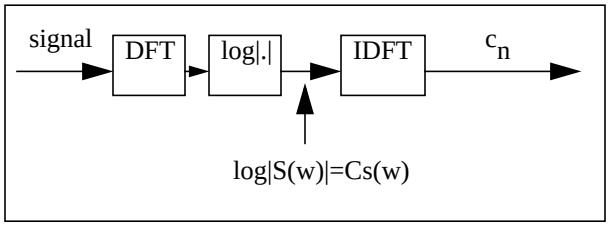

Figure 1. Computation of the RC.

The first two operations, the DFT and the log function, transforms the signal so that its convolved parts are resolved into additive components in the new domain (quefrency domain).

$$C_s(\omega) = \log|S(\omega)| \quad (14)$$

$$= \log|E(\omega)\Theta(\omega)|$$

$$= \log|E(\omega)| + \log|\Theta(\omega)|$$

$C_s(\omega)$ - cepstral magnitude spectrum

$S(\omega)$ - signal magnitude spectrum

$E(\omega)$ - glottal excitation magnitude spectrum

$\Theta(\omega)$ - vocal tract magnitude spectrum

As with the spectrum, the slow varying signal, $\Theta(\omega)$, and the fast varying signal, $E(\omega)$, manifest themselves at the low and high end of the quefrency domain respectively. The glottal excitation can be removed from the signal by performing a "low-time liftering" (a play on filtering) process. This process devoids the vocal tract component, $\log \Theta(\omega)$, of any excitation component. This, in essence, results in a cepstral smoothing of the vocal tract spectrum.

LP analysis does not resolve the vocal-tract characteristics from the glottal dynamics. The glottal excitation is a laryngeal characteristic that varies from person to person. Hence, LP parameters convey some information that can degrade performance, especially for speaker-independent systems.

To derive cepstral parameters from LP analysis, the transfer function, *A(z)*, from Equation 13 is utilized by expanding the logarithmic transfer function, *lnA(z)*, into a power series of z^{-1}. If all the poles of *A(z)* are inside the unit circle, *lnA(z)* can be expressed as [3]

$$\ln A(z) = C(z) = \sum_{n=1}^{\infty} c_n z^{-n} \quad (15)$$

Where $z = exp(jwt)$, w = frequency in radians, T = sampling interval, and c_n is the amplitude at the nth sampling instant, $t = nT$, of the inverse Fourier transform of *C(z)*. *C(z)* is considered as a function of the frequency variable *w*.

To obtain the relationship between cepstral and predictor coefficients, Equation 15 is substituted into Equation 15. The derivative of both sides is taken with respect to z^{-1}:

$$\frac{d}{dz^{-1}} = \ln\left[1 \Big/ \left\{1 - \sum_{k=1}^{p} a_k z^{-k}\right\}\right] = \frac{d}{dz^{-1}} \sum_{n=1}^{\infty} c_n z^{-n} \quad (16)$$

which simplifies to

(17)

$$\left\{\sum_{k=1}^{p} k a_k z^{-k+1}\right\} = \left\{1 - \sum_{k=1}^{p} a_k z^{-k}\right\} = \sum_{n=1}^{\infty} n c_n z^{-n+1}$$

and rewritten as

$$\sum_{k=1}^{p} k a_k z^{-k+1} = \left(1 - \sum a_k z^{-k}\right) \cdot \sum_{n=1}^{\infty} n c_n z^{-n-1} \qquad (18)$$

By equating the constant terms and the various powers of z^{-1} on both sides of Equation 18, we obtain the relationship between the cepstral and predictor coefficients [4]:

$$c_1 = a_1 \qquad (19)$$

$$c_n = \sum_{k=1}^{n-1} (1 - k/n) a_k c_{n-k} + a_n, \; 1 < n < p \qquad (20)$$

Equations 19 and 20were used to compute the LP based linear cepstral coefficients.

4. MEL CEPSTRAL ANALYSIS

In the 1980's, the cepstrum began to supplant the direct use of the LP parameters as the premiere feature in the important "hidden Markov modeling" strategy because of two enhancements that were found to improve speech recognition [8,9]. The first has already been mentioned--cepstral smoothing of the LP-based spectrum. The second enhancement is due to the "mel cepstrum."

A "mel" is a unit of measure of *perceived pitch* or *frequency* of a tone. Work done by Stevens and Volkman showed that the frequency resolution of the human ear is approximately linear below 1 kHz and logarithmic above 1 kHz. A mapping of pitch (mel) versus actual frequency gives us the mel scale. Researchers began investigating the benefits of using a warped frequency axis to correspond to the mel scale[17].

In using the DFT to compute the real cepstrum, the signal may be over sampled to obtain frequencies as close as possible to the desired frequencies. The LP-based mel cepsral parameters were computed by mel-frequency warping the log magnitude spectrum of the LP parameters. Samples were taken linearly below 1 kHz and logarithmically above 1 kHz. The values are shown in Table 1.

Table 1: Frequency Components; 16kHz data, 2048-point DFT

"Desired" Frequency (Hz)	"Quantized" DFT frequency (Hz)	"k" value DFT
100	101	13
200	203	26
300	304	39
400	406	52
500	507	65
600	609	78
700	710	91
800	812	104
900	914	117
1000	1015	130
1148	1148	147
1318	1320	169
1514	1515	194
1737	1734	222
1995	1992	255
2291	2289	293
2630	2632	337
3020	3023	387
3467	3468	444
4000	4000	512

It has been found that the perception of particular frequencies are influenced by energy within a critical band found around the frequencies. Some researchers suggest using the *total log energy* found within the critical bands around each frequency rather than just using the log magnitude. This results in applying a sequence of critical band filters to the log magnitude spectrum of the signal. The width of the filters are constant below 1 kHz and increases logarithmically above 1 kHz. These critical-band filters give us a sequence of weighted total energy:

$$Y(i) = \left(\sum_{k=0}^{N/2} \log|S(k;m)| \right) H_i(k(\{2\pi\}/N)) \quad (21)$$

where the summation is over a small range of k's around k_i (normalized center frequency).

$H_i(\omega)$ - ith critical-band filter.

$|S(k;m)|$ - weighted log terms within the ith critical-band filter.

The sequence is then inverse transformed to give the mel cepstral coefficients.

5. PERCEPTUAL LINEAR PREDICTION

One of the main disadvantages of the LP all-pole model is that it approximates the speech signal equally well over all frequencies within the analysis band [5]. LP analysis does not preserve and discard spectral details according to their auditory prominence.

Hermansky studied a class of spectral transform LP techniques that modify the power spectrum of speech prior to its approximation by the autoregressive LP model [5]. The steps involved are as follows:

A. Spectral analysis - DFT the signal and compute the power spectrum.

$$P(\omega) = Re[S(\omega)]^2 + Im[S(\omega)]^2 \quad (22)$$

B. Critical-band analysis - warp the power spectrum along its frequency axis into the Bark frequency.

(23)

$$\Omega(\omega) = 6\ln\{\omega/((1200\pi) + [(\omega/(1200\pi))^2 + 1])\}$$

Then convolve the warped power spectrum with the power spectrum of the simulated critical band masking curve (similar to spectral processing in mel cepstral analysis).

$$\psi(\Omega) = \quad (24)$$

0, for $\Omega < -1.3$

$10^{2.5(\Omega + 0.5)}$, for $-1.3 \le \Omega \le -0.5$

1, for $-0.5 < \Omega < 0.5$

$10^{-1.0(\Omega - 0.5)}$, for $0.5 \le \Omega \le 2.5$

0, for $\Omega > 2.5$

This is a rather crude approximation of what is known about the shape of the auditory filters. The above discrete convolution yields samples of the critical-band power spectrum:

$$\Theta(\Omega_i) = \sum_{\Omega=-1.3}^{2.5} P(\Omega - \Omega_i)\psi(\Omega) \quad (25)$$

This reduces the spectral resolution of $\Theta(\omega)$ in comparison with the original $P(\omega)$ and allows for the down-sampling of $\Theta(\omega)$. The exact value of the sampling intervals is chosen so that an integral number of spectral samples covers the whole analysis band. Typically, 18 spectral samples of $\Theta[\Omega(\omega)]$ are used to cover the 0 - 16.9-Bark (0 - 5 kHz) analysis bandwidth in 0.994-Bark steps.

C. The sampled $\Theta[\Omega(\omega)]$ is preemphasized by the simulated equal-loudness curve:

$$\Xi[\Omega(\omega)] = E(\omega)\Theta(\Omega(\omega)) \quad (26)$$

where

$E(\omega)$ is an approximation to the nonequal sensitivity of the human hearing at different frequencies.

(27)

$$E(\omega) = [(\omega^2 + 56.8 \times 10^6)\omega^4]/(\omega^2 + 6.3 \times 10^6)^2 \quad (\omega^2 + 0.38 \times 10^9)$$

D. The cubic-root amplitude compression

$$\Phi(\Omega) = \Xi(\Omega)^{0.33} \quad (28)$$

simulates the nonlinear relationship between the intensity of sound and its perceived loudness.

E. In the final operation of PLP analysis, $\Phi(\Omega)$ is approximated by the spectrum of an all-pole model using the autocorrelation method of all-pole spectral modeling.

6. TIME DERIVATIVES AND REGRESSION

The first time derivatives (the dynamic feature) and the second time derivatives (the acceleration feature) are used to represent the dynamic changes in the speech spectrum. Regression is an alternative representation of time derivatives. The first, second, and higher order regression features represent the first, second, and higher numerical derivatives of the feature vector. Normally, these dynamic and acceleration features are computed for cepstral coefficients.

Abrupt spectral changes found within the speech spectrum such as those found in consonant phonemes can be detected with time derivatives. In doing so, recognition rates are improved. Another use for time derivatives is to make recognition systems more robust to noise. The speech signal can be corrupted by ambient noise and distortions due to the "Lombard effect." The Lombard effect is the distortion of speech due to speaking louder[10]. One possible approach for improving recognition rates for noisy speech is to train the recognizer under conditions to match the test conditions or predict all conditions under which the recognizer will be used. However, one can not in general predict all the test conditions under which the recognizer will be used. A better approach is to find a more robust representation of the speech which will support high recognition rates in normal and noisy speech.

The time derivatives of the cepstral coefficients at time t are defined as follows[6]:

Static feature

$$S_k(t) = c_k(t) \tag{29}$$

Dynamic feature

$$D_k(t) = c_k(t+\delta_D) - c_k(t-\delta_D) \tag{30}$$

Acceleration feature

$$A_k(t) = c_k(t+\Delta_A+\delta_A) - c_k(t+\Delta_A-\delta_A) - c_k(t-\Delta_A+\delta_A) + c_k(t-\Delta_A-\delta_A) \tag{31}$$

where

c_k - denotes the *k*-th cepstral coefficient from a frame of speech at time "t".

δ_D - half of the delay between the frames differenced.

δ_A, Δ_A - play a similar role as δ_D; interchanging the values leaves the feature unchanged.

The *r*-th order regression can be written as[7]:

$$R_{rk}(t,T,\Delta T) = \frac{\sum_{X=1}^{L} P^2{}_r(X,L)\,C_k\left(t+\left(X-\frac{L+1}{2}\right)\Delta T\right)}{\sum_{X=1}^{L} P^2{}_r(\dot{X},L)} \tag{32}$$

where

C_k - denotes the *k*-th cepstral coefficient from a frame of speech at time t.

T - the time length over which the regression feature is calculated.

ΔT - the step size between speech analysis frames.

$L = T/\Delta T$ - (the number of analysis frames in time length *T*) is odd, so that the central frame at

X = (L+1)/2 is included in the sums.

The weighting function, $P_r(X,L)$, is the *r*-th orthogonal polynomial of length *L*. The first few orthogonal polynomials are:

$$P_0(X,L) = 1 \tag{33}$$

$$P_1(X,L) = X \tag{34}$$

$$P_2(X,L) = X^2 - \frac{1}{12}(L^2-1) \tag{35}$$

$$P_3(X,L) = X^3 - \frac{1}{20}(3L^2-7)X \tag{36}$$

7. CURRENT SYSTEMS

Current speech recognition systems typically incorporate the following elements into their systems: LP derived feature vectors for computational efficiency, static features combined with short-term time differences to capture the dynamic aspects of the spectrum, and energy measures (absolute energy, normalized energy, and/or differenced energy)[1]. Most common systems like the HTK LVSCR of Cambridge

University incorporates 12 mel cepstral coefficients along with the first and second differentials. The HTK LVSCR system includes normalized log energy. It uses a sample frequency of 16 kHz and a frame duration of 10ms[11].

Cambridge University's hybrid connectionist-hidden Markov model large-vocabulary speech recognition system, ABBOT, has used two sets of acoustic features in the past. These include a 20 channel mel-scaled filter bank with three voicing features and 12th order cepstral coefficients derived using perceptual linear prediction. The system uses a 32ms window and a 16ms frame duration[12].

The BBN/BYBLOS speech recognition system incorporates 45 features including 14 cepstral coefficients, energy, and the first and second derivatives of both[13].

Like the HTK system, IBM's large vocabulary continuous speech recognition system for the ARPA NAB news Hub-1 test, uses 16 kHz data and a frame duration of 10ms. The system computes a 24-band mel cepstra[14].

Dragon's continuous speech recognizer produces a set of 44 parameters for each 10ms frame: 1 total log energy, 7 log spectral energies, 12 Mel-cepstrals, 12 cepstral differences, and 12 cepstral second differences[15].

The AT&T Speech-To-Text system produces a 39-dimensional observation vector every 10ms. Twelve mel-cepstral parameters along with their first and second time derivatives are produced. The 0th cepstral parameter is used as an estimate of energy[16].

8. SUMMARY

The implementation of the front-end generates LP-based measurements. Window duration, frame duration, and LP-order are user defined. The Levinson-Durbin algorithm is used to compute the LP coefficients. From these coefficients, the linear cepstral coefficients of LP-order are derived. The system uses a fixed number of critical band filters to generate twenty mel-cepstral coefficients for 16 kHz data. First and second time derivatives and first and second order regression features computed over a fixed length of three frames are generated for the mel-cepstral coefficients. The twenty mel-cepstral coefficients, time derivatives, and absolute energy make up a 61 component feature vector.

The twenty mel-cepstral coefficients were computed according to the critical band center frequencies found in Table 1. In order to make the coding simpler, regression was calculated over a fixed length rather than an arbitrary length. Also, perceptual linear prediction was not implemented.

Unfortunately, the different feature vectors were not compared. Comparison, if time permitted, would have been done by producing the parameters for a corpus of several speakers--each speaking the same set of two different phonemes. A scatter plot showing the relative tightness of clusters for the same phonemes and separation for the two different phonemes would have been compared for each parameter.

REFERENCES

1 J. Picone, "Continuous Speech Recognition Using Hidden Markov Models," in *IEEE ASSP Magazine*, pp. 26-41, July 1990.

2 E.L. Bocchieri and R Doddington, "Frame-Specific Features for Speaker Independent Speech Recognition," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 4, pp.755-764, August 1986.

3 B.S. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," in *J. Acoust. Soc. Am.*, Vol. 55, No. 6, pp. 1304-1312, June 1974.

4 S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification," in *IEEE Transactions on Acoustic, Speech, and Signal Processing*, Vol. ASSP-29, No. 2, April 1981.

5 H. Hermansky, "Perceptual Linear Prediction (PLP) Analysis of Speech," in *J. Acoust. Soc. Am.*, Vol 87, No 4, pp. 1738-1752. April 1990.

6 B.A. Hanson and T.H. Applebaum, "Robust Speaker-Independent Word Recognition Using Static, Dynamic and Acceleration Features: Experiments with Lombard and Noisy Speech," in *ICASSP*, pp. 857-860, 1990.

7 B.A. Hanson and T.H. Applebaum, "Regression Features for Recognition of Speech in Quiet and in Noise," in *ICASSP*, pp. 985-988, 1991.

8 J.R. Deller, J.G. Proakis, J.H.L. Hansen, Discrete-Time Processing of Speech Signals, Macmillan Publishing Company, New York, New York, 1993.

9 J.P. Haton, Automatic Speech Analysis and Recognition, D. Reidel Publishing Company, Dordrecht, Holland, 1982.

10 S. Furui, "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum," in IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 1, pp. 52-

59, February 1986.

11 P.C. Woodland, M.J.F. Gales, P.V. Valtchev, “The HTK Large Vocabulary Recognition System for the 1995 ARPA H3 Task,” Cambridge University Engineering Department, Cambridge, England.

12 D.J. Kershaw, A.J. Robinson, S.J. Renals, “The 1995 Abbot Hybrid Connectionist-HMM Large-Vocabulary Recognition System,” Cambridge University Engineering Department, Cambridge, England, Department of Computer Science, University of Sheffield, Sheffield England.

13 L. Nguyen, et. al. “The 1994 BBN/BYBLOS Speech Recognition System,” BBN Systems and Technologies, Cambridge, Massachusetts, Northeastern University.

14 L.R. Bahl, et. al. “The IBM Vocabulary Continuous Speech Recognition System for the ARPA NAB News Task,” IBM T. J. Watson Research Center, Yorktown Heights, New York.

15 S. Wegmann, et. al. “Marketplace Recognition Using Dragon’s Continuous Speech Recognition System,” Dragon System Incorporated, Newton, Massachusetts.

16 D.M. Hindle, A. Ljolje, M.D. Riley, “Recent Improvements to the AT&T Speech-To-Text (STT) System,” AT&T Research, Murray Hill, New Jersey.

17 S.B. Davis and P. Mermelstein, “Comparison of Parametric Representaion in Continuously Spoken Sentences,” in IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-28, No. 4, pp. 357-366, August 1980.