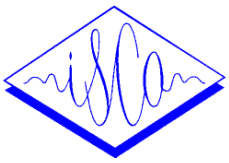

2nd International Conference on
Spoken Language Processing (ICSLP 92)
Banff, Alberta, Canada
October 12-16, 1992

A Novel Speech Recognizer for Keyword Spotting*

Gregory J. Clary and John H. L. Hansen
Department of Electrical Engineering
Duke University
Durham, North Carolina 27706

ABSTRACT

This paper presents a newly formulated speech recognition algorithm for keyword spotting which uses a feature enhancing artificial neural network, a semi-continuous hidden Markov model, and a likelihood ratio test based on optimal detection theory to make decisions regarding possible keywords. The speech recognizer can be used to detect the occurrences of a single word within connected input speech streams in noisefree neutral or Lombard stressed environments. A keyword-dependent neural network [1] enhances speech parameters and reduces the probability of false acceptances of non-keywords by adapting its weights and input layer width based on extracted speech characteristics [2]. Using the neural network reduces false acceptances by more than $\frac{1}{3}$ for mono-syllable keywords in a defined keyword spotting application [3]. Enhanced features are submitted to a semi-continuous hidden Markov model which produces a score indicating the presence of the represented keyword. A likelihood ratio test uses functions formed from keyword and non-keyword recognizer training data for detection. Receiver operating characteristics (ROC's) show that the new recognition algorithm can improve keyword spotting performance for neutral and Lombard effect speaking conditions.

10.21437/ICSLP.1992-6

1 Introduction

For the purposes of this paper, keyword spotting is considered to be the detection and classification of words from a small (target) vocabulary within a connected speech stream. It is assumed that connected speech streams can be parsed into individual words so that each input to the proposed speech recognizer is a single possible keyword. An "endpoint detector" which can determine the beginning and ending points of individual words has been demonstrated in a speech enhancement application for mild levels of background noise [2]. This method has also been extended to the problem of connected-word parsing in streams of multiple token speech. For keyword recognizer evaluations, it is assumed that the endpoint detector has been previously applied to connected speech streams under test to generate sets of parsed words. Previous studies have demonstrated successful methodologies for keyword spotting in neutral connected speech streams [4]. In this study, keyword spotting of speech under stress is considered. Isolated word recognition rates have been shown to decrease by as much as 13-68% when speech is produced in stressful conditions (task induced or emotion stressed)[14, 10]. Studies suggest that keyword spotting performance, employing traditional neutral trained recognition frameworks, degrade when stress is introduced [5]. This decrease in performance is due to changes which occur during speech production when a talker is under stress. The stress condition considered here is Lombard effect[9].

The speech recognizer proposed here uniquely combines three processing steps to detect or reject possible keywords. First, features are enhanced by a keyword-dependent neural network [1]. The feature enhancing artificial neural network (FEANN) is applied to short-time segments of extracted speech parameters as in [6]. The neural network has keyword-dependent weights and reduces the probability of false acceptances of non-keywords. It is unique in that the weights and width of the input layer adapt based on extracted speech characteristics [2] from the input speech signal. The information-preserving weights are determined using a principal component analysis process and can be found by applying iterative or conventional algorithms [7]. Second, because the neural network produces continuously-valued output vectors, enhanced features are submitted to a semi-continuous hidden Markov model (SCHMM) [8]. Initial means and covariance matrices of the mixture densities for the model are determined using a process based on vector quantization. The SCHMM produces a score for a possible keyword to indicate whether or not it is an instance of the represented keyword. Finally, a likelihood ratio test based on optimal detection theory is applied to the SCHMM score. The detection scheme uses likelihood functions formed from keyword and non-keyword recognizer training data.

Issues relating to algorithm formulation and training of the FEANN and SCHMM based recognizer are described in this paper. The applicability of detection theory to the keyword spotting problem is discussed and demonstrated. Performance evaluation is shown for both neutral and speech spoken under Lombard effect. In a noisy environment, a speaker is able to hear background noise, causing him to alter his speech characteristics in an effort to increase communication efficiency over the noisy medium (known as the Lombard effect [9]). Lombard effect was simulated by having speakers produce speech while listening to 85 dB SPL pink noise through headphones (i.e., noisefree tokens are used here).

This paper is organized as follows. Sec. 2 describes the three major processing steps for the proposed keyword recognizer–FEANN application, SCHMM score calculation, and likelihood ratio test. Sec. 3 presents evaluations of the keyword recognizer under neutral and Lombard effect conditions. Conclusions are drawn in Sec. 4.

2 Formulation of a Keyword Recognizer

The following steps are employed by the keyword recognizer. Parameters extracted from parsed word data are applied to a feature enhancing neural network. Enhanced features are then submitted to a word-level semi-continuous hidden Markov model to produce a score for each word. A likelihood ratio test is applied to the score to form a decision of whether the input is a keyword. A block-diagram representation of the recognizer is shown in Fig. 1.

The challenge of applying word-level hidden Markov models to keyword spotting is to formulate a recognizer which will reject non-members of the target vocabulary. It is especially difficult to reject words which are confusable with words in the target vocabulary. Here, a feature enhancing neural network unique to each keyword model is applied to each model input to effectively partition the pattern space by a SCHMM. The following subsections present details of each of the score-producing steps and the rationale for the likelihood ratio test.

2.1 Parameterization

Nine mel-cepstral coefficients $C^n, n = 1 \ldots 9$ are extracted from each analysis window using energy from 20 critical mel frequency bands. Analysis windows contain 128 samples of speech sampled at 8 kHz. A previously developed adaptive endpoint detector is applied prior to mel-cepstral coefficient calculation to generate parsed word data and to perform speech classification into voiced/transitional/unvoiced speech sections[10].

2.2 Feature Enhancing Artificial Neural Network

The feature enhancing neural network employs short-time transformations in a manner similar to those in [6] in order to closely track changes in the mel-cepstral coefficients. Further details of the FEANN algorithm and its training and operation can be found in [1, 3]. A brief review is presented here.

*This work sponsored in part by National Science Foundation, NSF-IRI-90-10536.

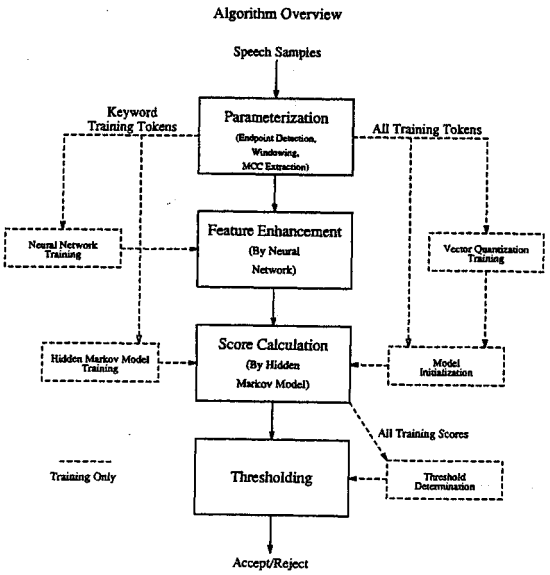

Figure 1: A block-diagram representation of the recognizer which judges each possible keyword.

The criteria employed for the design of linear, feature enhancing neural networks are that they should have class-dependent weights, should preserve information, and should take advantage of application-specific knowledge of the input signal. These criteria are met by imposing the following constraints. To provide class-dependent weights, the weights are determined from only training data tokens of the modeled class. In addition, the Karhunen-Loeve transformation is used to insure that the neural network is information-preserving based on a minimum mean square error between the actual network input and the input reconstructed from the network output and weights. Finally, the width of the input layer of the neural network adapts as characteristics of the speech signal vary and new segment types are encountered. Segment types are classified in the parameterization step as voiced/transitional/unvoiced.

A time sequence of vectors, each consisting of nine mel-cepstral coefficients, provides the input to the neural network and is linearly transformed by sets of weights. Each NT-MCC[1] time series is transformed by a subnetwork, which "slides" across the input frames. The size of a subnetwork's input layer depends on segment type. At a particular instant in time, all subnetworks have the same input layer width, but different weights. As long as the segment type remains the same, the input layer width remains the same. The subnetworks are advanced by a number of frames less than the current input layer width (normally by one frame). The input layer widths are chosen based on how fast the mel-cepstral coefficients change in a given segment type and how often the type occurs. Parameters change most slowly in a voiced section; thus, the largest input layer width is chosen for voiced segments.

Fig. 2 shows how the input layer width of the neural network changes as new segments are encountered for a single mel-cepstral coefficient from a partial instance of the word "six." Note that all 9 mel-cepstral coefficients undergo a transformation, but only instances of mel-cepstral coefficient n, C_i^n, are pictured here, where i is the frame number. The resulting transform coefficient at time t is denoted by Y_t^n. Time t can be equal to frame number i but differs from i when the subnetworks are advanced by more than one frame. The network output is (assuming a mapping from i onto t):

$$Y_t^n = \sum_{k=0}^{M_j-1} W_{jk}^n * C_{i+k}^n \qquad (1)$$

[1]The notation NT-MCC refers to non-transformed vectors of mel-cepstral coefficients. We will introduce the notation T-MCC to represent mel-cepstral coefficients which have been transformed by a FEANN.

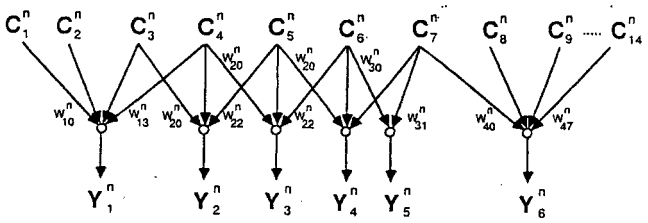

Figure 2: A snapshot of the C^n subnetwork applied to an instance of "six." Four out of five sets j of unique subnetwork weights are pictured, $W_{jk}^n, k = 0, \ldots, M_j - 1$. Only selected weights are labeled.

where the segment at time t is of type j and M_j is the corresponding input layer width.

To determine the sets of weights for the neural network, sample correlation matrices are formed from training data for each coefficient n and segment type j, and the principal eigenvector is found for each matrix. Sample correlation matrices are formed as follows:

$$\bar{Q}_j^n = \frac{1}{N_j} \sum_{i \epsilon j} (\{C_i^n, \cdots, C_{i+M_j-1}^n\}^T \{C_i^n, \cdots, C_{i+M_j-1}^n\}) \qquad (2)$$

where N_j is the number of training samples for the jth segment type, and T is the transpose operator. For the subnetwork pictured in figure 2, the following error quantity is therefore minimized:

$$E_j^n = \sum_{i,t \epsilon j} E_{jit}^n = \sum_{i,t \epsilon j} ((W_{jG(i,t)}^n * Y_t^n) - C_i^n)^2 \qquad (3)$$

where $G(i,t)$ maps frame i and time t onto the weight index corresponding to i and t. Although this work was motivated in part by iterative algorithms which implement the Karhunen-Loeve transform, the Jacobi method is used here to find the principal eigenvector.

2.3 The Semi-Continuous Hidden Markov Model

Five state semi-continuous hidden Markov models (SCHMM) are used to statistically model each keyword based on training data observation vectors [8]. SCHMMs have been shown to be superior to discrete and continuous hidden Markov models for a speaker-dependent phoneme recognition task [11]. The SCHMM employed here is a continuous mixture HMM, except that the multivariate densities are the same for each state. The mixture weighting coefficients C are unique for each state and each mixture density, just as in the continuous mixture HMM. In this work, there are $L = 64$ multivariate Gaussian densities which are based on an a priori trained codebook. The probability density function for a single state i, is:

$$b_i(\mathbf{x}) = \sum_{j=1}^{64} f_j(\mathbf{x}) C_{ij} \qquad (4)$$

where $f_j(\mathbf{x}) = N(\mathbf{x}, \mu_j, \Sigma_j)$ is a multivariate Gaussian density with mean vector μ_j and covariance matrix Σ_j. A SCHMM initially in state 1 can be represented by $\lambda = (\mathbf{A}, C, \mu, \Sigma)$, where $\mathbf{A}$ is the state transition matrix. Model parameter re-estimation for training the SCHMM is accomplished via the Baum-Welch forward-backward algorithm for maximum likelihood estimation. Diagonal entries for the state transition matrix $\mathbf{A}$ are initialized to $\frac{i}{N} + Q$, where i is the state number, $N = 5$ is the total number of states, and Q is a small uniformly distributed perturbation. Remaining upper triangular entries are initialized to uniformly distributed random values. A left-to-right Markov state assumption requires lower triangular entries to be 0. Normalization is performed so that the entries in each row sum to 1. Initial values for mixture weighting coefficients are random uniformly distributed values between 0 and 1, again with the constraint that row entries sum to one. Initial values of μ and Σ are obtained by first performing a codebook training procedure for all training data to obtain a set of mean vectors and then calculating sample covariance matrices to correspond to each mean vector from the training data vectors which quantize to a particular mean vector.

Speaker Dependent Noisefree Neutral Evaluation Results for "Break"				
Recognizer	Training Data		Testing Data	
Type	p_d	p_f	p_d	p_f
NT-MCC	1.0	0.0882	1.0	0.0931
T-MCC	1.0	0.0294	1.0	0.0490

Table 1: Detection and false alarm probabilities for two neutral "break" recognizers with thresholds set to show theoretically best possible performance.

Speaker Dependent Lombard Effect Evaluation Results for "Help"		
Recognizer	Testing Data	
Type	p_d	p_f
NT-MCC	1.0	0.0149
T-MCC	1.0	0.0

Table 2: Detection and false alarm probabilities for two Lombard effect "help" recognizers with thresholds set to show theoretically best possible performance.

Multiple Speaker Lombard Effect Evaluation Results for "Break"				
Recognizer	Training Data		Testing Data	
Type	p_d	p_f	p_d	p_f
NT-MCC	1.0	0.0	1.0	0.0133
T-MCC	1.0	0.0	1.0	0.0167

Table 3: Detection and false alarm probabilities for two Lombard effect "break" recognizers with thresholds set to show theoretically best possible performance.

Finally, a score can be calculated for an observation sequence by either summing over all frames the natural log of the sum over all states of the forward variable, or using its' mean.

$$score = \sum_{t=1}^{T} \ln(\sum_i \alpha_t(i)) \tag{5}$$

$$score = \frac{1}{T} \sum_{t=1}^{T} \ln(\sum_i \alpha_t(i)). \tag{6}$$

2.4 The Likelihood Ratio Test

Three methods for setting thresholds for scores include a method to i) determine "theoretically best" performance, ii) set the threshold as the minimum score produced for keyword training tokens, or iii) employ optimal detection theory. Results in Sec. 3 show that the likelihood ratio test can be used to make a decision based on produced scores.

For a recognizer to "detect" a keyword, the score produced must be greater than a pre-determined threshold. It is assumed that each word is submitted to each recognizer (if there are multiple keywords) and that either one and only one recognizer accepts the word as a keyword (or that a scheme is applied to choose among multiple acceptances[2]) or none of the recognizers accept the word. Therefore, performance which depends on the threshold is measured in terms of probability of detection p_d and probability of false alarm p_f[12].

One goal if this study is to determine the difference in performance between recognizers which may or maynot employ a FEANN. Initial recognizers are evaluated by setting their thresholds to the minimum score produced by an instance of the represented keywords for both training and testing data. Although this is not an automatic method for selecting thresholds, it serves to demonstrate the rejection potential benefits of the feature enhancing neural network. In the discussion below, this threshold is used to determine "theoretically best" results.

The first attempt to provide an automatic threshold is to choose the minimum score produced by training data keyword tokens [3]. Unfortunately, performance deteriorates significantly from the 'theoretically best', especially for testing data. This is attributed to the fact that the FEANN weights and adaptation are based on training data, so all testing tokens produce scores which are shifted from those of the training tokens, regardless of whether they are instances of a keyword. We make mention of this because the shift must be accounted for when forming probability density functions from training data for a likelihood ratio test.

The optimal detection scheme is based on a likelihood ratio test. Hypothesis one (H1) is that the submitted word is the desired keyword. Hypothesis zero (H0) is that the word submitted is not the keyword represented by the recognizer. A decision rule can be determined by minimizing the Bayes average cost. For this purpose, the a priori probabilities are assumed to be equal. There is no cost associated with making a correct decision. The decision rule is, if:

$$\frac{p_1(y)}{p_0(y)} \geq \frac{C_{10}}{C_{01}}$$

choose H1, where C_{10} is the cost of choosing H1 when the correct decision is H0, C_{01} is the cost of choosing H0 when the correct decision is H1, and $\frac{p_1(y)}{p_0(y)}$ is the likelihood ratio.

[2]To fuse decisions if there are multiple acceptances, the input can be classified as an instance of the keyword whose recognizer produced the score with the greatest positive distance from its threshold.

To find $p_1(y)$ and $p_0(y)$, Maxwell probability density functions (pdf's) are fit to sample pdf's obtained from scores under each hypothesis for training data. The Maxwell pdf is formed as follows:

$$f(x) = \frac{\sqrt{2}x^2}{\alpha^3\sqrt{\pi}} e^{(\frac{-x^2}{2\alpha^2})} \qquad \text{with mean:} \quad \mu = 2\alpha\sqrt{\frac{2}{\pi}}, \tag{7}$$

which yields the following probability density function:

$$f(x) = \frac{\mu x^2}{2\alpha^4} e^{(\frac{-\sqrt{2}x^2}{\alpha\mu\sqrt{\pi}})}. \tag{8}$$

3 Neutral & Lombard Keyword Recognizers

The speech database used for noisefree neutral and Lombard effect recognition evaluations is presented in [13, 14]. Tokens of 35 aircraft communication words were spoken by 9 speakers to create the portion of the database used here. Speaker-dependent keyword recognizers were developed and evaluated using noisefree neutral and Lombard effect data from a "general" (dialect-neutral American) speaker. A multiple speaker keyword recognizer was developed and evaluated using Lombard effect speech from each of the 9 speakers. Noisefree Lombard effect speech was obtained by having speakers produce speech while listening to 85 dB SPL pink noise through headphones. For each of the three recognizers, one of the words was selected to be the represented keyword, and all tokens of the remaining words were used to estimate p_f. For the evaluations, SCHMM's were first trained using non-transformed mel-cepstral coefficients (NT-MCC). Separate SCHMM's were trained on the coefficients resulting from the keyword-dependent transformation (T-MCC) provided by the FEANN.

3.1 "Theoretically Best" Threshold Evaluations

"Theoretically Best" evaluations show that FEANN reduces the number of incorrectly accepted tokens for a recognizer for "break" under neutral conditions and reduces the number of incorrectly accepted tokens for "help" for Lombard effect speech. Results for a multiple speaker "break" recognizer for Lombard effect speech are presented as well. Extensive results are presented in [3]. Scores for speaker dependent evaluations were obtained using Eqn. 5. Multiple speaker scores were obtained using Eqn. 6. Neutral results are presented in Table 1.

Results show that FEANN reduced the number of incorrectly accepted tokens for "break" for the neutral case by $\frac{2}{3}$ for training data and nearly $\frac{1}{2}$ for test data. Fig. 3 shows receiver operating characteristics formed based on the testing data for the keyword "break" for NT-MCC (solid line) and T-MCC (dotted line) recognizers.

Lombard effect results for "help" for recognizers trained using noisefree neutral data are presented in table 2. The results show that the recognizer which used the FEANN made no false acceptances.

Multiple speaker Lombard effect results for "break" are presented in Table 3. Both recognizers were trained using Lombard effect data from all 9 speakers. The results show improved rejection probabilities versus the speaker dependent case, but that T-MCC performance was lower than NT-MCC. These results suggest two observations; first, that additional training data for keyword recognition under Lombard effect does improve performance, and second, that increased intra-speaker variability of speech under Lombard effect may require some form of an adaptive FEANN across speakers. Statistical analysis of speech under stress suggests that different stress relayers may be emphasized depending on individual speaker traits [13].

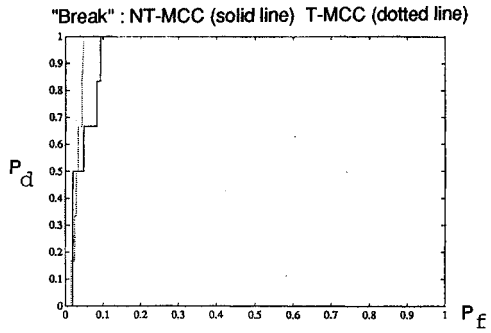

Figure 3: Theoretically best receiver operating characteristics for NT-MCC and T-MCC "break" recognizers.

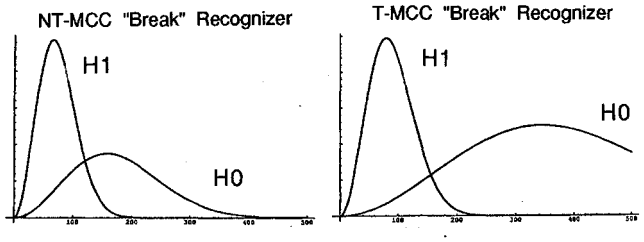

Figure 4: i) Probability density functions for the speaker dependent neutral NT-MCC "break" recognizer for both hypotheses. ii) Probability density functions for the speaker dependent neutral T-MCC "break" recognizer for both hypotheses.

3.2 Minimum Average Cost Threshold Evaluations

A likelihood ratio test was added to both the speaker dependent neutral T-MCC "break" recognizer and the multiple speaker Lombard effect T-MCC "break" recognizer.

To fit the Maxwell probability density functions as discussed, sample means and pdf's were provided by training data scores under both hypotheses. The values of α corresponding to the optimal (in the least mean square sense) pdf's were found using an algorithm based on simulated annealing.

Because sample pdf's could not be provided based on a histogram for H1 due to the small number of H1 training tokens, each score was assigned an equal probability, so that the "best fit" Maxwell density approximated a uniform density. For neutral recognition, each H1 token was assigned a probability equal to the largest probability from the histogram formed from the corresponding H0 tokens. A similar scaling process was applied for the Lombard recognizer. Two different processes were necessary because different scoring procedures were used for the two cases, as noted above. The mean for the neutral case was adjusted to account for the shift in H1 training data scores due to the FEANN.

Two sets of "best fit" pdf's are shown in figure 4 for neutral "break" recognizers. The FEANN has the effect of increasing the variance of scores under H0 especially, causing the pdf's under each hypothesis to "separate" more for the T-MCC recognizer. The increased separation yields improved performance.

A "semi-open" ROC for each T-MCC recognizer was obtained by varying the threshold of the likelihood ratio test for training and testing data (see Fig. 5). The speaker dependent ROC follows closely the ROC pictured in figure 3 for the T-MCC recognizer. The fact that the ROC obtained by applying a likelihood ratio test closely matches the "theoretically best" ROC verifies that reliable probability density functions can be formed from training data scores and used to make optimal decisions for testing data. For the multiple speaker recognizer, performance is near the theoretically best possible, as is shown by the multiple speaker ROC.

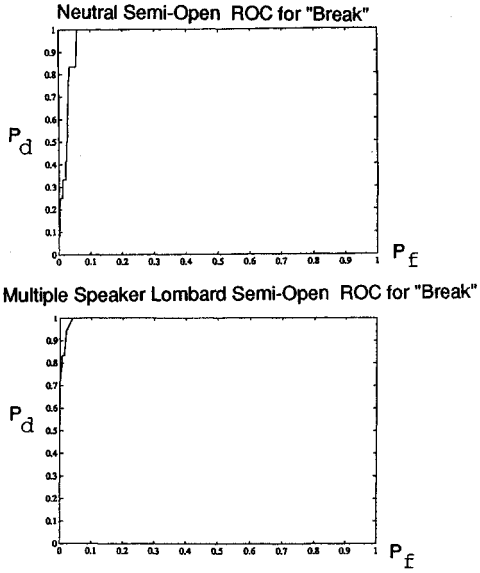

Figure 5: i) Receiver operating characteristic for the speaker dependent neutral T-MCC "break" recognizer. ii) Receiver operating characteristic for the multiple speaker Lombard effect T-MCC "break" recognizer.

4 Conclusions

This paper demonstrates a newly formulated speech recognizer for keyword spotting. Both noisefree neutral and Lombard effect speech were used to evaluate the proposed algorithm. A likelihood ratio test has been shown to effectively allow recognition decisions in an automatic manner. A feature enhancing artificial neural network can improve recognizer performance under both noisefree neutral and Lombard effect conditions.

References

[1] G.J. Clary and J.H.L. Hansen, "Feature Enhancement for Multi-layer Perceptron and Semi-Continuous Hidden Markov Model Based Classifiers Using Neural Networks," *SPIE 1992 Inter. Sym. on Optical Applied Sci. and Eng.*, pp. 1766-54.1-12, San Diego, CA, July 1992.

[2] J.H.L. Hansen, "A New Speech Enhancement Algorithm Employing Acoustic Endpoint Detection and Morphological Based Spectral Constraints," *IEEE Proc. ICASSP 1991*, pp. 901-904.

[3] G.J. Clary, *A Tandem Neural Network and HMM Based Speech Recognizer for Keyword Spotting*, M.S. Thesis, Department of Electrical Engineering, Duke University, Durham, N.C., Feb. 1992.

[4] J.G. Wilpon, L.R. Rabiner, C.H. Lee, E.R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. ASSP*, Nov., 1990, pp. 1870-1878.

[5] J.H.L. Hansen, "Detection and Recognition of Key Words under Noisy, Stressful Conditions," Duke Univ. Tech. Report DSPL-92-2, Grant No. NSF-IRI-90-10536, National Science Foundation, 248 pgs, March 1992.

[6] A.H. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. ASSP*, March, 1989, pp. 328-339.

[7] T. Leen, M. Rudnick, and D. Hammerstrom, "Hebbian Feature Discovery Improves Classifier Efficiency," *Proc. IEEE IJCNN*, pp. I-51-56, 1990.

[8] X.D. Huang, Y. Ariki, and M.A. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh: Edinburgh University Press, 1990.

[9] E. Lombard, "Le Signe de l'Elevation de la Voix," *Ann. Maladies Oreille, Larynx, Nez, Pharynx*, vol. 37, pp. 101-119, 1911.

[10] J.H.L. Hansen, O.N. Bria, "Lombard Effect Compensation for Robust Automatic Speech Recognition in Noise," *Proc. ICSLP 1990*, pp. 1125-1128.

[11] X.D. Huang, "Phoneme classification using semi-continuous hidden Markov models," *IEEE Trans. SP*, May, 1992, pp. 1062-1067.

[12] A.D. Whalen, *Detection of Signals in Noise*, Academic Press, 1971.

[13] J.H.L. Hansen, "Analysis and Compensation of Stressed and Noisy Speech With Application to Robust Automatic Recognition," Ph.D. Thesis, Georgia Inst. of Tech., 428 pages, July 1988.

[14] J.H.L. Hansen, M.A. Clements, "Stress compensation and noise reduction algorithms for robust speech recognition," *Proc. IEEE ICASSP 1989*, pp. 266-269.