# SIMULTANEOUS ANN FEATURE AND HMM RECOGNIZER DESIGN USING STRING-BASED MINIMUM CLASSIFICATION ERROR (MCE) TRAINING

*Mazin G. Rahim and Chin-Hui Lee*[†]
*AT&T Research Laboratories, Murray Hill, NJ 07974*
[†] *Lucent Technologies, Bell Laboratories, Murray Hill, NJ 07974*

## ABSTRACT

Conventional features used in state-of-the-art hidden Markov model (HMM) based speech recognition systems are commonly inspired by *scientific knowledge* and *expertise* of the human vocal and auditory system. Although the intent when performing feature analysis is to extract "relevant" and "discriminative" information from the signal that is useful for speech recognition, this information may *not* be consistent with the objective of minimizing error rate in the recognition process. In this paper, we utilize feed-forward artificial neural networks (ANNs) to generate a new class of features for speech recognition. We propose a system for integrating the feature extraction process with the recognition process under a unified statistical framework with a consistent objective function that is designed to minimize recognition error rate. Results on a telephone-based speaker-independent connected digit task indicate that this integrated system with 12 ANNs is able to reduce the per digit error rate by a further 28% over a similar system using a single ANN and 16% over our previously best results in which feature transformation was not incorporated.

## 1. INTRODUCTION

The two basic components of an automatic speech recognition (ASR) system are feature extraction and speech recognition. Feature extraction includes analyzing the speech signal and converting it into a set of features, such as cepstrum and energy. Speech recognition uses stochastic modeling (e.g., HMMs) to convert input features into a set of meaningful symbols. From the feature side, one desires to extract a set of coefficients that are invariant to extraneous environmental conditions and carry all the discriminative information necessary to perform speech recognition. From the recognizer side, the objective is to train the model parameters so that to provide the best possible class discrimination. Although current ASR systems do apply discriminative training techniques for designing the recognizer (e.g., [8]), little effort has been done in extending this concept to designing discriminative features that enhance the interaction process between feature analysis and recognizer design.

In an effort to minimize recognition error rate by enhancing feature extraction, several studies have incorporated knowledge of the recognition process during feature design. For example, Hunt et. al. [7] proposed a linear discriminative analysis (LDA) method which applies a transformation matrix to conventional features in order to maximize a suitable criterion of class separability. In pattern classification, several studies such as Katagiri et. al. [9], Biem and Katagiri [2], Watanabe et. al. [15] and Paliwal et. al. [11], have shown improved classification performance when applying minimum classification error (MCE) training for both feature and classifier design. In speech recognition, Bengio et. al. [1], Bridle and Dodd [3] and Euler [6] described systems that apply discriminative training techniques for both feature transformation and HMM recognizer design.

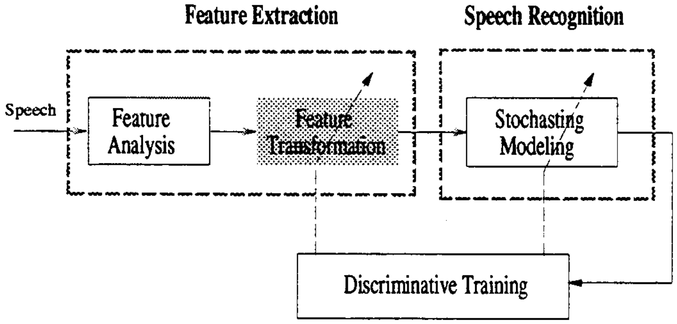

Figure 1. Speech recognition system design with feature transformation [12,13].

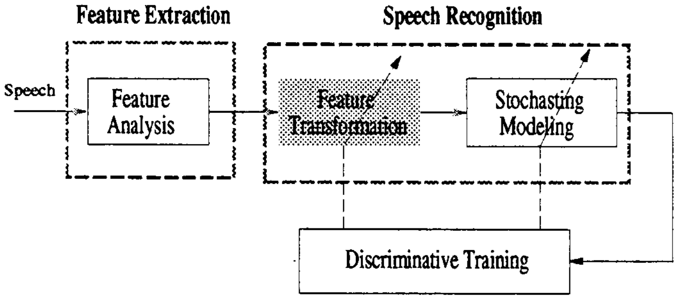

Figure 2. Proposed speech recognition system design with feature transformation.

In [12, 13], we proposed integrating both feature extraction and classifier design into a single training process. A simplified block diagram of the system is shown in Fig. 1. The integrated system adopted a single ANN with a linear activation function to conduct feature transformation, and a set of context-dependent sub-word HMMs to perform stochastic modeling. Thus training of the integrated system included adjusting the connection weights and offsets of the ANN, and the means, variances and mixture gains of the HMMs. We applied the MCE method for discriminative training which essentially helped in two ways. One was to train the parameters of the integrated system simultaneously and discriminatively, thus maximizing class separability and the other was to provide a forward-backward interaction between the feature extractor and the recognizer under a unified objective function.

In this study, we extend the capacity of our integrated system to enable feature transformation to be performed as part of the recognition process as shown in Fig. 2. The intent is to apply a different transformation to each feature vector depending on the state or the unit model the vector is assigned to during decoding. Thus class-specific ANNs are used and trained along with their corresponding HMMs under a unified framework using the MCE framework. The nature of the ANN transformation proposed in this study could reflect some of the correlation that may exist in the empirical cepstral coefficients. This is particularly important since our ASR system adopts diagonal covariances, rather than full covariances, when estimating the observation distributions.

## 2. STRING-BASED MCE TRAINING

Unlike ML estimation which maximizes a likelihood function of a sequence of observations given a set of HMMs, in MCE training [4, 8], the goal is to maximize class separability by minimizing the expected recognition error rate over the entire training data. This has the advantage in helping to improve the robustness and the generalization property of the speech recognition system. Although a "class" in this context may refer to a linguistic or an acoustic unit, this paper uses this concept to refer to an entire string [4].

Consider a training phase of the system shown in Fig. 2 where the parameters of the feature transformation, $\Theta$, and the stochastic modeling, $\Lambda$, are to be adjusted using a corpus of $P$ input strings, $\{\mathbf{O}^p\}$. If $\mathbf{O}^p$ is a sequence of $T$ frames, $\mathbf{O}_1^p, \mathbf{O}_2^p, .., \mathbf{O}_T^p$, that belongs to class string $i$, then the objective in MCE training is to minimize the expected value of the class loss function

$$E[l\{d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}]. \tag{1}$$

$l\{\cdot\}$ is a loss function which can have a non-linear activation, $d_i(\cdot)$ is a *misclassification* measure and $\mathcal{F}(\mathbf{O}^p;\Theta)$ is a transformation which operates on the feature vector sequence $\mathbf{O}^p$ with $\Theta$ as the parameters associated with the transformation $\mathcal{F}(\cdot)$. Minimizing the loss function in Eqn. (1) is achieved by optimizing any one or all of the quantities $l\{\cdot\}$, $d_i(\cdot)$, $\mathcal{F}(\cdot)$, $\Theta$, and/or $\Lambda$. Over the past several years, much work has been done in optimizing $l\{\cdot\}$, $d_i(\cdot)$, and $\Theta$, without imposing any functional transformation on the features $\mathbf{O}$. In this section, we will consider a general transform function $\mathcal{F}(\cdot)$ and provide mathematical formulation for the MCE method when performing joint estimation of the parameters associated with this function, i.e., $\Theta$, as well as the stochastic model parameters, $\Lambda$.

There are essentially three steps when performing MCE training. The first step includes defining the misclassification measure in Eqn. (1). In light of the work done by Katagiri *et. al.* [9], this measure is written as a generalized log likelihood ratio of the form

$$d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda) = -g_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda) + G_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda), \tag{2}$$

where $g_i(\cdot)$ is a discriminant function which is equal to the average likelihood, $\mathcal{L}(\cdot)$, of the correct class $i$:

$$g_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda) = \frac{1}{T}\sum_{t=1}^{T}\log\mathcal{L}(\mathcal{F}(\mathbf{O}_t^p;\Theta);\Lambda_i). \tag{3}$$

And $G_i(\cdot)$ is considered as an anti-discriminant function to class string $i$ which is computed by averaging the values of the discriminant functions for the $N-1$ competing strings to $i$:

$$G_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda) = \log\left[\frac{1}{N-1}\sum_{j,j\neq i}^{N}\exp\{\eta g_j(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}\right]^{\frac{1}{\eta}}, \tag{4}$$

where $\eta$ is a positive constant. Competing string classes are identified using an N-best search [4].

Having computed a value for $d_i(\cdot)$, the next step is to define a measure of error count. One possibility is to use a loss function which is characterized by a smooth 0-1 sigmoid of the form

$$l\{d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\} = \frac{1}{1+\exp\left[-a\cdot d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)+e\right]}, \tag{5}$$

where $a$ and $e$ are constants which control the slope and the shift of the smoothing function, respectively.

Based on the criterion set in Eqn. (1), MCE training involves minimizing the expected value of the loss function in Eqn. (5) by adjusting the parameters of either $\Lambda$ or/and $\Theta$. This is the third and final step in MCE training which effectively maximizes the separation of string $i$ from other competing strings. Due to the lack of a closed form solution to this problem, $\Lambda$ and $\Theta$ are updated using the generalized probabilistic descent (GPD) method where at the $n^{th}$ iteration

$$\Gamma_{n+1} = \Gamma_n - \epsilon_n V_n \nabla l\{d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}|_{\Gamma=\Gamma_n}, \qquad \epsilon_n > 0. \tag{6}$$

$\Gamma = \{\Theta, \Lambda\}$, $\epsilon_n$ is a learning rate and, $V_n$ is a positive definite matrix.

Updating $\Lambda$ and $\Theta$ according to Eqn. (6) requires finding the gradient $\nabla l\{d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}$. In this study, we assume that the transformation function $\mathcal{F}(\cdot)$ and the stochastic modeling are represented by a feed-forward ANN and a HMM, respectively. Therefore, the partial derivative is written as[1]

$$\frac{\partial l_i}{\partial\Gamma} = \frac{\partial l_i}{\partial d_i}[\frac{\partial d_i}{\partial g_i}\cdot\frac{\partial g_i}{\partial\Gamma} + \frac{\partial d_i}{\partial G_i}\cdot\frac{\partial G_i}{\partial\Gamma}]. \tag{7}$$

It follows from Eqn. (5) that

$$\frac{\partial l_i}{\partial d_i} = a\cdot l_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\cdot[1 - l_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)], \tag{8}$$

from Eqn. (2),

$$\frac{\partial d_i}{\partial g_i} = -1, \qquad\qquad \frac{\partial d_i}{\partial G_i} = +1, \tag{9}$$

and finally, from Eqn. (4)

$$\frac{\partial G_i}{\partial\Gamma} = \sum_{j,j\neq i}^{N}\frac{\exp\{\eta\cdot g_j(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}\cdot\frac{\partial g_j}{\partial\Gamma}}{\sum_{m,m\neq i}^{N}\exp\{\eta\cdot g_m(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}}. \tag{10}$$

The remaining partial derivative $\partial g_i/\partial\Gamma$ (or $\partial g_j/\partial\Gamma$ in Eqn. (10)) is formulated differently depending whether we are updating $\Theta$ or $\Lambda$. The process of updating the parameters of $\Lambda$ is described in [4]. In the following section, we will extend the basic formulation of MCE training to update the parameters of $\Theta$.

### Optimizing the ANN parameters

Feature transformation is performed using feed-forward ANNs. The networks are essentially MLPs with cascaded layers of nodes that are fully interconnected via weights and internal thresholds which represent the parameters of the network. Supervised training of MLPs has traditionally been done using the error back-propagation algorithm.

[1] The subscript $n$ in $\Gamma_n$ is neglected from this point forward.

In [12], it was shown that applying the back-propagation algorithm for training an ANN embedded within the system shown in Fig. 1 raises several problems. For example, minimizing a function based on the mean square error criterion as suggested in the back-propagation algorithm is not related to the objective of minimizing the recognition error rate. In addition, the target values of the network are not available during the training process. To alleviate the difficulty of the back-propagation algorithm in the absence of available target data, a new training objective is introduced in this section. With the aid of the stochastic model, a loss function based on minimizing recognition error rate is used as an alternative measure of fit in back-propagation training. The aim is to train the ANN parameters discriminatively by minimizing the recognition loss function of Eqn. (5) rather than a mean square error function. Thus, back-propagation training would then have a direct impact on maximizing recognition performance than simply minimizing a mean square error distance that is unrelated in any way to the recognition process.

Now to minimize the loss function in Eqn. (5) by adjusting the ANN parameters, let's first consider one such ANN, $\Theta_\varphi$, which includes a set of connection weights $\{w_{ljk}\}$ and offsets $\{b_{lk}\}$. When applying GPD for updating $\Theta_\varphi$ as suggested in Eqn. (6), the derivative $\nabla l\{d_i(\mathcal{F}(\mathbf{O}^p;\Theta);\Lambda)\}$ is defined according to Eqn. (7) with

$$\begin{aligned}\frac{\partial g_i}{\partial \Theta_\varphi} &= \frac{1}{T}\sum_{t=1}^{T} \frac{1}{\mathcal{L}(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi);s_t,\Lambda_{\phi_t})} \\ &\cdot \frac{\partial \mathcal{L}(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi);s_t,\Lambda_{\phi_t})}{\partial \Theta_\varphi}\cdot\delta(s_t-\phi_{t\alpha}), \quad (11)\end{aligned}$$

where

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi);s_t,\Lambda_{\phi_t})}{\partial \Theta_\varphi} &= \frac{\partial \mathcal{L}(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi);s_t,\Lambda_{\phi_t})}{\partial \mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)} \\ &\cdot \frac{\partial \mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)}{\partial \Theta_\varphi}. \quad (12)\end{aligned}$$

We define

$$\begin{aligned}\frac{\partial \mathcal{L}(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi);s_t,\Lambda_{\phi_t})}{\partial \mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)} &= -\mathcal{N}(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi);\mu_{\phi\alpha\beta},\sigma^2_{\phi\alpha\beta}) \\ &\cdot \mathbf{c}_{\phi\alpha\beta}\frac{(\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)-\mu_{\phi\alpha\beta})}{\sigma^2_{\phi\alpha\beta}} \quad (13)\end{aligned}$$

where $\mathcal{N}(\cdot)$ is a normal distribution of the transformed vectors $\mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)$ having a mean vector $\mu_{\phi\alpha\beta}$ and a standard deviation vector $\sigma_{\phi\alpha\beta}$. For each node in $\Theta_\varphi$ with a corresponding connection weight $w_{ljk}$ and offset $b_{lk}$,

$$\begin{aligned}\frac{\partial \mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)}{\partial w_{ljk}} &= \mathcal{H}' O^p_{t,(l-1)j}, \\ \frac{\partial \mathcal{F}(\mathbf{O}_t^p;\Theta_\varphi)}{\partial b_{lk}} &= -\mathcal{H}'. \quad (14)\end{aligned}$$

Here $\mathcal{H}'$ is the derivative of the activation function, $\mathcal{H}$, which describes the behavior of a unit $j$ belonging to the $l^{th}$ output layer in terms of its net input values, $\{O^p_{t,(l-1)j}\}$. When $\mathcal{H}$ is set to a linear activation function, then the derivative $\mathcal{H}'$ reduces to one.

## 3. DATABASE AND EXPERIMENTAL RESULTS

A speaker-independent telephone-based connected digits database was used in this study. Digit strings ranging from one to sixteen digits in length were extracted from different field-trial collections with varied environmental conditions and transducer equipment. The training set consisted of 16089 digit strings which was used for designing the recognition models. The testing set consisted of 713 digit strings which included 16-digit credit card numbers.

The baseline system adopted in this study is similar to that shown in Fig. 2 but without feature transformation and with strictly ML-trained recognition models. It operates as follows. An input utterance is first segmented every 10 msec intervals into frames of 30 msec duration. Each frame is then processed to give 12 LPC-derived liftered cepstral coefficients along with a normalized energy feature. The combined feature vector is augmented with its first and second order time derivatives to generate a vector of 39 features per frame. Since the signal has been recorded under various telephone conditions and with different transducer equipment, each feature vector was further processed using the hierarchical signal bias removal (HSBR) method [14] in order to reduce the effect of channel distortion.

Following feature analysis, each feature vector is passed to the recognizer which models each word (i.e., digit) in the vocabulary by a set of left-to-right continuous density quasi-triphonic HMMs [10]. A total of 274 context-dependent sub-word models were used, each being trained with ML estimation. Sub-word models consisted of 3 to 4 states with each state having a mixture of 8 Gaussian components. In order to enable the background/noise model to capture more acoustic variations in the training data, it was designed to have a single state with 32 Gaussian components.

Table 1 presents results of the baseline system (labeled as "Baseline-ML") when using ML-trained recognition models without imposing feature transformation. Word insertion, deletion and substitution rates ("%Ins,Del,Sub") are presented as well as word error rate ("%Wd_er") and string error rate ("%St_er").

Table 1. Recognition results including word and string error rates.

| System | %Ins,Del,Sub | %Wd_er | %St_er |
|---|---|---|---|
| Baseline-ML | 0.63,0.14,1.02 | 1.80 | 17.2 |
| HMM-MCE | 0.17,0.16,0.82 | 1.14 | 11.7 |
| ANN-MCE | 0.08,0.27,1.03 | 1.39 | 13.4 |
| ANN12-MCE | 0.03,0.26,0.72 | 1.00 | 10.7 |
| Integ-ANN | 0.05,0.24,0.84 | 1.14 | 11.3 |
| Integ-ANN12 | 0.03,0.27,0.66 | 0.96 | 10.0 |

The next stage in our work was to evaluate the recognition system when further training the HMMs with the MCE method. This procedure included updating the HMM parameters as suggested in [4], and performing an integrated MCE/HSBR training [5]. Table 1 shows the results following six iterations of training (second row labeled as "HMM-MCE"). Those results correspond to our state-of-the-art system that was reported in [5]. They indicate a reduction in the word and string error rates by about 37% and 32%, respectively. The major improvement comes from a reduction in the substitution rate and the insertion rate at a relatively no cost to the deletion rate.

In our next set of experiments we evaluated the recognition system in Fig. 2 when having the ML-trained HMMs with either a single ANN or multiple ANNs performing feature transformation. Note that the two systems in Fig. 1

and Fig. 2 perform identically when using a single ANN. Each ANN which was designed to have a linear activation function was trained using the MCE method. In these experiments, the network weights and offsets were adjusted as suggested in Eqns. (11)-(14). Prior to MCE training, network parameters were initialized to perform a "self" mapping (i.e., copying the input values to the output nodes). Selecting such an initialization scheme enables each ANN to be embedded within the system shown in Fig. 2 (or Fig. 1) without affecting the overall baseline performance and without changing the distributions of the HMMs.

Recognition results when applying the MCE method for a single ANN or 12 ANNs are presented in the third and fourth rows in Table 1 (labeled as "ANN-MCE" and "ANN12-MCE", respectively). In the case of 12 ANNs, a neural net was assigned for each digit, including the background/noise model. Note that each ANN was not "fully" connected as paths between different streams (e.g., cepstrum and delta-cepstrum) were disjoined in the hope of improving generalization. The results shown in Table 1 suggest that even a single ANN with a ML-trained HMM can results in a moderate improvement in recognition accuracy. It is surprising that such a small sized network with limited number of parameters could produce an improvement that is not significantly different than that achieved for "HMM-MCE". However, it is clear that most of the improvement is obtained as a result of a reduction in the insertion rate and not the substitution rate. This is certainly not the case for "ANN12-MCE" where the word error rate is less than that achieved for "HMM-MCE" due to the lower substitution error rate. In fact using multiple ANNs as opposed to a single ANN has resulted in a further reduction in word and string error rates of 28% and 20%, respectively.

Finally, we performed an experiment integrating the training of the ANNs and the HMMs within the proposed unified MCE framework. Upon initializing the system in Fig. 1 with ML-trained HMMs and self-mapped ANNs, the integrated system was then trained by applying the MCE method for six iterations. The results of this process for a single and multiple ANNs are shown in the fifth and sixth rows in Table 1 (labeled as "Integ-ANN" and "Integ-ANN12 respectively). Although we observed a substantial reduction in error rate during training, the testing results seem to be somewhat indifferent as shown in the Table.

## 4. SUMMARY

This paper proposed a system for ANN feature and HMM recognizer design using minimum classification error training. The intent was to combine training of ANNs which were introduced for feature transformation with HMMs which were applied for stochastic modeling under a unified objective function that minimizes recognition error rate. A preliminary study using 12 feed-forward ANNs with a linear activation function was reported in which the parameters of the networks were optimized using the MCE objective function. Experimental results on a field-trial connected digit task have demonstrated a further 28% reduction in digit error rate over a single ANN. When integrating the training of the ANNs and the HMMs under the MCE framework, the word and string error rates were reduced by about 16% and 25%, respectively, over our previously reported high-performance system in which discriminative training was applied to the HMMs only.

The particular set up of our integrated system provides several benefits that are worth mentioning. These include the generation of task-specific features and models which could facilitates new understanding of speech/speaker characterization. Also, by including a system of a single or multiple ANNs using a non-linear activation function we have the potential to carry out complex transformations such as a direct mapping from the speech signal itself to a set of discriminative recognition features. This set-up also provides the ability to perform discriminative feature reduction, thus minimizing computational effort as well as providing a more robust speech recognition system.

## REFERENCES

[1] Bengio, Y., De Mori, R., Flammia, G and Kompe, R. (1990) "Global Optimization of a Neural Network - Hidden Markov Model Hybrid," *McGill University Technical Report*, TR-SOCS-90.22.

[2] Biem, A., and Katagiri, S. (1993) "Feature Extraction Based on Minimum Classification Error/Generalized Probabilistic Descent Method," *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, **2**, pp.275-278.

[3] Bridle, J. S., and Dodd, L. (1991) "An Alphanet Approach to Optimizing Input Transformations for Continuous Speech Recognition," *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, **1**, pp. 277-280.

[4] Chou, W., Juang, B.-H., and Lee, C.-H., and Soong, F. K. (1994) "A Minimum Error Rate Pattern Recognition Approach to Speech Recognition," *J. Pattern recognition and Artificial Intelligence*, **8**(1), pp. 5-31.

[5] Chou, W., Rahim, M., Buhrke, E. (1995) "Signal Conditioned Minimum Error Rate Training," *Proc. European Conf. on Signal Processing*, **1**.

[6] Euler, S. (1995) "Integrated Optimization of Feature Transformation for Speech Recognition," *Proc. European Conf. on Speech Communication and Technology*, **1**, pp. 109-112.

[7] Hunt, M., Richardson, S., Bateman, C., and Piau, A. (1991) "An Investigation of PLP and IMELDA Acoustic Representations and of their Potential for Combination," *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, **1**, pp. 881-884.

[8] Juang, B.-H., and Katagiri, S. (1992) "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Signal Processing*, **40**(12), pp. 3043-3054.

[9] Katagiri, S., Lee, C.-H. and Juang, B.-H. (1991) "Discriminative Multi-layer Feed-forward Networks," *IEEE Proc. Neural Networks for Signal Processing*, pp. 11-20.

[10] Lee, C.-H. Giachin, E., Rabiner, L. R., Pieraccini, R., and Rosenberg, A. E. (1992) "Improved Acoustic Modeling for Large Vocabulary Continuous Speech Recognition," *Computer, Speech & Language*, **6**(2), pp. 103-127.

[11] Paliwal, K. K., Bacchiani, M., and Sagisaka, Y. (1995) "Minimum Classification Error Training Algorithm for Feature Extractor and Pattern Classifier in Speech Recognition," *Proc. European Conf. on Speech Communication and Technology*, **1**, pp. 541-544.

[12] Rahim, M. and Lee, C.-H. (1996) "Joint ANN Feature and HMM Recognizer Design Using String-based Minimum Classification Error Training," *World Congress on Neural Networks*, San Diego.

[13] Rahim, M., Lee, C.-H., and Juang, B-H. (1995) "An Integrated ANN-HMM Speech Recognition System Based on Minimum Classification Error Training," *Automatic Speech Recognition Workshop*, Snowbird.

[14] Rahim, M., Juang, B-H., Chou, W., and Buhrke, E. (1996) "Signal Conditioning Techniques for Robust Speech Recognition," *IEEE Signal Processing Letters*, to be published.

[15] Watanabe, H., Yamaguchi, T., and Katagiri, S. (1995) "Discriminative Metric Design for Pattern Recognition," *Proc. Int. Conf. on Acoust., Speech, and Signal Processing*, **5**, pp. 3439-3442.