Speech Recognition Using Long-Span Temporal Patterns in a Deep Network Model

Sabato Marco Siniscalchi, *Member, IEEE*, Dong Yu, *Senior Member, IEEE*, Li Deng, *Fellow, IEEE*, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—In recent years, there has been a renewed interest in the use of artificial neural networks (ANNs) for speech applications, and it seems that a new trend to move the speech technology forward has begun. Two main contributions have triggered such a new trend: 1) a major advance has been made in training the weights in deep neural networks (DNNs), and a pre-trained deep neural network hidden Markov model (DNN-HMM) hybrid architecture has outperformed a conventional Gaussian mixture model hidden Markov model (GMM-HMM) automatic speech recognition (ASR) system on a challenging business search dataset, and 2) it has been shown that phoneme classification can be boosted by using a hierarchical structure of multi-layer perceptrons (MLPs) trained to model long-span temporal patterns with beneficial effects on language recognition tasks. In this work, we combine these two lines of research and demonstrate that word recognition accuracy can be significantly enhanced by arranging DNNs in a hierarchical structure to model long-term energy trajectories. The proposed solution has been evaluated on the 5000-word Wall Street Journal task, resulting in consistent and significant improvements in both phone and word recognition accuracy rates. We have also analyzed the effects of various modeling choices on the system performance, and several architectural solutions have been compared.**

***Index Terms*—Automatic speech recognition, deep neural networks, large vocabulary continuous speech recognition.**

I. Introduction

ARTIFICIAL neural networks (ANNs) are powerful pattern recognition tools that have been used for several real world applications over the past years, and the notion of using ANNs to improve speech-recognition performance has been around since the latest 1980s. In connectionist speech recognition systems [1], which is the topic of this paper, ANNs are employed to estimate the state emission probabilities of hidden Markov model (HMM) ASR using Bayes rule. These connectionist speech recognizers are commonly referred to as ANN-HMM hybrid systems in the literature. Several neural architectures have been proposed to build ANN-HMM hybrid systems, but the feed-forward multi-layer perceptron (MLP) is by far the most popular as it provides an attractive compromise between recognition rate, recognition speed, and memory resources. These ASR hybrid systems were seen as a promising technique for large vocabulary continuous speech recognition (LVCSR) in the mid-1990s. Indeed, it is known that ANNs are much better than GMMs, commonly used in the speech community to estimate emission probabilities of HMM states, in learning models of data that lie on or near a nonlinear manifold [2]; nonetheless, ANN-HMM systems have never outperformed conventional GMM-HMM systems trained with discriminative parameter estimation techniques [3]–[5]. As discussed in [2], the ANN-HMM hybrid system's performance has been damped down by the lack of adequate algorithms to train ANN with more than a single hidden non-linear layer on large amounts of speech data. Therefore, *shallow* neural architectures have often been adopted for speech applications.

Neural architectures have thus had scarce use in commercial speech-recognition solutions so far except for providing probabilistic features to be appended to the standard short-time spectral features. The so-called TANDEM approach [6], [7] is a successful example of using one or more ANNs to extract speech features based on phoneme posterior probabilities, which are in turn used to augment the input of conventional HMM based ASR systems. Bottle-Neck features, first introduced in [8], are an extension of these TANDEM features, and allow better recognition results. The key difference between TANDEM and Bottle-Neck features is that the latter are obtained as linear outputs of the neurons in the bottle-neck layer. Nevertheless, two key events have revitalized interest in neural architectures for ASR systems in recent years, namely: (1) it has been shown that high-accuracy automatic language recognition systems can be designed by training a hierarchical structure of MLPs to learn long-span temporal patterns [9]–[13]. The rationale behind this success is that longer term dependencies have proven to provide phonetic, sub-lexical and lexical knowledge [14], and (2) a major advance has been made in training densely connected, generative deep belief nets (DBNs) with many hidden layers [15]. The core idea of the DBN training algorithm suggested in [16] is to first initialize the weights of each layer greedily in a purely unsupervised way by treating each pair of layers as a restricted Boltzmann machine (RBM) and then to fine-tune all the weights jointly to further improve the likelihood. The resulting DBN can be considered as a hierarchy of nonlinear feature detectors that can capture complex statistical patterns in data. For classification tasks, the same DBN pre-training algorithm can be used to initialize the weights in deep neural networks (DNNs)—MLPs with many hidden layers. In [17], the context-dependent DNN-HMM was proposed and successfully applied to LVCSR tasks, and good phonetic features classification accuracies were reported in

Manuscript received October 20, 2012; revised December 18, 2012; accepted January 02, 2013. Date of publication January 04, 2013; date of current version January 24, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Frederic Bechet.

S. M. Siniscalchi is with the Faculty of Architecture and Engineering, University of Enna "Kore", 94100 Enna, Italy (e-mail: marco.siniscalchi@unikore.it).

D. Yu and L. Deng are with the Speech Research Group, Microsoft Research, Redmond, WA 98052 USA (e-mail: dong.yu@microsoft.com; li.deng@microsoft.com).

C.-H. Lee is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: chl@ece.gatech.edu).

Digital Object Identifier 10.1109/LSP.2013.2237901

[18]. Pre-training can serve as a regularizer that prevents over-training [19] especially useful when largest neural architectures have more parameters than there are frames in the training set.

In this work, these two lines of research are combined together, and the overall proposed system consists of a cascade of DNNs that are trained on long-term energy trajectories. Evaluation on the Nov92 test set of the 5000-word Wall Street Journal task [20] demonstrates that high-accuracy frame-wise classification can be attained. As for word recognition, the proposed hierarchical structure of DNNs outperforms both a stand-alone DNN trained on short-term spectral features and similar hierarchical configurations that employ MLPs with a single hidden layer. Furthermore, notably improved accuracy is attained by using senones (tied triphone states) [21].

The remainder of the paper is organized as follows: Section II briefly introduces the deep neural architecture used in this work. The experimental setup and results are presented in Section III, where the long-span temporal patterns used in this work are also introduced. Our findings are given in Section IV.

II. Deep Neural Networks

A DNN is a multi-layer perceptron with many hidden layers. The main challenge in learning DNNs is to devise efficient training strategies in order to escape poor local optima of the complicated nonlinear error surface introduced by the large number of hidden layers. A common practice is to initialize the parameters of each layer greedily and generatively by treating each pair of layers in DNNs as a restricted Boltzmann machine (RBM) before performing a joint optimization of all the layers [15]. This learning strategy enables discriminative training to start from well initialized weights and is used in this study.

A. Restricted Boltzman Machines

A bipartite graph with a visible layer and a hidden layer can be used to represent an RBM. The stochastic units in the visible layer only connect to the stochastic units in the hidden layer. The units in the visible layer are typically represented by Bernoulli or Gaussian distributions and the units in the hidden layer are commonly represented with Bernoulli distributions. Gaussian-Bernoulli RBMs can convert real-valued stochastic variables (such as short-term spectral features) to binary stochastic variables that can then be further processed using the Bernoulli-Bernoulli RBMs.

Given the model parameters θ, the joint distribution $p(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta)$ over the visible units $\underline{\mathbf{v}}$ and hidden units $\underline{\mathbf{h}}$ in the RBMs can be defined as

$$p(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta)=\frac{\exp(-E(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta))}{Z}, \tag{1}$$

where $E(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta)$ is an energy function, and the partition function is represented by $Z=\sum_v\sum_h\exp(-E(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta)$. The marginal probability that the model assigns to a visible vector $\underline{\mathbf{v}}$ is

$$p(\underline{\mathbf{v}};\theta)=\frac{\sum_h\exp(-E(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta))}{Z}. \tag{2}$$

For a Bernoulli (visible)-Bernoulli (hidden) RBM, the energy is

$$E(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta)=-\sum_{i=1}^{V}\sum_{j=1}^{H}w_{ij}v_ih_j-\sum_{i=1}^{V}b_iv_i-\sum_{j=1}^{H}a_jh_j, \tag{3}$$

where w_{ij} represents the symmetric interaction between visible unit v_i and hidden unit h_j, b_i and a_j are the bias terms at the visible and hidden layers, respectively, and V and H are the numbers of visible and hidden units. The conditional probabilities can be efficiently calculated as

$$p(h_j=1\,|\,\underline{\mathbf{v}};\theta)=\sigma\left(\sum_{i=1}^{V}w_{ij}v_i+a_j\right), \tag{4}$$

$$p(v_i=1\,|\,\underline{\mathbf{h}};\theta)=\sigma\left(\sum_{j=1}^{H}w_{ij}h_j+b_i\right). \tag{5}$$

where $\sigma(x)=1/(1+\exp(-x))$.

Similarly, for a Gaussian-Bernoulli RBM, the energy is

$$E(\underline{\mathbf{v}},\underline{\mathbf{h}};\theta)=-\sum_{i=1}^{V}\sum_{j=1}^{H}w_{ij}v_ih_j+\frac{1}{2}\sum_{i=1}^{V}(v_i-b_i)^2-\sum_{j=1}^{H}a_jh_j. \tag{6}$$

The corresponding conditional probabilities become

$$p(h_j=1\,|\,\underline{\mathbf{v}};\theta)=\sigma\left(\sum_{i=1}^{V}w_{ij}v_i+a_j\right), \tag{7}$$

$$p(v_i=1\,|\,\underline{\mathbf{h}};\theta)=\mathcal{N}\left(\sum_{j=1}^{H}w_{ij}h_j+b_i,1\right). \tag{8}$$

where v_i takes real values and follows a conditional Gaussian distribution with mean $\sum_{j=1}^{H}w_{ij}h_j+b_i$ and variance one.

The parameters in RBMs can be optimized to maximize log likelihood, log $p(\underline{\mathbf{v}};\theta)$, and can be updated as

$$\Delta w_{ij}=\langle v_i\,h_j\rangle_{\text{data}}-\langle v_i\,h_j\rangle_{\text{model}}, \tag{9}$$

where $\langle v_i\,h_j\rangle_{\text{data}}$ is the expectation that v_i and h_j occur together in the training set and $\langle v_i\,h_j\rangle_{\text{model}}$ is the same expectation under the distribution defined by the model. Because $\langle v_i\,h_j\rangle_{\text{model}}$ is extremely expensive to compute exactly, the contrastive divergence (CD) approximation to the gradient is used, where $\langle v_i\,h_j\rangle_{\text{model}}$ is replaced by running the Gibbs sampler initialized at the data for one full step [15].

B. Deep Neural Network Training Process

The last layer of a DNN transforms a number of Bernoulli distributed units into a multinomial distribution using the softmax operation

$$p(l=k\,|\,\underline{\mathbf{h}};\theta)=\frac{\exp\left(\sum_{i=1}^{H}\lambda_{ik}h_i+a_k\right)}{Z(\underline{\mathbf{h}})} \tag{10}$$

where $l = k$ denotes the input being classified into the kth class, and λ_{ik} is the weight between hidden unit h_i at the last layer and class label k. To learn the DNNs, we first train a Gaussian-Bernoulli RBM generatively in which the visible layer is the continuous input vector constructed from $2n+1$ frames of speech features, and n is the number of look-forward and look-backward frames. We then use Bernoulli-Bernoulli RBMs for the remaining layers. When pre-training the next layer, $E(h_j \mid \underline{\mathbf{v}}; \theta) = p(h_j = 1 \mid \underline{\mathbf{v}}; \theta)$ from the previous layer is used as the visible input vector based on the mean-field theory. This process continues until the last layer, where error back-propagation (BP) is used to fine-tune all the parameters jointly by maximizing the frame-level cross-entropy between the true and the predicted probability distributions over class labels.

III. Experiments

In the following sections, the experimental setup is first presented. Then the experimental results are given and discussed. In all experiments, the classical back-propagation algorithm with cross-entropy error function was used to train the shallow MLPs. The DNN training procedure adopted in this work was outlined in Section II.

A. Experimental Setup

Corpus: The 5,000-word speaker independent WSJ0 (5k-WSJ0) corpus [20] was used for all experiments. The parameters of all classifiers presented in this study were estimated using training material from the SI-84 set (7077 utterances from 84 speakers, i.e., 15.3 hours of speech). A cross-validation (cv) set was generated by extracting 200 sentences out of the SI-84 training set. The cv set accounts for about 3% of the SI-84 set and was used to terminate the training. The remaining 6877 SI-84 sentences were used as training material. Evaluation was carried out on the Nov92 evaluation data (330 utterances from 8 speakers). The number of phonemes was 40.

Speech Features: Two sets of speech features were extracted and evaluated, namely short-term spectral-based features and long-span temporal patterns of spectral energies.

Short-term spectral-based feature: Mel-frequency Cepstral Coefficients (MFCCs) [22] were used as acoustic parametrization of the speech signal. Spectral analysis was performed using a 23 channel Mel filter bank from 0 Hz to 8 kHz. The cepstral analysis was carried out with a Hamming window of 25 ms and a frame shift of 10 ms. For each frame, twelve MFCC features plus the zero-th coefficient were computed. The first and second time derivatives of the cepstra were also computed and appended to the static components to yield a 39-dimensional feature-vector.

Long-span temporal patterns of spectral energies: Split temporal context (STC) features [10] were adopted as long-span temporal patterns. In order to generate these features a hierarchical structure of neural networks was built and trained on long-term energy trajectories. Specifically, spectral analysis was performed using a 23 channel Mel filter bank. For each critical band a window of 310 ms centered around the frame being processed was considered and split in two halves: left-context and right-context. Two independent "lower networks" were trained on those two halves and generated left- and right-context class posterior probabilities, such as (sub-)phoneme posteriors, respectively. The discrete cosine transform was applied to the input of these lower nets to reduce the input dimensionality to 253 (i.e., only 11 coefficients are kept for each critical band). The outputs of the two lower nets were then sent to a third "upper network," which acted as a merger and gave the class posterior probabilities at a (sub-)phoneme level.

Neural Architectures: One- and two-stage configurations were investigated. In the one-stage configuration, a single neural architecture was used. The STC architecture (i.e., a cascade of two neural networks) was adopted for the two-stage configuration. Several neural architectures were thus built, and we here present those solutions that delivered the best performance:

MFC-MLP_PHN: An MLP with a single hidden layer having 2000 nodes is trained to model a window of eleven (11) frames of 39-dimensional MFCC vectors, and produces phoneme posterior probabilities. Therefore, the input size is equal to 429, and the output size is equal to 40.

MFC-DNN_PHN: A DNN with a 5 hidden layers, and 2048 nodes per hidden layer is trained to model a window of eleven (11) frames of 39-dimensional MFCC vectors, and produces phoneme posterior probabilities. Therefore, the input size is equal to 429, and the output size is equal to 40.

STC-MLP_PHN: Shallow MLPs are used to implement the building blocks of the STC features. Each MLP has 1500 hidden nodes and produces phoneme posterior probabilities. Thus, the input size of the "lower nets" is 253, after dimensionality reduction with discrete cosine transform, and the output size is equal to 40. The input size of the "upper net" is therefore 80, and its output size is equal to 40.

STC-DNN_PHN: The difference with the STC-MLP_PHN architecture is that DNNs with 5 hidden layers, and 2048 nodes per hidden layer are used in place of shallow MLPs.

MFC-MLP_SEN: A shallow MLP with a single hidden layer having 2000 nodes was trained to model a window of eleven (11) frames of 39-dimensional MFCC vectors, and produces senone posterior probabilities. Senones were generated using the procedure outlined in [21], and the final number of tied triphone states was equal to 2818. Therefore, the input size is equal to 429, and the output size is equal to 2818.

MFC-DNN_SEN: With respect to the *MFC-MLP_SEN* configuration, a DNN with 5 hidden layers, and 2048 nodes per hidden layer is used.

STC-DNN_SEN: In this configuration senones are used along with STC features generated with DNNs. If we were to use senones to train "lower level" DNNs, the input dimension of the feature vectors for the "upper net" will be too big, and that may lead to over-fitting issues. To overcome this problem, the "lower nets" produce phoneme posterior probabilities; whereas, the "upper net" uses posterior probabilities of senones as network outputs (i.e., 2818 target classes). All DNN have 5 hidden layers, and each hidden layer has 2048 nodes.

The total number of parameters for each neural architecture is reported (in parentheses) in the second column of Table I.

B. Experimental Results

Table I reports frame-wise classification results for the above mentioned neural architectures. By a visual inspection of this

TABLE I
FRAME CLASSIFICATION RATES (IN %) WITH DIFFERENT NEURAL ARCHITECTURES. NOTE THAT THE NUMBER OF SENONE CLASSES IS MUCH HIGHER THAN THAT OF PHONE CLASSES

Output Class	Neural Architecture	Frame Accuracy
Phonemes	*MFC-MLP_PHN* (940,040)	79.8
	MFC-DNN_PHN (16,957,312)	85.3
	STC-MLP_PHN (1,063,620)	85.4
	STC-DNN_PHN (51,808,376)	88.3
Senones	*MFC-MLP_SEN* (6,498,818)	54.8
	MFC-DNN_SEN (23,440,130)	66.3
	STC-DNN_SEN (57,500,498)	69.6

TABLE II
WORD ERROR RATES (WERS) (IN %) ON THE NOVEMBER 1992 TEST SENTENCES WITH DIFFERENT HYBRID ASR CONFIGURATIONS

Output Class	Hybrid Architecture	WER on Nov'92 test set
Phoneme	*STC-MLP_PHN*-HMM	6.3%
	STC-DNN_PHN-HMM	6.0%
Senones	*MFC-DNN_SEN*-HMM	5.7%
	STC-DNN_SEN-HMM	5.2%

table, it can be concluded that the two-stage configuration always leads to higher accuracies. Therefore, classification results can be hierarchically improved. Second, DNNs always boost system performance independently of the input speech features. Table II shows the word error rates (WERs) attained with hybrid ASR systems with the two best neural architectures using phoneme-based output classes, and two best solutions with senone-based output classes. It can be argued that the higher the classification accuracy is, the lower the WER is. Therefore, there is a tight link between quality of the posterior estimates and ASR performance. Lower WERs can be attained by using posterior probabilities of senones as network outputs, and DNNs always boost recognition accuracy. The use of hierarchical structure of neural architectures has also a beneficial effect on LVCSR task. Finally, the *STC-DNN_SEN*-HMM hybrid system significantly outperforms all of the other ASR system reported in Table II with a final WER of 5.2%. To the best of the authors' knowledge this represent the best reported results on the Nov'92 task obtained using an ANN-HMM hybrid system, and it therefore confirms our initial intuition: LVCSR performance can be improved by combining DNNs and long-span temporal patterns.

IV. SUMMARY

Experimental evidence has shown that two-stage configurations allow for better frame-classification accuracies. System performance can also be improved by replacing MLP with a single non-linear hidden layer with deep neural architectures with 5 hidden layers. Long-span temporal patterns, previously proven useful for automatic language recognition [9], have been here successfully applied to a LVCSR task. Furthermore, it has been proven that by arranging DNNs in a hierarchical structure to model long-term energy trajectories, the word recognition accuracy can be significantly enhanced, and the best WER of 5.2% has been reported on the WSJ0 task using a ANN-HMM hybrid system. We believe that such a neural architecture can play a major rule also in detection-based approach to ASR [23].

REFERENCES

[1] H. Bourlard and N. Morgan*, Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA, USA: Kluwer, 1994.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[3] L. R. Bahl, P. F. Brown, P. V. deSouza, and R. L. Mercer, "Maximum mutual information estimation of HMM parameters for speech recognition," in *Proc. ICASSP*, Tokyo, Japan, Apr. 1986, pp. 49–52.

[4] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification minimum error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, 1997.

[5] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP*, Orlando, FL, USA, May 2002, pp. 105–108.

[6] H. Hermansky and S. Sharma, "Temporal patterns (TRAPS) in ASR of noisy speech," in *Proc. ICASSP*, Phoenix, AZ, USA, Mar. 1999, pp. 289–292.

[7] H. Hermansky, S. Sharma, and P. Jain, "Data-derived non-linear mapping for feature extraction in HMM," in *Proc. ASRU*, Keystone, CO, USA, Dec. 1999.

[8] F. Grézl, M. Karafia, S. Kontar, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP*, Honolulu, HI, USA, Apr. 2007, pp. 757–760.

[9] P. Matějka, P. Schwarz, J. Černocký, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Proc. Interspeech*, Lisboa, Portugal, Sep. 2005, pp. 2237–2240.

[10] P. Schwarz, P. Matějaka, and J. Cernock, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, Toulouse, France, May 2006, pp. 325–328.

[11] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploring universal attribute characterization of spoken languages for spoken language recognition," in *Proc. Interspeech*, Brighton, U.K., Sep. 2009, pp. 168–171.

[12] S. M. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, pp. 875–887, 2012.

[13] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Universal attribute characterization of spoken languages for automatic spoken language recognition," *Comput. Speech Lang.*, vol. 27, no. 1, pp. 209–227, 2013.

[14] P. Fousek and H. Hermansky, "Towards ASR based on hierarchical posterior-based keyword recognition," in *Proc. ICASSP*, Toulose, France, May 2006, pp. 433–436.

[15] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, pp. 1771–1800, Aug. 2002.

[16] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computat.*, vol. 18, pp. 1527–1554, July 2006.

[17] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 20, pp. 30–42, 2012.

[18] D. Yu, S. M. Siniscalchi, L. Deng, and C.-H. Lee, "Boosting attribute and phone estimation accuracy with deep neural networks for detection-based speech recognition," in *Proc. ICASSP*, Kyoto, Japan, Mar. 2012, pp. 4169–4172.

[19] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *Proc. AISTATS*, Clearwater Beach, FL, USA, Apr. 2009, pp. 153–160.

[20] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Proc. Workshop on Speech and Natural Language*, Banff, AB, Canada, Oct. 1992, pp. 899–902.

[21] M.-Y. Hwang and X. Huang, "Shared-distribution hidden Markov models for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 1, pp. 441–320, Oct. 1993.

[22] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllable word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, 1980.

[23] I. Bromberg, Q. Fu, J. Hou, J. Li, C. Ma, B. Matthews, A. Moreno-Daniel, J. Morris, S. Siniscalchi, Y. Tsao, and Y. Wang, "Detection-based ASR in the automatic speech attribute transcription project," in *Proc. Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 1829–1832.