Analysis and Comparison of Two Speech Feature Extraction/Compensation Algorithms

Li Deng, *Fellow, IEEE*, Jian Wu, *Member, IEEE*, Jasha Droppo, *Member, IEEE*, and Alex Acero, *Fellow, IEEE*

***Abstract*—Two feature extraction and compensation algorithms, feature-space minimum phone error (fMPE), which contributed to the recent significant progress in conversational speech recognition, and stereo-based piecewise linear compensation for environments (SPLICE), which has been used successfully in noise-robust speech recognition, are analyzed and compared. These two algorithms have been developed by very different motivations and been applied to very different speech-recognition tasks as well. While the mathematical construction of the two algorithms is ostensibly different, in this report, we establish a direct link between them. We show that both algorithms in the run-time operation accomplish feature extraction/compensation by adding a posterior-based weighted sum of "correction vectors," or equivalently the column vectors in the fMPE projection matrix, to the original, uncompensated features. Although the published fMPE algorithm empirically motivates such a feature extraction operation as "a reasonable starting point for training," our analysis proves that it is a natural consequence of the rigorous minimum mean square error (MMSE) optimization rule as developed in SPLICE. Further, we review and compare related speech-recognition results with the use of fMPE and SPLICE algorithms. The results demonstrate the effectiveness of discriminative training on the feature extraction parameters (i.e., projection matrix in fMPE and equivalently correction vectors in SPLICE). The analysis and comparison of the two algorithms provide useful insight into the strong success of fMPE and point to further algorithm improvement and extension.**

***Index Terms*—Discriminative training, feature compensation, feature extraction, hidden Markov model, minimum classification error, minimum phone error, piecewise linear mapping, posterior probability, speech processing.**

I. Introduction

RECENT significant progress in large vocabulary conversational speech recognition is largely attributed to a novel feature extraction technique based on the use of posterior probabilities as intermediate "features" in conjunction with discriminative training [8]. The technique, called feature-space minimum phone error (fMPE), however, has not been well understood in terms of its inner workings and of its surprisingly good performance. For example, the new posterior-based features in fMPE are added to the traditional perceptual linear prediction (PLP) cepstral-based features, instead of appending to them, as commonly done with the use of posterior-based features (e.g., [16]) and other types of features such as delta/acceleration parameters (e.g., [9]). On the other hand, a different class of successful speech feature extraction techniques has been designed for the purpose of compensating for acoustic environment distortions of the speech signal. This technique, called stereo-based piecewise linear compensation for environments (SPLICE) [1], [3], [13] has been motivated by the goal of robust speech recognition against noise, a very different one from fMPE, which is aimed at conversational speech recognition with minor acoustic environment distortions. The purpose of this letter is to analyze fMPE and SPLICE algorithms in the same light of mathematical construction and to establish their equivalence in the underlying algorithmic operation. It is our hope that this analysis and comparison can not only provide a better understanding of both algorithms—fMPE in particular—but can also serve to point to further algorithm improvement and extension.

Manuscript received December 6, 2004; revised January 19, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Israel Cohen.

L. Deng, J. Droppo, and A. Acero are with the Microsoft Research, Microsoft Corporation, Redmond, WA 98052 USA (e-mail: deng@microsoft.com).

J. Wu is with the Speech Platform Group, Microsoft Corporation, Redmond, WA 98052 USA.

Digital Object Identifier 10.1109/LSP.2005.847861

II. Analysis of fMPE Algorithm

The feature extraction algorithm fMPE recently published in [8] can be succinctly described by the following computation in run time:

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{M}\mathbf{h}_t \tag{1}$$

where $\mathbf{x}_t \in \Re^p$ is the original low-dimensional feature vector (dimension $p \times 1$) at time frame t, $\mathbf{y}_t \in \Re^p$ is the new feature vector extracted by the algorithm, $\mathbf{h}_t \in \Re^q$ is an intermediate, high-dimensional feature vector ($q \gg p$) whose elements consist of posterior probabilities $p(k|\mathbf{x}_t)$

$$\mathbf{h}_t = (p(1|\mathbf{x}_t), p(2|\mathbf{x}_t), \ldots, p(q|\mathbf{x}_t))' \tag{2}$$

and $\mathbf{M} \in \Re^{p\times q}$ is a transformation matrix that projects the high-dimensional vector $\mathbf{h}_t$ into the subspace of dimension p. In the highly successful implementation of fMPE algorithm reported in [8], the dimensionalities above are set to be $p = 39$, $q \approx 700\,000$. Also, matrix $\mathbf{M}$ is trained via minimizing the discriminative objective function known as minimum phone error (MPE) [7] by gradient descent. This training is embedded in iteration, where each updated fMPE feature set is used to retrain hidden Markov model (HMM) parameters via maximum likelihood. The calculation of the gradient takes into account the change of the HMM parameters after such retraining.

We now analyze the above fMPE algorithm by decomposing the second term in (1) into a large number of individual components.

A. Decomposition Scheme 1

The first scheme of decomposition is to block the long, rectangle matrix $\mathbf{M} \in \Re^{p\times q}$ into many smaller, square matrices $\mathbf{M}^{(i)} \in \Re^{p\times p}$, $i = 1, 2, \ldots, n = q/p$, and block the long vector $\mathbf{h}_t$ into subvectors $\mathbf{h}_t^{(i)} \in \Re^p$, $i = 1, 2, \ldots, n$. Here, i denotes

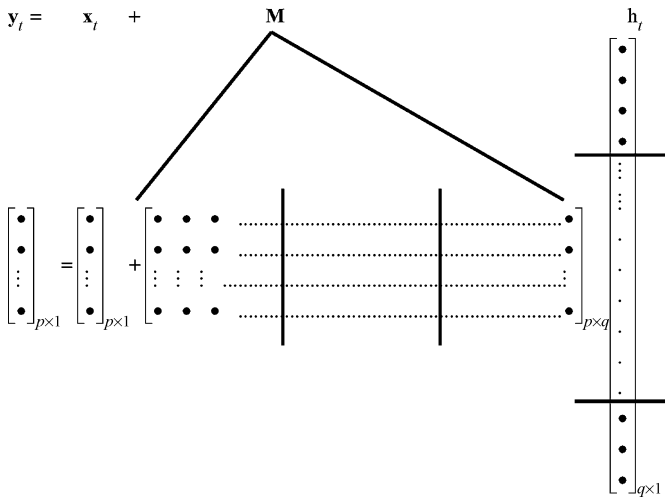

Fig. 1. Decomposition of the fMPE's projection matrix.

the block index, and n is the total number of blocks when q is an integer number of p. The blocking of these matrix and vector can be illustrated in Fig. 1.

This then gives the decomposed form of (1)

$$\mathbf{y}_t = \mathbf{x}_t + \sum_{i=1}^{n} \mathbf{M}^{(i)} \mathbf{h}_t^{(i)}. \tag{3}$$

Equation (3) offers the following interpretation of fMPE: Compensation of the original feature $\mathbf{x}_t$ is carried out by adding a large number of bias vectors, each of which is computed as a full-rank rotation of a small set of posterior probabilities. This contrasts the original interpretation of (1) as a projection of a very high-dimensional posterior-probability vector into a subspace with a much smaller dimension. Under this original interpretation, numerical difficulties would prevent maximum-likelihood estimation of matrix $\mathbf{M} \in \Re^{p \times q}$ (due to its nonsquare nature where $q \gg p$). Under the interpretation expressed in (3), approximations can be easily made to remove the numerical difficulties in carrying out maximum-likelihood estimation. One straightforward approximation is

$$\mathbf{y}_t = \mathbf{x}_t + \sum_{i=1}^{n} \mathbf{M}^{(i)} \mathbf{h}_t^{(i)} \approx \mathbf{x}_t + \mathbf{M}^{(i^*)} \mathbf{h}_t^{(i^*)} \tag{4}$$

where i^* denotes the term on the right-hand side of (3) that is greater than all the remaining $(n-1)$ terms. In fact, the approximation of (4) and the related maximum-likelihood estimation has been successfully implemented in the study of [12] and [14], giving noticeable performance improvement in noise-robust speech recognition.

B. Decomposition Scheme 2

In analyzing the fMPE algorithm in (1), we have developed a second decomposition scheme for the term $\mathbf{M}\mathbf{h}_t$ in (1). In this scheme, matrix $\mathbf{M}$ is decomposed into a total of q vectors column-wise: $\mathbf{m}_k$, $k = 1, 2, \ldots, q$, (i.e., each column vector $\mathbf{m}_k \in \Re^p$ constitutes a "block"). Then, (1) can be rewritten as

$$\begin{aligned} \mathbf{y}_t &= \mathbf{x}_t + \mathbf{M}\mathbf{h}_t = \mathbf{x}_t + [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_q][h_1, \ldots, h_q]' \\ &= \mathbf{x}_t + \sum_{k=1}^{q} h_k \mathbf{m}_k = \mathbf{x}_t + \sum_{k=1}^{q} p(k|\mathbf{x}_t)\mathbf{m}_k. \end{aligned} \tag{5}$$

An interpretation of the decomposed form of (5) is as follows: The final extracted feature by fMPE is the original (PLP) cepstral feature compensated by a frame-dependent bias vector. The compensation vector consists of a linear weighted sum of a set of frame-independent correction vectors, where the weight is the posterior probability associated with the corresponding correction vector. It is worth noting that the bias-compensation interpretation of fMPE above bears some resemblance to the feature-space stochastic matching approach developed for environment-robust speech recognition [10], [11]. The key difference is, however, that the bias vector for compensation in fMPE is specific to each time frame t, whereas the bias vector in feature-space stochastic matching is common over all frames in the utterance. Such frame dependence is due to the posterior weight $p(k|\mathbf{x}_t)$ in (5), which is specific to each frame. Additional differences are retraining of the HMM parameters after feature compensation and discriminative training of the base bias vectors instead of maximum-likelihood training.

Importantly, the decomposed form of (5) for the fMPE algorithm in run-time operation also establishes its equivalence to the SPLICE algorithm, which is constructed by a formal optimization principle and which we review and analyze now.

III. Analysis of SPLICE Algorithm

The original version of SPLICE was developed for solving the noise robustness problem for speech recognition and was published in [1] and [2]. SPLICE assumes that the noisy speech cepstral vector $\mathbf{x}_t \in \Re^p$ is distributed according to a mixture of q Gaussians. (This is analogous to the assumption in fMPE that the original cepstral feature set $\mathbf{x}_t \in \Re^p$ are used to create q Gaussian clusters as the basis for generating q-dimensional posteriors as intermediate "features" subject to further projection [8].) These Gaussians, called "codebook" in the SPLICE literature, partition the acoustic space in terms of noisy speech $\mathbf{x}$, and the parameters of the Gaussians are determined by performing VQ followed by training each of the means and variances in the Gaussians using the training vectors classified into the corresponding VQ codewords.[1]

The SPLICE algorithm further assumes that the true, unobserved clean speech feature vector $\overline{\mathbf{y}}_t$ and its corresponding noisy speech counterpart $\mathbf{x}_t$ are "piecewise linearly"[2] related according to

$$\overline{\mathbf{y}}_t = \mathbf{x}_t + \mathbf{r}(\mathbf{x}_t) \approx \mathbf{x}_t + \mathbf{r}_i(\mathbf{x}_t) \tag{6}$$

where $\mathrm{i}(\mathbf{x})$ is an index to the "correction vector" $\mathbf{r}$ of the mixture component to which $\mathbf{x}_t$ belongs. That is, the index determines

[1]In this paper, we use $\mathbf{x}$ to represent the noisy speech's cepstral features (analogous to the original PLP-cepstral features before applying fMPE) and $\mathbf{y}$ to represent the enhanced speech features (analogous to the final fMPE-compensated features). In our original SPLICE publications [1], [2], $\mathbf{x}$ was used to represent clean speech features and $\mathbf{y}$ noisy speech features.

[2]The "slope" (or rotation) in the linear relationship is assumed to be unity (i.e., rotation by identity matrix) in the SPLICE implementation for saving parameters. Only the "intercept" (or bias) is used here.

which "piece" of the local linear approximation is used for the "piecewise" linear approximation to the nonlinear relationship between the noise and clean speech feature vectors.

Given these assumptions, the SPLICE feature compensation algorithm can be rigorously derived using the minimum mean square error (MMSE) rule as follows:

$$\hat{\mathbf{y}}_t = \int_{\mathbf{y}_t} \mathbf{y}_t p(\mathbf{y}_t|\mathbf{x}_t) d\mathbf{y}_t \approx \int_{\mathbf{y}_t} (\mathbf{x}_t + \mathbf{r}_i) p(\mathbf{y}_t|\mathbf{x}_t) d\mathbf{y}_t$$
$$= \mathbf{x}_t + \int_{\mathbf{y}_t} \mathbf{r}_{\mathbf{i}(\mathbf{x})} p(\mathbf{y}_t|\mathbf{x}_t) d\mathbf{y}_t = \mathbf{x}_t + \int_{\mathbf{y}_t} \sum_{i=1}^{q} \mathbf{r}_i p(\mathbf{y}_t, i|\mathbf{x}_t) d\mathbf{y}_t$$
$$= \mathbf{x}_t + \sum_{i=1}^{q} \mathbf{r}_i \int_{\mathbf{y}_t} p(\mathbf{y}_t, i|\mathbf{x}_t) d\mathbf{y}_\mathbf{t} = \mathbf{x}_t + \sum_{i=1}^{q} p(i|\mathbf{x}_t) \mathbf{r}_i. \quad (7)$$

The posterior probabilities in (7) in the SPLICE algorithm of [1] and [2] were computed from Bayes rule using the clustered parameters in the mixture of Gaussians for $\mathbf{x}$, in the same way as the posterior features are computed in fMPE reported in [8] (both used identical mixture weights or no prior for Gaussians). Hence, *the run-time operation of the SPLICE algorithm in (7) is exactly the same as that of the fMPE algorithm in (5)*. Here, the correction vectors $\mathbf{r}_i$ in SPLICE are equivalent to the column vector $\mathbf{m}_i$ in the fMPE projection matrix $\mathbf{M}$.

The fMPE algorithm, as presented in [8], is empirical; in particular, the addition (instead of appending) of the transformed high-dimensional posterior features to the original cepstral features was justified in terms of a way of reasonable initialization for the fMPE parameter training. In contrast, the above analysis on the SPLICE algorithm demonstrates that the operation of the addition is a natural consequence of the rigorous MMSE optimization rule. Since (7) is derived based on the assumption that noisy and clean speech features are piecewise linearly related, the practical success of the fMPE algorithm on conversational speech, as reported in [8], suggests that the relationship between the highly effective fMPE features (after compensation) and the less effective (PLP) cepstral features (before compensation) may also be adequately described by piecewise linearity for conversational speech.

IV. Comparisons of fMPE and SPLICE Algorithms

A. Algorithm Comparison

The equivalence of fMPE and SPLICE algorithms in run time was established above, both drawing on the fact that feature extraction and compensation are accomplished by adding a posterior-based weighted sum of correction vectors to the original features.[3] The main difference between the two algorithms lies in the ways of training these correction vectors.

[3]It is noted that the VTS algorithm originally proposed in [6] for environment-robust speech recognition also uses posterior-based sum of linear corrections. Two main differences from SPLICE/fMPE are as follows: 1) the VTS algorithm works in the domain of log filter bank energy instead of cepstrum, and 2) the "codebook" (i.e., mean vectors of Gaussians for posterior computation) in the VTS algorithm is created from clean speech, and the "codebook" in SPLICE/fMPE is created from distorted speech. For conversational speech for which fMPE was developed, it is impossible to create the codebook from idealized, undistorted speech (analogous to clean speech, as in the VTS algorithm). Also, in [2], it is shown experimentally that the use of distorted speech to create the "codebook" (consistent with fMPE) performs much better than the use of undistorted speech for noise-robust speech recognition.

TABLE I
WER for (Left) SPLICE Algorithm and for (Right) fMPE Algorithm Evaluated in Aurora2 Task and in DARPA-EARS Rich-Transcription-2004 Conversational Telephone Speech-Recognition Task, Respectively. Latter WER Numbers Are Extracted From Fig. (1b) in [8] for the Speaker Adapted System. WER Results With Similar (Analogous) Techniques Are Listed on the Same Lines. Note While the Evaluation Tasks for SPLICE and fMPE Are Different, Similar Relative WER Reductions Are Apparent With Analogous Techniques in Both Algorithms

SPLICE-Related Performance (WER on Aurora2 task)		**fMPE-Related Performance** (WER on Conversational Telephone Speech)	
Baseline (Cepstral features; no SPLICE; ML-trained HMMs)	13.0%	22.0%	Baseline line (Cepstral features, no fMPE, ML-trained HMMs)
ML training of correction vectors	11.8%	--	ML training of projector matrix
Discriminative training of correction vectors (MCE)	10.9%	20.2%	Discriminative training of projector matrix (MPE) imbedded in ML training of HMM parameters
MCE training of HMM parameters alone	10.0%	20.9%	MPE training of HMM parameters alone
Joint MCE on correction vectors & HMM parameters	9.2%	19.2%	MPE training of projector matrix followed by MPE training of HMM parameters

In the early versions of SPLICE [1], [2], the maximum-likelihood estimate was derived and effectively used assuming the availability of stereo training data (i.e., simultaneous clean and distorted speech). This requirement was removed in two extended versions of the SPLICE algorithm—one with new maximum-likelihood training [12] and another with discriminative training based on the criterion of minimum classification error (MCE) [13], [15]. The discriminative version of SPLICE becomes close to the fMPE implementation in [8], where the objective function in the training is MPE instead of MCE. Both require no expensive "stereo" data.

Additional differences between the fMPE and SPLICE algorithms are the slightly different ways of using frame context expansion in determining the posterior weights and of performing the estimate's smoothing (see [4], [5], and [12] for details of SPLICE in these two aspects of implementation). Also, the size of the Gaussian codebook used in SPLICE is much smaller than that in fMPE, because the amount of training data available in the benchmark evaluation task (Aurora) for SPLICE is much smaller than that in the DARPA/EARS conversation telephone speech-to-text task for fMPE. Finally, scheduling of iterative gradient-descent-based feature updates and HMM parameter updates in fMPE [8] is slightly different from that in the SPLICE with MCE training (which is also gradient-descent based [12], [13]).

B. Performance Comparison

Here, we provide some speech-recognition performance comparison between the fMPE and SPLICE algorithms (which are formally equivalent in run rime but differ in training, as analyzed above). The left side of Table I shows WERs for the various versions of the SPLICE algorithm evaluated on the Aurora2 task (noisy connected-digit recognition [4], [13]), and the right side of Table I shows WERs for the fMPE algorithm constructed in various ways evaluated on the DARPA-EARS rich-transcription-2004 evaluation task for conversational

telephone speech [data extracted from Fig. (1b) in [8] for the speaker adapted system]. In arranging these WER results, similar techniques in training are listed on the same lines. Since the rich transcription task is much more difficult than the Aurora2 task, the WER is substantially higher. However, relative WER reduction after applying the fMPE and SPLICE algorithms in several ways is generally consistent. One exception is that discriminative training applied to HMM parameters alone gives lower WER than applied to features (correction vectors) alone for SPLICE (from 10.9% to 10.0%), but the opposite holds for fMPE (20.2% versus 20.9%).

Note in Table I that ML training of the correction vectors in SPLICE gives sizable WER reduction (from 13.0% to 11.8%). Such training was made possible due to the analogous decomposition Scheme-I in SPLICE and to the related approximation, as discussed in Section II-A. No ML results are available for fMPE. Indeed, without decomposition and approximation, the computation for rigorous ML training would be prohibitive—matrix inversion of $\mathrm{q} \times \mathrm{q} = 700\,000 \times 700\,000$ in size would be required.

V. Summary and Conclusion

Two feature extraction and compensation algorithms, fMPE and SPLICE, are analyzed and compared. While SPLICE is motivated by noise-robust speech recognition, where the main difficulty is acoustic environment variations, fMPE is motivated by conversational speech recognition, where the main difficulty is speaking style variations. Feature extraction in fMPE is formally formulated as a projection of large-dimensional posterior-based intermediate "features" into a small subspace with the same dimensionality as the commonly used cepstral features (e.g., 39), where the result of projection is treated as a bias vector. This algorithm construction was justified as "a reasonable starting point for training" [8]. We analyzed such algorithm construction via matrix/vector decomposition and concluded that it is equivalent to the run-time algorithm operation in SPLICE, where feature compensation is accomplished by adding a posterior-based weighted sum of "correction vectors" (equivalent to the column vectors in the fMPE projection matrix) to the uncompensated features. Since SPLICE is developed using the rigorous MMSE optimization principle, the same principle applies to fMPE as well. In addition to the above analysis, various aspects of training in fMPE and SPLICE, as well as their performances in two speech-recognition tasks, are compared, and their differences are noted. It is hoped that this analysis and comparison can not only provide a better understanding of the inner workings of fMPE responsible for its apparent success but can also serve to point to further algorithm improvement and extension. One simple example of such extension is to generalize the bias-only compensation to include rotation compensation as well. The mathematical underpinning of this more general form of compensation was briefly discussed in [1] in the context of SPLICE and noise robustness.

References

[1] L. Deng, A. Acero, M. Plumpe, and X. D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. Int. Conf. Spoken Lang. Process.*, vol. III, Oct. 2000, pp. 806–809.

[2] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. D. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. I, Salt Lake City, UT, Apr. 2001, pp. 301–304.

[3] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 568–580, Nov. 2003.

[4] J. Droppo, L. Deng, and A. Acero, "Evaluation of the SPLICE algorithm on the Aurora2 database," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. 1, Aalborg, Denmark, Sep. 2001, pp. 217–220.

[5] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. I, Orlando, FL, May 2002, pp. 57–60.

[6] P. Moreno, B. Raj, and R. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, 1996, pp. 733–736.

[7] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. Acoust,, Speech, Signal Process.*, vol. I, Orlando, FL, May 2002, pp. 105–108.

[8] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. DARPA EARS RT-04 Workshop*, Palisades, NY, 2004, Paper no. 35.

[9] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[10] M. Rahim, B.-H. Juang, W. Chou, and E. Buhrke, "Signal conditioning techniques for robust speech recognition," *IEEE Signal Process. Lett.*, vol. 3, no. 4, pp. 107–109, Apr. 1996.

[11] A. Sankar and C.-H. Lee, "A maximum-likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 3, pp. 190–202, May 1996.

[12] J. Wu, "Discriminative speaker adaptation and environmental robustness in automatic speech recognition," Ph.D. dissertation, Univ. of Hong Kong, 2004.

[13] J. Wu and Q. Huo, "An environment compensated minimum classification error training approach and its evaluation on Aurora2 database," in *Proc. Int. Conf. Spoken Lang. Process.*, Denver, CO, pp. 453–456.

[14] ——, "An environment compensated maximum likelihood training approach based on stochastic vector mapping," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, Philadelphia, PA, Mar. 2005, pp. 429–432.

[15] ——, "Several HKU approaches to robust speech recognition and their evaluation on Aurora connected digit recognition tasks," in *Proc. Eur. Conf. Speech Commun. Technol.*, vol. I, Geneva, Switzerland, Sep. 2003, pp. 21–24.

[16] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Proc. Int. Conf. Spoken Lang. Process.*, Jeju Island, Korea, Oct. 2004, pp. 554–557.