Multi-dialect acoustic modeling using phone mapping and online i-vectors

Harish Arsikere *Ashtosh Sapru* *Sri Garimella*

Alexa Machine Learning, Amazon, Bangalore, India
{arsikere, sapru, srigar}@amazon.com

Abstract

This paper proposes a simple phone mapping approach to multi-dialect acoustic modeling. In contrast to the widely used shared hidden layer (SHL) training approach (hidden layers are shared across dialects whereas output layers are kept separate), phone mapping simplifies model training and maintenance by allowing all the network parameters to be shared; it also simplifies online adaptation via HMM-based i-vectors by allowing the same T-matrix to be used for all the dialects. Using the LSTM-HMM framework, we compare phone mapping with transfer learning and SHL training, and we also compare the efficacy of online i-vectors with that of one-hot dialect encoding. Experiments with a 2K hour dataset comprising four English dialects show that (1) phone mapping yields significant WER reductions over dialect-specific training (14%, on average) and transfer learning (5%, on average); (2) SHL training is only slightly better than phone mapping; and (3) i-vectors provide useful additional reductions (3%, on average) while one-hot encoding has little effect. Even with a large 40K hour dataset (comprising the same four English dialects) and fully optimized sequence discriminative training, we show that phone mapping provides healthy WER reductions over dialect-specific models (10%, on average).

Index Terms: multi-dialect acoustic modeling, phone mapping, i-vectors, shared hidden layers, one-hot encoding

1. Introduction

Deep neural networks (DNNs) have enabled a rapid development of multi-dialect and multilingual acoustic modeling (AM) techniques—most of which are derived from the general philosophy of shared representation learning and knowledge transfer. As several studies have shown, unified AM training can be used to improve automatic speech recognition (ASR) performance in low-resource settings [1, 2, 3, 4, 5], and to deploy robust models in environments where a mix of dialects is expected [6, 7]. Broadly speaking, unified AM training approaches rely on one or more of the following: universal phone sets that cover all languages or dialects of interest [8, 9]; training strategies such as multitask learning [10, 1, 4, 5], ensemble learning [7] and transfer learning or adaptation [6, 11, 2]; and auxiliary input features such as session-level i-vectors [12] and one-hot language or dialect vectors [3, 13, 14].

This paper investigates multi-dialect acoustic modeling using long short-term memory (LSTM) networks; most previous studies, barring a few exceptions [5, 10, 4, 13], have used feed-forward DNNs. Our investigation has two objectives: (1) pooling data such that ASR performance is enhanced for all the dialects and not just the under-resourced ones, and (2) simplifying model training and maintenance (e.g., for deployment purposes) by allowing all the network parameters to be shared. Previous studies have observed that a multi-dialect model trained from scratch by simply pooling data from all the dialects tends to perform consistently worse than dialect-specific models [13, 14]. In this paper, we propose a phone mapping technique that allows data to be pooled easily while achieving the desired objectives. Using a multi-dialect dataset of 40K hours, we show that phone mapping consistently outperforms dialect-specific models, even after sequence discriminative training.

To train an AM whose parameters are fully shared across dialects, the conventional *from scratch* approach (e.g. as in [13]) is to unify the dialect-specific phone sets, estimate the acoustic decision tree and flat-start alignments by pooling all the data, and learn to classify the tied context dependent hidden Markov model (HMM) states (senones) as defined by the acoustic tree. Although this approach is fully data driven, the resulting multi-dialect model could be suboptimal in performance—as also observed by previous studies. This could be attributed, in part, to the *mixed* nature of the acoustic tree which has to model cross-dialect variations along with intra-dialect triphone contexts. The proposed phone mapping approach, in contrast, assumes one of the dialects (and its acoustic tree) to be *canonical* and maps the phone sets of all other dialects to the canonical phone set. Once the canonical dialect is identified, the proposed framework also allows new dialects to be incorporated relatively easily as compared to the conventional approach.

The shared hidden layer (SHL) approach is a popular alternative approach to multi-dialect AM training, where the hidden layers of the model are shared by all the dialects while the output softmax layers are kept separate [1, 4, 5]. While SHL training allows the individual phone sets and acoustic trees to differ from one another, it incurs an increase in training time owing to multitask loss minimization (e.g., the authors of [1] report that SHL training with eleven output layers requires twice as much time as isolated training). Also, unlike phone mapping, this approach, by design, does not allow the network parameters to be fully shared by all the dialects.

The efficacy of phone mapping and SHL training could potentially be enhanced by using auxiliary input features such as one-hot dialect embeddings and i-vectors. Since one-hot encoding has been shown to be effective in several studies [3, 13, 14], we assess its performance in the context of phone mapping and SHL training. In addition, we study the benefit of using HMM-based frame-level i-vectors for online speaker and accent adaptation. Frame-level i-vectors (as implemented in [15]) are well suited to online adaptation in digital voice assistants, e.g. Amazon Echo or Google Home, where (1) the speaker interacting with the device can change often and (2) i-vector extraction cannot wait until the utterance has been fully observed. Chen *et al.* used i-vectors for multi-dialect training, but the i-vectors were extracted in an offline fashion [12]. It is worth noting that phone mapping simplifies adaptation via HMM-based online i-vectors by allowing the same T-matrix to be used for all the dialects.

The rest of this paper describes the techniques used and the experiments conducted. It is assumed in all cases that an LSTM or DNN AM is available for generating training targets (i.e. flat-start training is not involved).

2. Methods

This section begins with a discussion of transfer learning, which, although not a unified AM training technique per se, serves as a competitive baseline to assess the benefit of training all locales together as compared to sharing knowledge between just two locales. The remainder of this section describes the unified AM training strategies and auxiliary inputs explored in this paper.

2.1. Transfer learning

Transfer learning implementations can differ depending on how the output and hidden layers of the seed model (the model whose knowledge is transferred to the target locale) are reused. In this study transfer learning from locale *X* to locale *Y* is implemented as follows.

1. Obtain targets for training data of locale *Y* using an *alignment* model (model used for generating forced alignments).
2. Remove the output layer of the seed model (which is trained on data from locale *X* using cross-entropy loss) and initialize a new output layer with random weights corresponding to the target size of locale *Y*.
3. Train the output layer for two epochs using cross-entropy loss while keeping the hidden layers fixed; then run cross-entropy training for the entire network.

Since transfer learning does not impose any constraints on the tree of the target locale, one can use it as a quick and effective knowledge sharing technique when a well trained seed model is available.

2.2. Phone mapping

The key idea behind phone mapping is to treat the phone set of one of the locales as a canonical representation, and map the phone sets of all other locales to it. For a set of locales $\{L_1, L_2, \ldots, L_n\}$, unified AM training via phone mapping is implemented as follows.

1. Choose a canonical phone set such that it has the maximum amount of overlap with the remaining phone sets. Let us treat L_1's phone set to be canonical.
2. Obtain training targets for L_1 using its alignment model.
3. **For each of the locales** $L_2, L_3, \ldots, L_n$:
   • Obtain pronunciations for all the training tokens by looking up a background lexicon whose entries are in the locale's native phone set; if a word is not present in the lexicon, obtain its pronunciation using a grapheme-to-phoneme (g2p) model which is trained on the locale's native lexicon and phone set. For each native pronunciation, map all non-canonical phones to their *nearest* phones in the canonical set—these mappings require linguistic inputs, but that is a minor overhead since it needs to happen only once for a fixed canonical phone set.
   • Obtain forced alignments for training data using the alignment model from L_1.
   • Train an AM (pretraining followed by cross-entropy training) using targets produced by the L_1 alignment model.
   • Regenerate training targets using the AM trained above.
4. Combine and shuffle the training data prepared for all locales (from steps (2) and (3)), and train a unified AM via pretraining followed by cross-entropy training.

Note that the last two sub-steps of step (3) (training an intermediate AM using targets from the canonical model and regenerating the final training targets from that AM) are optional; they can be omitted if the locale in question has little training data or is *similar* to the canonical locale by virtue of sharing most of its phones with the canonical phone set.

2.3. Shared hidden layer (SHL) training

By allowing locales to share the hidden layers of the network while maintaining separate output (classification) layers, SHL training provides an effective cross-lingual knowledge transfer mechanism without imposing any constraints on the phone sets and acoustic decision trees, and thereby the tied continuous-density hidden Markov model (HMM) states or senones, of the locales being unified. In this framework the training targets for each locale are obtained using a locale-specific alignment model. After combining and shuffling the datasets from all locales, the unified AM is built via pretraining followed by cross-entropy training.

SHL training can be viewed as multitask learning except that not all network parameters are affected by every training sample; the algorithm works such that for a given training sample belonging to a particular locale, only the corresponding output layer and the shared hidden layers are updated while all the other output layers remain fixed. Since SHL training tends to be slow owing to multitask loss minimization (especially as the number of jointly-trained locales increases), one way to speed up training is to incorporate phone mapping into the SHL framework—have a common output layer for all accents of a given language while maintaining separate output layers across languages.

2.4. One-hot dialect vectors

Language-adaptive training through one-hot auxiliary input vectors has been explored by Müller and Waibel [3]; they concatenated one-hot language vectors with bottleneck features to train a unified DNN AM. One-hot locale vectors, which are constant for all training samples of a given locale, translate essentially to locale-specific bias values. Conceptually, one-hot locale vectors could be supplied to not only the input layer but also some or all of the hidden layers; in this paper they are used at the input layer as well as all the hidden layers of the unified AM (trained via phone mapping or the SHL approach). While it is possible that one-hot vectors provide better results when used selectively (e.g., only at the first or last hidden layer), such detailed investigations are not part of this paper.

2.5. Frame-level online i-vectors

Using i-vectors as auxiliary inputs is a popular approach to adaptive AM training [16, 17]. While the entire target utterance (the utterance to be decoded) can be used for i-vector estimation in offline decoding scenarios, real-time decoding (e.g., for digital voice assistants) requires *on-the-fly* computation. As shown in [15], decoder one-best alignments and AM senone posteriors can be used to accumulate and update sufficient statistics such that i-vector estimation happens on a frame-by-frame basis and in a causal manner.

Let $\{\mu_i, \mathbf{\Sigma}_i\}_{i=1}^{M}$ denote the means and covariances of the Gaussians that correspond to the M senones of an acoustic decision tree, and let $\mathbf{T} = \left[\mathbf{T}_1; \mathbf{T}_2; \ldots; \mathbf{T}_M\right]$ denote the total-variability matrix (trained *a priori*). Assuming that l frames of an incoming target utterance have been observed, let $[\mathbf{x}_1, \ldots, \mathbf{x}_l]$ denote the sequence of feature vectors. The i-vector estimated at frame l, denoted by $\mathbf{u}_l$, is then given by Eq. (1):

$$\mathbf{u}_l = \left[\mathbf{I} + \mathbf{S}_0\left(l\right)\right]^{-1} \mathbf{S}_1\left(l\right), \tag{1}$$

where the partial online statistics $\mathbf{S}_0\left(l\right)$ and $\mathbf{S}_1\left(l\right)$ are given by:

$$\begin{aligned} \mathbf{S}_0\left(l\right) &= \mathbf{S}_0\left(0\right) e^{-\tau l} + \sum_{i=1}^{M} \gamma_i\left(l\right) \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{T}_i \\ \mathbf{S}_1\left(l\right) &= \mathbf{S}_1\left(0\right) e^{-\tau l} + \sum_{i=1}^{M} \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{f}_i\left(l\right). \end{aligned} \tag{2}$$

In Eq. (2), $\mathbf{S}_0\left(0\right)$ and $\mathbf{S}_1\left(0\right)$ denote the sufficient statistics accumulated from history (i.e. all previously-decoded utterances); τ (> 0) is an exponential decay factor which is used to emphasize the most recent frames; and $\gamma_i\left(l\right)$ and $\mathbf{f}_i\left(l\right)$ $(1 \leq i \leq M)$ are given by Eq. (3):

$$\begin{aligned} \gamma_i\left(l\right) &= \sum_{t=1}^{l} e^{-\tau(l-t)} P_{AM}\left(i|\mathbf{x}_t\right) \\ \mathbf{f}_i\left(l\right) &= \sum_{t=1}^{l} e^{-\tau(l-t)} P_{AM}\left(i|\mathbf{x}_t\right)\left(\mathbf{x}_t - \mu_i\right). \end{aligned} \tag{3}$$

In Eq. (3), $P_{AM}\left(\cdot|\mathbf{x}\right)$, the posteriors from the AM, serve as *soft* assignments between feature vectors and Gaussians. The history statistics, $\mathbf{S}_0\left(0\right)$ and $\mathbf{S}_1\left(0\right)$, are accumulated in a manner similar to partial online statistics, but the assignments between feature vectors and Gaussians are provided by decoder one-best alignments.

In the SHL framework, the $\mathbf{T}$ matrix and the Gaussian parameters must be estimated separately for each locale or group of locales having a distinct output layer, but this is not necessary in the phone mapping approach since the senones are shared by all locales.

3. Experimental Results

This section describes the feature extraction pipeline and AM training setup, followed by an empirical discussion of unified AM training across accents (Sec. 3.1), and accents and languages (Sec. 3.2).

Log filter-bank energies (LFBEs) are used as the primary acoustic features for AM training; they are extracted at 10 ms intervals using 25 ms analysis windows. For each frame, the power spectrum is integrated using 64 Mel-warped filters, and the resulting filter-bank outputs are transformed using natural log to obtain LFBEs; they are also normalized by subtracting the time-varying mean estimate that is obtained through an autoregressive update [18].

For extracting i-vectors, senone Gaussians are assumed to have diagonal covariance matrices. Gaussian and $\mathbf{T}$ matrix parameters are estimated using 40 dimensional features, which are obtained by first stacking 9 LFBE frames (current frame plus a left and right context of 4) and then transforming them through a block diagonal discrete cosine transform (DCT), a linear discriminant analysis (LDA) transform and a maximum likelihood linear transform (MLLT) [19]. The i-vector dimension is set to 100, and $\mathbf{T}$ matrix estimation is done via the expectation-maximization (EM) algorithm [20]. The exponential decay factor τ (see Eqs. (2) and (3)) is set to 0.002; this corresponds to an effective time window of 500 frames or 5 seconds.

Table 1: *Train and test data distributions of English accents for experiments discussed in Section 3.1.*

dialect	train data	test data
en-US (American)	1100	37
en-GB (British)	500	21
en-IN (Indian)	300	27
en-AU (Australian)	100	14

All the AMs trained in this paper are 5 hidden-layer LSTM networks. Features are input with an 8-frame delay, i.e. features at time t + 80 ms are used to predict targets at time t. Prior to AM training, LFBEs and i-vectors are mean and variance normalized using training data statistics. One-hot locale vectors, if used, are supplied to the input layer as well as all the LSTM hidden layers.

Well-trained seed models are used to initialize transfer learning, but for all other experiments, networks are pretrained using 30 hours of training data. A learning rate of 8e−4 is used for pretraining and also for the initial output layer training of transfer-learned models. In pretraining, LSTM hidden layers are trained jointly (not layer wise), and auxiliary senone classification layers with small task weights are connected to all the non-final hidden layers (the final hidden layer is connected to the primary senone classification output layer(s)).

Cross-entropy training happens in two phases: a *chunking* phase where the LSTM hidden states are maintained for 32-frame chunks, and a *finetuning* phase where the hidden states are propagated until utterance-final frames. New-bob learning rate scheduling is applied to both phases by monitoring frame classification accuracy on a one-hour held out set; the learning rate decay factor is set to 0.5. In the chunking phase, the initial learning rate is 8e−4 and the minibatch size is 2048; in the finetuning phase, the initial learning rate is 5e−5 and the minibatch size is 20480.

Sequence discriminative training is done using the boosted maximum mutual information (BMMI) criterion [21]. AMs trained with cross-entropy loss are first used to obtain numerator alignments and boosted denominator lattices (the acoustic scaling factor and the lattice boosting factor are set to 0.08 and 0.1, respectively), after which they are BMMI trained for two epochs using a learning rate of 2e−6.

All acoustic models are trained using distributed, synchronous, stochastic gradient descent; to decode them, locale-specific language models are borrowed from our in-house ASR systems.

3.1. Experiments with a 2K hour dataset

The following English accents are considered in the first set of unified AM training experiments: American (en-US), British (en-GB), Indian (en-IN) and Australian (en-AU); the amount of train and test data used per accent is shown in Table 1. For preparing experimental datasets, utterances are randomly sampled from collections obtained through far-field device units that vary in their microphone characteristics; the train and test datasets do not have any device units or speakers in common. For the phone mapping approach, British English is treated as the canonical accent. For transfer learning, the AM trained using American English data (1100 hours) is used as the seed model. LDA and MLLT transforms, $\mathbf{T}$ matrices and

Table 2: *Relative WERRs (%), with respect to mono-dialect training, provided by phone mapping, SHL training and the use of auxiliary inputs (all results are at the cross-entropy stage).*

method	US	GB	IN	AU
mono-dialect training	0.0	0.0	0.0	0.0
transfer learning	-	4.2	8.6	22.8
phone mapping	8.3	14.4	8.4	27.4
+ one-hot vectors	8.7	11.8	10.9	29.3
+ i-vectors	12.0	16.5	11.2	31.3
+ one-hot & i-vectors	13.3	15.9	12.2	32.0
SHL training	7.4	14.1	10.9	31.3
+ one-hot vectors	7.6	14.3	11.9	29.7
+ i-vectors	11.3	18.4	12.9	34.4
+ one-hot & i-vectors	11.3	17.1	12.8	32.5

Table 3: *Relative WERRs (%), with respect to mono-dialect training, provided by phone mapping (40K hour dataset).*

method	US	GB	IN	AU
mono-dialect training	0.0	0.0	0.0	0.0
+ sMBR training	9.6	12.3	7.7	8.3
phone mapping	3.1	3.9	4.5	18.1
+ sMBR training	12.7	17.2	14.5	30.9

Gaussian models (for i-vector estimation) are borrowed from in-house systems.

Table 2 shows the relative word error rate reductions (WERRs) achieved (with respect to mono-dialect baselines) via unified training and the use of auxiliary inputs; note that the results are obtained using models trained with cross entropy loss. The below observations can be made from Table 2.

- Transfer learning provides healthy gains over the baseline but the unified training approaches (phone mapping as well as SHL training) offer larger improvements in general. This confirms that knowledge sharing across more than two locales is beneficial.
- The gains due to unified training are larger for accents with smaller amounts of training data (en-GB, en-IN and en-AU), as expected. Interestingly, the gains for en-US, which contributes to more than half of the total training data, are also significant. The gains for en-IN are somewhat lower than for en-GB and en-AU, suggesting that Indian English is possibly quite different from the other accents.
- SHL training offers better gains than phone mapping, on average. While the difference between the two approaches in not large, SHL training seems to be more consistent (e.g. phone mapping is slightly worse than transfer learning for en-IN).
- Frame-level i-vectors provide useful gains with phone mapping as well as SHL training. However, one-hot vectors appear to have little effect on performance—both in isolation and in combination with i-vectors. While our findings contradict the observations made in [3], it is possible that one-hot vectors provide better results when used selectively (e.g., only at the first or last hidden layer).

3.2. Experiments with a 40K hour dataset

Table 3 compares phone mapping with mono-dialect training using a large 40K training hour corpus; the WERRs are with respect to the mono-dialect cross-entropy baselines. The 40K hour dataset comprises the same four English dialects as in Section 3.1 and the dialect distribution is similar to that captured by Table 1. The test datasets are the same as in the 2K hour setup. The training data for each dialect is composed of the original recordings plus their corrupted copies obtained via simulated reverberation. Note that the corrupted copies use the same training targets as the original recordings.

It is clear from Table 3 that even with large amounts of training data, phone mapping provides healthy gains over dialect-specific training. More importantly, the gains continue to hold after sequence discriminative training. It is interesting to note that the gains after sMBR training (row 4 vs. 2) are in fact larger compared to the gains after cross-entropy training (row 3 vs. 1). This suggests that the multi-dialect model is somewhat *coarse* at the cross-entropy stage and that correcting the phone confusions via sMBR training significantly improves ASR performance.

4. Conclusions

Three main types of knowledge sharing (among locales) were evaluated in this paper: transfer learning, phone mapping and SHL training. The effect of using one-hot locale vectors and frame-level online i-vectors was also studied.

Experiments with multi-accent training show that transfer learning is a quick and effective approach which one can use when a well trained seed model is available. Phone mapping and SHL training—the two unified modeling approaches discussed—offer better performance than transfer learning; since SHL training is only slightly better, on average, than phone mapping, the latter approach can be used if training time is critical (especially when dealing with a large number of locales). While frame-level i-vectors provide useful gains with unified training, one-hot vectors, as per our implementation, do not provide much benefit. Using i-vectors with phone mapping is convenient for multi-accent training because the $\mathbf{T}$ matrix and Gaussian models can be shared by all locales. An important observation from our experiments is that the gains provided by unified modeling continue to hold after sequence discriminative training.

5. References

[1] G. Heigold, V. Vanhoucke, A. W. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013, pp. 8619–8623.

[2] T. Alumäe, S. Tsakalidis, and R. Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages," in *INTERSPEECH*, 2016, pp. 3883–3887.

[3] M. Müller and A. Waibel, "Using language adaptive deep neural networks for improved multilingual speech recognition," in *IWSLT*, 2015.

[4] S. Feng and T. Lee, "Improving cross-lingual knowledge transferability using multilingual TDNN-BLSTM with language-dependent pre-final layer," in *INTERSPEECH*, 2018, pp. 2439–2443.

[5] S. Zhou, Y. Zhao, S. Xu, and B. Xu, "Multilingual recurrent neural networks with residual learning for low-resource speech recognition," in *INTERSPEECH*, 2017, pp. 704–708.

[6] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the KLD-regularized model adaptation," in *INTERSPEECH*, 2014, pp. 2977–2981.

[7] M. Elfeky, M. Bastani, X. Velez, P. J. Moreno, and A. Waters, "Towards acoustic model unification across dialects," in *SLT*, 2016, pp. 624–628.

[8] T. Schultz and A. Waibel, “Multilingual and crosslingual speech recognition,” in *DARPA Workshop on Broadcast News Transcription and Understanding*, 1998, pp. 259–262.

[9] H. Lin, L. Deng, D. Yu, Y. Gong, A. Acero, and C.-H. Lee, “A study on multilingual acoustic modeling for large vocabulary ASR,” in *ICASSP*, 2009, pp. 4333–4336.

[10] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, “Joint modeling of accents and acoustics for multi-accent speech recognition,” in *ICASSP*, 2018, pp. 5989–5993.

[11] A. Ghoshal, P. Swietojanski, and S. Renals, “Multilingual training of deep neural networks,” in *ICASSP*, 2013, pp. 7319–7323.

[12] M. Chen, Z. Yang, J. Liang, Y. Li, and W. Liu, “Improving deep neural networks based multi-accent Mandarin speech recognition using i-vectors and accent-specific top layer,” in *INTERSPEECH*, 2015, pp. 3620–3624.

[13] M. Grace, M. Bastani, and E. Weinstein, “Occam’s Adaptation: A Comparison of Interpolation of Bases Adaptation Methods for Multi-Dialect Acoustic Modeling with LSTMS,” in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 174–181.

[14] B. Li, T. N. Sainath, K. C. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, “Multi-dialect speech recognition with a single sequence-to-sequence model,” in *ICASSP*, 2018, pp. 4749–4753.

[15] H. Arsikere and S. Garimella, “Robust online i-vectors for unsupervised adaptation of DNN acoustic models: A study in the context of digital voice assistants,” in *INTERSPEECH*, 2017, pp. 2401–2405.

[16] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors,” in *ASRU*, 2013, pp. 55–59.

[17] A. Senior and I. Lopez-Moreno, “Improving DNN speaker independence with i-vector inputs,” in *ICASSP*, 2014, pp. 225–229.

[18] S. Tibrewala and H. Hermansky, “Multi-band and adaptation approaches to robust speech recognition,” in *EUROSPEECH*, 1997.

[19] M. J. F. Gales, “Semi-tied covariance matrices for hidden Markov models,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[20] P. Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” *CRIM, Montreal, (Report) CRIM-06/08-13*, 2005.

[21] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative training of deep neural networks,” in *INTERSPEECH*, 2013, pp. 2345–2349.