*INTERSPEECH 2019*
September 15–19, 2019, Graz, Austria

# 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. While it is not a multi-dialect AM training technique in the strict sense, it serves as a competitive baseline to assess the benefit of training all dialects together as compared to sharing knowledge between just two dialects (note that most previous studies have not compared multi-dialect training with transfer learning).

### 2.1. Transfer learning

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

(1) Obtain training targets for dialect *Y* using an *alignment* model (model used to generate forced alignments).

(2) Discard the output layer of the seed model—which is trained with cross-entropy loss using data from dialect *X*—and initialize a fresh output layer with random weights corresponding to the targets of dialect *Y*.

(3) Train only the output layer for a few epochs with cross-entropy loss (i.e. keep the hidden layers fixed); then run cross-entropy training (and optionally, sequence discriminative training) for the entire network.

Since transfer learning does not impose any constraints on the acoustic decision tree of the target dialect, it can be used 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 assume one of the dialects (and its acoustic tree) to be *canonical* and map the phone sets of all the other dialects to the canonical phone set. For a set of dialects $\{D_1, \ldots, D_n\}$, multi-dialect AM training via phone mapping is implemented as follows.

(1) Choose a canonical phone set (and dialect) such that it has the maximum amount of overlap with the remaining phone sets. Assuming that $D_1$'s phone set is canonical, obtain training targets for $D_1$ using a well trained alignment model.

(2) **For each of the dialects** $D_2, D_3, \ldots, D_n$:

• Obtain pronunciations for all the training tokens by looking up a background lexicon whose entries are in the dialect'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 dialect's native lexicon and phone set.

• For each native pronunciation, map all non-canonical phones to their *nearest* phones in the canonical phone set. While these mappings require linguistic inputs, the overhead incurred in this process is minimal because the mappings, once determined, will remain unchanged for a given canonical phone set.

• Obtain forced alignments for training data using the alignment model corresponding to $D_1$.

• Train an AM (pretraining followed by cross-entropy training) using targets produced by the $D_1$ alignment model.

• Regenerate training targets using the AM trained above.

(3) Merge and shuffle the training data prepared in steps (1) and (2), and train a unified AM via pretraining and cross entropy training (and optionally, sequence discriminative training).

Note that the last two sub-steps of step (2) (training an intermediate AM using targets from the canonical alignment model and regenerating the training targets from that AM) are optional; they can be omitted if the dialect in question has very little training data or shares the majority of its phones with the canonical set. The efficacy of phone mapping for a given dialect depends on the percentage of canonical phones in its native phone set. If British English is assumed to be canonical, for instance, phone mapping could be less effective for Indian English that contains named entities in one or more Indic languages like Hindi, Tamil, etc. (since several native phones would be non-canonical).

### 2.3. Shared hidden layer (SHL) training

By allowing dialects to share the hidden layers of the network while maintaining separate output (classification) layers, SHL training provides an effective cross-dialect knowledge transfer mechanism without imposing any constraints on the phone sets and acoustic decision trees (and thereby senones) of the dialects being unified. In this approach, the training targets for each dialect are obtained using a dialect-specific alignment model. After merging and shuffling the datasets from all dialects, the unified AM is built via pretraining followed by cross-entropy training (and optionally, sequence discriminative training).

SHL training is similar to multitask learning except that not all the network parameters are affected by every training sample; backpropagation works such that for a given training sample belonging to a particular dialect, only the corresponding output layer and the 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 dialects increases), one way to speed up training is to incorporate phone mapping into the SHL framework by having shared output layers for groups of *similar* dialects (e.g., American and Canadian English).

### 2.4. One-hot dialect vectors

Adaptive training using one-hot auxiliary input vectors was explored initially in [3], where one-hot language vectors were concatenated with bottleneck features to train a unified DNN AM. More recently, Grace *et al.* studied one-hot dialect vectors as a special case of interpolation of bases methods [13]. As shown in [13], one-hot dialect vectors, which are constant for all training samples of a given dialect, translate essentially to locale-specific bias values. Conceptually, such vectors could be supplied to not only the input layer but also to some or all of the hidden layers. In this paper, they are provided to the input layer as well as all the hidden layers of the unified AM (trained via phone mapping or SHL training).

### 2.5. Frame-level online i-vectors

Using i-vectors as auxiliary inputs is a widely used 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 $\mathbf{T} = [\mathbf{T}_1; \mathbf{T}_2; \ldots; \mathbf{T}_M]$ denote the total variability matrix (T-matrix). 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):

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

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

$$\mathbf{S}_0(l) = \mathbf{S}_0(0)\, e^{-\tau l} + \sum_{i=1}^{M} \gamma_i(l)\, \mathbf{T}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{T}_i$$
$$\mathbf{S}_1(l) = \mathbf{S}_1(0)\, e^{-\tau l} + \sum_{i=1}^{M} \mathbf{T}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{f}_i(l). \tag{2}$$

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

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

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

In the SHL framework, the T-matrix and the Gaussian parameters must be estimated separately for each dialect or group of dialects having a distinct output layer, but this is not required in the phone mapping approach since the senones are shared by all the dialects.

## 3. Experimental results

This section describes the feature extraction pipeline and training setup, followed by a discussion of the unified AM training experiments conducted.

Log filter bank energies (LFBEs), extracted at intervals of 10 ms using 25 ms analysis windows, are used as the primary features for AM training. For each frame, the power spectrum is integrated using 64 Mel-warped filters and subsequently transformed using natural log to obtain LFBEs; they are also normalized by subtracting the autoregressive mean estimate in order to remove channel effects [18].

For extracting i-vectors, senone Gaussians are assumed to have diagonal covariance matrices. The Gaussian and T-matrix parameters are estimated using 40 dimensional features, which are obtained by first stacking 9 LFBE frames (the 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 100, and the T-matrix is estimated via the expectation-maximization (EM) algorithm [20]. The exponential decay factor $\tau$ (see Eqs. (2) and (3)) is set to 0.002, which corresponds to an effective time window of 500 frames or 5 seconds.

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 ($t$+80) ms are used to predict targets at time $t$. Prior to AM training, both LFBEs and i-vectors are mean and variance normalized using training data statistics. One-hot dialect vectors, if used, are supplied to the input layer as well as all the LSTM hidden layers. We tried a few initial experiments where the one-hot vectors were used at the input layer only (as in [13]), but the results were found to be slightly worse.

Table 1: *Train and test data distributions of English dialects (in hours) for the experiments discussed in Sec. 3.1 (train 2K) and Sec. 3.2 (train 40K). Test data is common to both the setups.*

| **dialect** | **train 2K** | **train 40K** | **test** |
|---|---|---|---|
| ***US*** (American) | 1.1K | 13K | 37 |
| ***GB*** (British) | 0.5K | 14.5K | 21 |
| ***IN*** (Indian) | 0.3K | 8.5K | 27 |
| ***AU*** (Australian) | 0.1K | 4K | 14 |

Table 2: *Relative WERRs (%) with respect to isolated (dialect-specific) training for experiments conducted with the 2K hour dataset (Sec. 3.1). All results are at the cross-entropy stage.*

| | **US** | **GB** | **IN** | **AU** | **avg.** |
|---|---|---|---|---|---|
| ***isolated training*** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| ***transfer learning*** | - | 4.2 | 8.6 | 22.8 | 11.9 |
| ***phone mapping*** | 8.3 | 14.4 | 8.4 | 27.4 | 14.6 |
| + 1-hot vectors | 8.7 | 11.8 | 10.9 | 29.3 | 15.2 |
| + i-vectors | 12.0 | 16.5 | 11.2 | 31.3 | 17.8 |
| + 1-hot & i-vectors | 13.3 | 15.9 | 12.2 | 32.0 | 18.4 |
| ***SHL training*** | 7.4 | 14.1 | 10.9 | 31.3 | 15.9 |
| + 1-hot vectors | 7.6 | 14.3 | 11.9 | 29.7 | 15.9 |
| + i-vectors | 11.3 | 18.4 | 12.9 | 34.4 | 19.3 |
| + 1-hot & i-vectors | 11.3 | 17.1 | 12.8 | 32.5 | 18.4 |

Well trained seed models are used as initialization for transfer learning, but for all the 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. For 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 is split into two phases: a chunking phase where the LSTM hidden states are maintained for chunks of 32 frames, and a finetuning phase where the hidden states are propagated until utterance ends. New-bob learning rate scheduling with annealing is applied to both the phases, and 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.

For sequence discriminative training, we use the state-level minimum Bayes risk (sMBR) criterion [21]. AMs trained with cross-entropy loss are first used to obtain the required numerator alignments and denominator lattices, and they are then sMBR trained for two epochs using a learning rate of 1e$-$5 and a non-speech accuracy weight of 0.1.

All AMs are trained using distributed synchronous stochastic gradient descent running on 16 GPUs. For decoding, we use dialect-specific 4-gram language models with Katz smoothing.

### 3.1. Experiments with a 2K hour dataset

This section presents exhaustive experiments using 2000 hours of training data that is composed of four English dialects: American (US), British (GB), Indian (IN) and Australian (AU). For faster turnaround, all models discussed here are trained using

Table 3: *Relative WERRs (%) with respect to isolated (dialect-specific) cross-entropy training for experiments conducted with the 40K hour dataset (Sec. 3.2).*

| | US | GB | IN | AU | avg. |
|---|---|---|---|---|---|
| ***isolated training*** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| + sMBR training | 9.6 | 12.3 | 7.7 | 8.3 | 9.5 |
| ***phone mapping*** | 3.1 | 3.9 | 4.5 | 18.1 | 7.4 |
| + sMBR training | 12.7 | 17.2 | 14.5 | 30.9 | 18.8 |

only cross-entropy loss. The amounts of train and test data used per dialect are shown in Table 1. To prepare these datasets, utterances are randomly sampled from recordings collected through far-field devices (with varied 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, T-matrices and Gaussian models (for i-vector estimation) are borrowed from our in-house ASR systems. Note that the i-vector extractor (T-matrix, etc.) is shared across dialects for phone mapping.

Table 2 shows the WERRs (relative word error rate reductions) achieved with respect to dialect-specific training. The below observations can be made from Table 2.

• Transfer learning provides significant gains over the baseline, but the unified training approaches (both phone mapping and SHL training) offer larger gains in general. This confirms that knowledge sharing across more than two dialects is not only elegant but also beneficial.

• As expected, the gains from unified training are larger for dialects with smaller amounts of training data (GB, IN and AU). Importantly, the gains are also significant for US (which contributes more than half of the total training data). This is a key result considering that a fully-shared network (phone mapping) is able to provide significant gains for all dialects.

• The gains from unified training are somewhat smaller for IN as compared to GB and AU, suggesting that Indian English is quite different from the other dialects. Phone mapping is in fact slightly worse than transfer learning for IN, and this could be due to the high percentage of (non-canonical) Indic phones arising from code switching, e.g. "*play songs from* [Hindi] *dil chahta hai* [Hindi]".

• SHL training offers only slightly better gains than phone mapping, on an average. The latter is therefore a simpler option from a training and maintenance perspective.

• Frame-level i-vectors provide useful additional gains with both phone mapping and SHL training, but one-hot vectors appear to have little effect on performance (both in isolation and in conjunction with i-vectors). This result contradicts the observations made in [13], and we offer a hypothesis in Sec. 3.3 as to why this might be the case.

### 3.2. Experiments with a 40K hour dataset

The purpose of these experiments is to understand if the gains due to phone mapping continue to hold for large-scale datasets and after sequence discriminative training.

In Table 3 we compare phone mapping with dialect-specific training using a large 40K hour training corpus, and the WERRs listed are with respect to the dialect-specific cross-entropy models. The 40K hour dataset comprises the same English dialects as in Section 3.1 and the dialectal breakdown is shown in Table 1. The test data are identical to those of the 2K hour setup.

Table 4: *Relative WERRs (%) for mixed tree multi-dialect model with respect to canonical tree (phone mapped) model.*

| | GB | IN | AU |
|---|---|---|---|
| ***canonical tree*** *(phone mapping)* | 0.0 | 0.0 | 0.0 |
| ***mixed tree*** | -2.2 | -0.6 | 0.3 |

The training data for each dialect is composed of the original recordings plus their corrupted copies generated via simulated reverberation. Note that the corrupted copies use the same training targets as the original recordings.

It is evident from Table 3 that even with large-scale training data, phone mapping provides significant 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.

### 3.3. Analysis: canonical tree versus mixed tree

The purpose of this investigation is to compare a phone mapped model (trained with a canonical tree as described in Section 2.2) with a model whose acoustic tree and senones are estimated by pooling data from all the dialects. The training data for this experiment comprises three English dialects: 3.5K, 2K and 0.5K hours of GB, IN and AU, respectively. Unlike the usual practice, the mixed tree is not built using flat start procedures and native phones; instead, the alignments obtained from a canonical GB model are used to seed the tree-building process. Note that the mixed tree has twice as many senones as the canonical tree.

Table 4 shows WERRs for the mixed tree model with respect to phone mapping (both models are cross entropy trained). We see that despite having a larger tree the mixed tree model does not perform any better than phone mapping; in fact its performance is worse for GB. This suggests that it is better to have a canonical tree that captures intra-dialect triphone contexts but not the cross-dialect variations. By building the mixed tree using flat start and native phones, one might expect the multi-dialect model performance to degrade further owing to higher cross-dialect variations; in such situations, using one-hot dialect vectors as auxiliary inputs could improve performance, as observed in [13]. With the proposed phone mapping approach, however, one-hot vectors seem to have little effect.

## 4. Conclusions

Three main types of knowledge sharing were evaluated in this paper: transfer learning, phone mapping and SHL training. The effect of using one-hot dialect vectors and frame-level online i-vectors was also studied. Experiments show that transfer learning is a quick and effective approach which one can use when a well trained seed model is available. Phone mapping as well as SHL training offer better performance than transfer learning, but since SHL training is only slightly better than phone mapping, on an average, the latter approach can be used to simplify model training and maintenance. While frame-level i-vectors provide useful gains, one-hot vectors do not provide much benefit. Using i-vectors with phone mapping is convenient for multi-dialect training because the T-matrix and Gaussian models can be shared by all the dialects. An important observation from our experiments is that the gains provided by unified modeling hold with large-scale data and 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.