Cross-lingual transfer learning during supervised training in low resource scenarios

Amit Das, Mark Hasegawa-Johnson

Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Illinois, IL 61801, USA
{amitdas, jhasegaw}@illinois.edu

Abstract

In this study, transfer learning techniques are presented for cross-lingual speech recognition to mitigate the effects of limited availability of data in a target language using data from richly resourced source languages. First, a maximum likelihood (ML) based regularization criterion is used to learn context-dependent Gaussian mixture model (GMM) based hidden Markov model (HMM) parameters for phones in target language using data from both target and source languages. Recognition results indicate improved HMM state alignments. Second, the hidden layers of a deep neural network (DNN) are initialized using unsupervised pre-training of a multilingual deep belief network (DBN). The DNN is fine-tuned jointly using a modified cross entropy criterion that uses HMM state alignments from both target and source languages. Third, another DNN fine-tuning technique is explored where the training is performed in a sequential manner - source language followed by the target language. Experiments conducted using varying amounts of target data indicate further improvements in performance can be obtained using joint and sequential training of the DNN compared to existing techniques. Turkish and English were chosen to be the target and source languages respectively.
Index Terms: cross-lingual speech recognition, transfer learning, deep neural networks, hidden Markov models.

1. Introduction

Many interesting research studies have improved the performance of state-of-the-art cross-lingual speech recognition. One of the earlier approaches includes bootstrapping target language acoustic models based on phonemic similarity either using existing monolingual [1], or multilingual models [2], [3]. Recently, DNNs have spurred interest in the speech recognition community due to their superior discriminative modeling capabilities compared to GMM-HMM based modeling techniques. In [4], the outputs of hybrid DNN-HMM system were used to represent posterior probabilities of shared context-dependent states (senones). DNNs have been used in cross-lingual recognition through tandem or hybrid approaches. In the class of tandem approaches: a) the Gaussianized posteriors as the final layer outputs of DNNs [5, 6], or b) the outputs of an intermediate layer (bottleneck extractions) [7, 8], followed by dimensionality reduction using Principal Component Analysis (PCA) are used as distinctive features for training GMM-HMM classifiers. In the class of hydrid approaches, the alignments from GMM-HMM systems are used to train DNNs. The DNN posteriors are used for classification. It has been shown that unsupervised pre-training of hidden layers of a DNN with multilingual data [9] have outperformed hidden layers trained with monolingual data [10], [11]. In [12], DNNs were used for knowledge transfer with zero training data using an "open-target MLP" - an MLP designed to generate posteriors for all possible monophones in the IPA table. DNNs have been effective since they are able to learn complex feature transformations and classify the transformed features using a logistic regression classifier.

Transfer learning has been successfully implemented for semi-supervised learning [13, 14] and supervised learning [15] of GMMs. This work is focused on knowledge transfer from richly resourced source language (English) to low resourced target language (Turkish) using supervised training methods while retaining existing unsupervised methods. First, we use a variable weighting maximum likelihood based supervised training criterion in the HMM framework to recognize Turkish by learning the phonetic structure in English. Improved alignments from HMMs are used for training DNNs. Next, we show that DNNs can also benefit from supervised training of context-dependent senones borrowed from source languages. We outline two supervised training methods. In the first method, we train the DNNs using a weighted cross-entropy error criterion using labeled data from both Turkish and English. In the second method, we train the DNNs in a sequential fashion - first using English data as a means of achieving good initialization and then using Turkish. The rest of the paper is organized as follows. The supervised training methods are outlined in Section 2, experiments and results in Section 3, and finally conclusions in Section 4.

2. Algorithm

Let $\mathcal{X}^{(l)}$ comprise of a sequence of tokens generated from a language with language identity l. Hence, $\mathcal{X}^{(l)} = \left\{\mathbf{x}_1^{(l)}, \mathbf{x}_2^{(l)}, \ldots, \mathbf{x}_{N^{(l)}}^{(l)}\right\}$ where the subscript indicates token index. For the n^{th} token is the set of features vectors during time $t = 1, \ldots, T$ given by $\mathbf{x}_n^{(l)} = \left\{\mathbf{x}_{n,1}^{(l)}, \mathbf{x}_{n,2}^{(l)}, \ldots, \mathbf{x}_{n,T}^{(l)}\right\}$ such that $\mathbf{x}_{n,t}^{(l)} \in \mathbb{R}^D$. Corresponding to $\mathcal{X}^{(l)}$, there are phoneme labels in $\mathcal{Y}^{(l)} = \left\{y_n^{(l)}\right\}$ where $y_n^{(l)} \in \left\{1, 2, \ldots, C^{(l)}\right\}$ where $C^{(l)}$ is the total number of phoneme classes in language l. Let $l \in \{1, 2\}$ where $l = 1$ is the language identity for target language and $l = 2$ is the language identity of all the other source languages. The target language is the language to be recognized. The set of source languages represent all the other languages whose data is shared with the target language in the model estimation process. The set of HMM models $\mathbf{\Theta}$ is given by $\{\mathbf{\Theta}_c\}_{c=1}^{C^{(1)}}$.

2.1. HMM Transfer Learning

The objective is to learn the parameters $\mathbf{\Theta}$ by using limited training data of the target language and and using large amounts of data from source languages. To learn the parameters of a HMM, the objective function to be maximized is the log-likelihood function of the training data. Since the training data consists of both the target and source languages the likelihood of the target data is regularized with the weighted likelihood of the source data. Hence, the new objective is to maximize the total likelihood which is given by,

$$\mathcal{J}(\mathbf{\Theta}_c) = \mathcal{L}(\mathcal{X}^{(1)}; \mathbf{\Theta}_c) + \rho\mathcal{L}(\mathcal{X}^{(2)}; \mathbf{\Theta}_c), \tag{1}$$

where $c = 1, \ldots, C^{(1)}$, and ρ is a constant such that $\rho < 1$. The optimal parameter set is given by,

$$\mathbf{\Theta}_c^{\star} = \arg\max_{\mathbf{\Theta}_c} \mathcal{J}(\mathbf{\Theta}_c)$$

Using the Expectation-Maximization (EM) algorithm, finding the parameters is straightforward and is given by the equations,

$$a_{ij} = \frac{\sum_{n,t} \xi_{n,t}^{(1)}(i,j) + \rho \sum_{n',t'} \xi_{n',t'}^{(2)}(i,j)}{\sum_{n,t} \gamma_{n,t}^{(1)}(i) + \rho \sum_{n',t'} \gamma_{n',t'}^{(2)}(i)},$$

$$\omega_{j,m} = \frac{n_{jm}^{(1)}(1) + \rho n_{jm}^{(2)}(1)}{\sum_m n_{jm}^{(1)}(1) + \rho \sum_m n_{jm}^{(2)}(1)},$$

$$\boldsymbol{\mu}_{jm} = \frac{n_{jm}^{(1)}(\mathbf{X}) + \rho n_{jm}^{(2)}(\mathbf{X})}{n_{jm}^{(1)}(1) + \rho n_{jm}^{(2)}(1)},$$

$$\mathbf{\Sigma}_{jm} = \frac{n_{jm}^{(1)}(\mathbf{X}^2) + \rho n_{jm}^{(2)}(\mathbf{X}^2)}{n_{jm}^{(1)}(1) + \rho n_{jm}^{(2)}(1)}.$$

The quantities $\gamma_{n,t}^{(l)}(j,m), \xi_{n,t}^{(l)}(i,j),\ n_{jm}^{(l)}(\mathbf{X}),\ n_{jm}^{(l)}(\mathbf{X}^2)$ are given in [16, eq.(27, 37, 52, 53, 54)].

2.2. DNN Transfer Learning

This section provides a quick overview of the DNN system used for speech recognition. A DNN generates a vector of probabilities where an element in the vector is the posterior probability of a senone (or monophone) s_k given by $p(q_t = s_k|\mathbf{x}_t)$ for an observed feature vector $\mathbf{x}_t$ extracted from the speech signal at time t. We have dropped the subscript n used to denote token index in $\mathbf{x}_{n,t}$ from all subsequent references since the token index is no longer required. A DNN is composed of multiple layers of affine transforms and activation functions. The output vector of layer l, denoted by $\mathbf{u}^l$, is obtained by applying the affine transform to the outputs of the previous layer followed by a sigmoid activation. This is given by,

$$\mathbf{u}^l = \sigma(\mathbf{W}^l \mathbf{u}^{l-1} + \mathbf{b}^l), \quad 1 \leq l < L. \tag{2}$$

Here, $\mathbf{W}^l$ is the weight matrix between layers $l-1$ and l, $\mathbf{b}^l$ is the bias vector at layer l. For the first layer, $\mathbf{u}^0 = \mathbf{x}_t$. For the final layer L (soft-max layer), the output at node k, $u^L(k)$, is given by,

$$u^L(k) = \frac{\exp(\mathbf{w}_{k,\cdot}^{L^T} \mathbf{u}^{L-1} + b_k^L)}{\sum_j (\exp(\mathbf{w}_{j,\cdot}^{L^T} \mathbf{u}^{L-1} + b_j^L)}, \tag{3}$$

where $\mathbf{w}_{k,\cdot}^{L^T}$ is the k^{th} row of matrix $\mathbf{W}^L$. The output $u^L(k)$ is simply the posterior probability $y_k \triangleq p(q_t = s_k|\mathbf{x}_t)$ where $k = 1, \ldots, C$ and C is the number of senones. Therefore, $\mathbf{u}^L = \mathbf{y} \in [0,1]^C$. The emission probability $p(\mathbf{x}_t|q_t)$ is obtained using $p(\mathbf{x}_t|q_t) = p(q_t|\mathbf{x}_t)p(\mathbf{x}_t)/p(q_t)$ where the state priors $p(q_t)$ are obtained by simply counting the senone labels from HMM forced alignments of the training set, and $p(\mathbf{x}_t)$ is ignored since it is a constant across all states at time t during Viterbi decoding. The DNN is trained to minimize the negative log posterior probability

$$E = -\sum_t \log p(\mathbf{d}|\mathbf{x}_t) = -\sum_t \sum_k d_k \log y_k \tag{4}$$

which is also the cross-entropy error for binary targets where the desired binary target $d_k \in \{0,1\}$ for each $\mathbf{x}_t$ is obtained using HMM based forced alignment and y_k is obtained from (3). In practice, the hidden layers of DNN are initialized using unsupervised layer wise pretraining of RBMs. This is followed by adding a soft-max layer and training the DNN further in a supervised fashion. The supervised training involves updating the parameters (weights and biases) of each layer of the DNN using backpropagation and stochastic gradient descent.

In this study, we use a slightly modified training error criterion of the DNN that takes into account the posterior probabilities of the target and source languages similar to (1). The modified error criterion is,

$$E = E^{(1)} + \rho E^{(2)}, \tag{5}$$

where $E^{(1)}, E^{(2)}$ are the cross-entropy errors of the form (4) for target and source languages respectively and $\rho < 1$. A DNN trained using such an error criterion has a slightly modified weight update rule. Since the training error E is a sum of training errors of individual frames, the error due to a frame originating from source language, i.e. $\mathbf{x}_t^{(2)}$, can be considered separately. Denoting the error associated with $\mathbf{x}_t^{(2)}$ as $\rho E_t^{(2)}$, the term δ_k^L at node k of the final layer (L) of the DNN is simply,

$$\delta_k^L \triangleq \frac{\partial \rho E_t^{(2)}}{\partial a_k^L} = \rho(y_k - d_k),\ \forall k \tag{6}$$

where $a_k = \mathbf{w}_{k,\cdot}^{L^T} \mathbf{u}^{L-1} + b_k^L$. This error is backpropagated to the layers below to compute $\delta_k^{L-1}, \delta_k^{L-2}, \ldots$ etc. for each node k in the lower layers. During backpropagation, the error at the layers below is computed as a linear combination of the errors at the layer above with the weights being the connection weights between two successive layers. Thus the effect of having a scaling term ρ in (6) is reflected as scaled errors at the lower layers. Since the error gradient with respect to the weights at the l^{th} layer $\mathbf{w}^l$ is directly proportional to $\boldsymbol{\delta}^l$, the gradients are also scaled by ρ. During training, frames from both target and source languages are presented in a randomized fashion. Hence, the weight update rule using gradient descent will contain gradients from both languages as follows,

$$\boldsymbol{w}(\tau) = \boldsymbol{w}(\tau - 1) - \eta \nabla E^{(1)} - \eta\rho \nabla E^{(2)}, \tag{7}$$

where τ is the iteration step, and η is the learning rate. Thus the effect of multiplying ρ with $E^{(2)}$ in (5) is a reduced learning rate $\rho\eta$ for frames belonging to the source language as given in (7).

3. Experiments and Results

3.1. Dataset

Modern standard Turkish almost has a one-to-one mapping between written text and its pronunciation [17],[18]. The Turkish corpus in [17] was used. Its training set consists of a total of 3974 utterances (4.6 hours) spoken across 100 speakers. On an average, each training utterance is about 4.12 seconds long. Its full test set consists of 752 utterances spoken acorss 19 speakers. In this study, 558 utterances from 14 randomly selected speakers constitute the test set. The remaining utterances across 5 speakers is the development set. For English, the TIMIT training set consists of 3696 (462 speakers, 3.14 hours). The Turkish corpus follows the METUBET based phonemic representation [17]. Since the phonemic systems are different for Turkish and TIMIT, it is important that both the systems be mapped to a single system prior to running any experiment. In this study, the WORLDBET [19] system was used since its alphabets cover a wide range of multilingual phonemes and it is represented in the amicable ASCII format. A summary of Turkish and English phoneme inventories is given in Table 1. Turkish has a more compact phoneme set than English. There are only 4 vowels that are common to both the languages. Hence the vowel coverage of Turkish using English is only 40% (4/10). However, most overlap occurs in consonants as the consonant coverage is 71% (20/28). The overall monophone coverage is about 63% (24/38).

Table 1: Turkish and English Phoneme Set. "M" = Monopthongs, "D" = Dipthongs, "NS" = "Non-Syllabics", "S" = "Syllabics".

Language	Vowels		Consonants		Total
	M	D	NS	S	
Turkish	10	0	28	0	38
English	13	5	27	3	48
Common	4	0	20	0	24

3.2. Baseline HMM

Context-dependent GMM-HMM acoustic models for Turkish and English were trained using 39-dimensional MFCC features which include the delta and acceleration coefficients. Temporal context was included by splicing 7 successive 13-dimensional MFCC vectors (current +/- 3) into a high dimensional supervector and then projecting the supervector to 40 dimensions using linear discriminant analysis (LDA). Using these features, a maximum likelihood linear transform (MLLT) [20] was computed to transform the means of the existing model. The final model is the LDA+MLLT model. For the English recognition system, the forced alignments obtained from the LDA+MLLT model were further used for speaker adaptive training (SAT) by computing feature-space maximum likelihood linear regression (fMLLR) transforms [21] per subset of speakers. This is the LDA+MLLT+SAT model. The forced alignments from this model were used for training Turkish models which is discussed next. The resulting phoneme error rates (PER) from a total of 27K phonemes are given in Table 2 . The results for Turkish serve as performance ceiling if the full training set was to be available. The results for TIMIT are based on the folded phoneme set which reduces the set to 39 phonemes. All experiments were conducted using the Kaldi toolkit [22].

3.3. HMM Transfer Learning

Phonemes between the two languages sharing the same WORLDBET symbol were mapped. This work differs from previous works [18] involving such hard semantic maps in that we do not completely rely on the knowledge transfer involving such maps. This is true even if certain phonemes between the two languages share the same WORLDBET symbol. This is because the phonetic variations associated with a phoneme in one language can be different from the phonetic variations in another language even though both languages may have identical phonemic representations. Another distinct aspect of this work is that we also map some phonemes from English to Turkish that do not have the same WORLDBET symbols. This many-to-one mapping was based on the degree of similarity in articulation between the two sounds. This is important in the context of limited availability of data in the target language. The question we try to address is can the target model improve its generalization capability by learning from neighboring phonemes if the number of target phonemes present in the training set is insufficient.

Table 2: Phoneme error rates of context-dependent GMM-HMM models using full training sets of Turkish and TIMIT.

GMM-HMM Models	PER (%)
Turkish (LDA+MLLT)	24.25
TIMIT (LDA+MLLT+SAT)	19.6

We converted the triphone alignments of English to Turkish using the above mapping rules before proceeding for monophone training. Monophones were trained using the criterion in (1). For triphone training, as usual, we build a decision tree for each central phoneme with the leaves representing a variety of senones for that central phoneme. Since each senone can represent multiple contexts, differences in contexts between Turkish and English are easily addressed through these senones. Therefore, cross-lingual knowledge transfer occurs both at the monophone and triphone stages using (1) although it is more effective at the triphone stage due to larger number of model parameters. At the LDA+MLLT stage of training, there is no knowledge transfer. This is because the LDA transform cannot be shared between languages. However, knowledge transfer during the triphone stage helps in generating better forced alignments thereby leading to better models at any subsequent stage of training.

In Table 3, the PERs are shown for varying amounts of Turkish training data (100 to 1000 utterances). The first row is the baseline (BL) LDA+MLLT system trained only on the limited Turkish training set. There is no knowledge transfer from English in this system. In the second row is the transfer learned (TL) LDA+MLLT system that uses data from both the languages. The relative improvement in performace is in the range 0.95%-2.35%. Expectedly, with increasing amounts of training data the difference in performance begins to shrink. The value of ρ can be determined from the dev set. We used $\rho = 10^{-2}$ for the first two cases (100, 200) and decreased this by an order of magnitude when the amount of data doubled. The PER scores indicate that improvements due to transfer learning at the HMM stage is marginal. However, when cascaded with DNN, the forced alignments obtained from TL LDA+MLLT models yield signficant improvements as discussed in the next section.

Table 3: Phoneme error rates for LDA+MLLT models trained with limited Turkish utterances and the entire TIMIT set.

	PER (%)			
# Turkish Utts.	100	200	500	1000
(a) BL LDA+MLLT	44.75	39.50	33.65	29.47
(b) TL LDA+MLLT	43.70	38.57	32.92	29.19
Relative PER $\downarrow$ (%)	2.35	2.35	2.17	0.95

3.4. DNN Transfer Learning

In the first step, we build multilingual shared hidden layers (MSHLs) by using greedy layer-wise unsupervised training of stacked restricted Boltzmann machines (RBMs). We do not build monolingual SHLs since it is well known that they are outperformed by MSHLs [10], [11]. Hence, all DNN experiments, inlcuding the baseline, use MSHLs.

We obtained multilingual audio files from the Special Broadcasting Service (SBS) network which contains multilingual radio broadcasts in Australia. It contains over 1000 hours of data in 70 languages. We used about 20 hours of data divided equally between all 70 languages since choice of languages is not important for pre-training and larger amounts of data may not necessarily yield significant gains [9]. We use 6 layers to build the MSHLs with 1024 nodes per layer. The input features to the bottom layer, the Gaussian-Bernoulli RBM, included 5 neighboring frames containing 39-dimensional MFCC vectors spliced together and globally normalized to zero mean and unit variance. The learning rate was set to 0.01. For all subsequent layers, the Bernoulli-Bernoulli RBMs, we used a learning rate of 0.4. Mini-batch size was set to 100 for all layers. All layers were randomly initialized.

After training the MSHLs, we proceed for supervised training of the Turkish DNN by adding a randomly initialized softmax layer and then training the DNN by limited number of available labeled Turkish utterances. Therefore, all DNNs reported in Table 4 use MSHLs and a randomly initialized softmax layer. The DNNs differ in the fine-tuning stage. The learning rate was fixed at 0.008 until improvement in the cross-validation set between two successive epochs fall below 0.5%. The learning rate is halved for all subsequent epochs until the overall accuracy fails to increase by 0.5% or more. At this point, the algorithm terminates.

The PER results are given in Table 4. The first DNN is the baseline (BL) DNN trained on alignments generated by the BL LDA+MLLT system (no transfer) in Table 3. The second DNN is trained on alignments generated by the TL LDA+MLLT system. The relative improvement PERs range from 0.36%-6.18%. Both the DNNs are trained in the same way - MSHLs, then add random soft-max, then Turkish alignments to fine-tune. The only difference is in the HMM alignments obtained from the LDA+MLLT systems. Compared to the PER improvements at the HMM stage in Table 3, the improvements in Table 4 are much better.

In the third DNN, the DNN is trained (fine-tuning step) using the modified training error criterion mentioned in (5). This requires using alignments from both Turkish and English. While Turkish alignments were obtained from TL LDA+MLLT system, English alignments were based on the TIMIT LDA+MLLT+SAT system. We refer to this type of supervised training as "joint" training as mentioned in Table 4. The relative PERs improve further except for the last case (1000 utterances). The relative improvement is always with respect to the BL DNN.

In the next set of DNNs, we again use alignments from both Turkish and English as before, although in a sequential manner. First, we train the DNN using English alignments using early stopping and then retrain the DNN using Turkish alignments until the termination criterion determined by cross-validation accuracy. We refer to this type of supervised training as "sequential" training where we first train using the source language (L2) and then using the target language (L1). We also observed that early stopping while training in L2 leads to better

Table 4: Phone error rates for DNN models trained with HMM state alignments obtained from Table 3.

	PER (%)			
MSHL + rand soft-max +	100	200	500	1000
BL DNN (No Transfer):				
(a) Train using 3(a) ali	45.98	38.75	31.73	26.63
TL DNN (Transfer):				
(b) Train using 3(b) ali	43.14	38.61	30.96	26.10
Relative PER $\downarrow$ (%) (b-a)	6.18	0.36	2.43	1.99
TL DNN (Transfer):				
(c) Joint	42.11	37.81	30.55	26.23
Relative PER $\downarrow$ (%) (c-a)	8.42	2.43	3.72	1.50
TL DNN (Transfer):				
(d) Seq: L2 (2 iter)	39.90	35.98	29.78	25.73
(e) Seq: L2 (6 iter)	39.57	35.61	**29.44**	**25.37**
(f) Seq: L2 (10 iter)	**39.25**	**35.51**	29.56	25.39
Best relative PER $\downarrow$ (%)	14.64	8.36	7.22	4.73

PERs. Here, the early stopping criterion is simply the number of epochs which is 2 or 6 or 10 as shown in the table. For cases where target data was very limited (100 or 200), the number of L2 epochs was 10. Otherwise, 6 epochs was sufficient. More epochs do not guarantee better accuracies. As demonstrated in Table 4, the relative PERs for all cases improve further in the range 4.73%-14.64%. In terms of absolute PER improvement compared to the baseline, the improvement is in the range 1.26%-6.73%. On an average, the absolute PER improvement compared to the BL DNN is about 3.38%. Through these experiments, it is clear that knowledge transfer can also occur at the supervised training stages.

We think that initializing weights by sequential training is closest to the work on MLP initialization schemes of Vu et al. [12]. In [12], they use the weights of multilingual MLP to initialize the weights of a target language MLP (including the softmax layer). But their MLPs use monophone based posteriors. In this work, we showed that the DNNs are able to leverage the knowledge of the phonetic structure of the context-dependent space by using the weights of source language senones assuming there is a mapping between the phonemes of the languages. In addition, we showed that the DNNs are also able to leverage knowledge by using many-to-one mapping. Therefore, even neighboring phonemes of source language can also help model target phonemes in the target language. This is helpful especially in low resource scenarios. Future work includes using labeled multilingual data at the supervised fine-tuning stage. Furthermore, instead of a single weight for all source language data, phoneme dependent weights could perhaps be used to improve the performance of joint DNN training.

4. Conlcusions

In this study, cross-lingual transfer learning techniques using supervised training were investigated for low resource scenarios. First, a maximum likelihood transfer learning technique was proposed for training GMM-HMM models using labeled data from both target and source languages. Next, using a modified training error criterion, a joint DNN training method was proposed which also uses labeled data from both languages. Finally, DNNs could also be trained sequentially by applying early stopping for the source language.

5. References

[1] B. Wheatley, K. Kondo, W. Anderson, and Y. Muthusamy, "An evaluation of cross-language adaptation for rapid HMM development in a new language," in *ICASSP.*

[2] T. Schultz and A. Waibel, "Fast bootstrapping of LVCSR systems with multilingual phoneme sets," in *Eurospeech.*, 1997.

[3] J. Kohler, "Language adaptation of multilingual phone models for vocabulary independent speech recognition tasks," in *ICASSP.*, 1998, vol. 1, pp. 417–420.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 30–42, Jan 2012.

[5] A. Stolcke, F. Grezl, M.-Y. Hwang, X. Lei, N. Morgan, and D. Vergyri, "Cross-domain and cross-lingual portability of acoustic features estimated by multilayer perceptrons," in *ICASSP.*, 2006, pp. 321–324.

[6] S. Thomas, S. Ganapathy, and H. Hermansky, "Cross-lingual and multi-stream posterior features for low resource LVCSR systems," in *Interspeech*, 2010.

[7] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *ICASSP.*, 2007.

[8] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual MLP features for low-resource LVCSR systems," in *ICASSP.*, 2012.

[9] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *IEEE SLT Workshop*, 2012.

[10] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross language knowlege transfer using multilingual deep neural network with shared hidden layers," in *ICASSP.*, 2013.

[11] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP.*, 2013.

[12] N. Vu, W. Breiter, F. Metze, and T. Schultz, "An investigation on initialization schemes for multilayer perceptron training using multilingual data and their effect on ASR performance," 2012.

[13] J.-T. Huang, "Semi-supervised learning for acoustic and prosodic modeling in speech applications," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2012. [Online]. Available: http://www.isle.illinois.edu/sst/pubs/2012/huang12thesis.pdf

[14] J.-T. Huang and M. Hasegawa-Johnson, "On semi-supervised learning of Gaussian mixture models for phonetic classification," in *NAACL HLT Workshop on Semi-Supervised Learning, 2013*.

[15] P. Huang and M. Hasegawa-Johnson, "Cross-dialectal data transferring for Gaussian mixture model training in Arabic speech recognition," *4th International Conference on Arabic Language Processing*, pp. 119–123, 2012.

[16] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[17] Özgül Salor and M. Demirekler, "On developing new text and audio corpora and speech recognition tools for the Turkish language," in *International Conf. Spoken Language Processing*, 2002, pp. 349–352.

[18] T. Schultz and A. Waibel, "Language independent and language adaptive acoustic modeling for speech recognition," vol. 35, pp. 31–51, Aug 2001.

[19] J. L. Hieronymus, "ASCII phonetic symbols for the world's languages: WORLDBET," Bell Labs Technical Memorandum, Tech. Rep.

[20] R. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *ICASSP.*, 1998, pp. 661–664.

[21] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language.*, vol. 12.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *IEEE ASRU Workshop.*, 2011.