# MULTI-LINGUAL SPEECH RECOGNITION WITH LOW-RANK MULTI-TASK DEEP NEURAL NETWORKS

*Aanchan Mohan, Richard Rose*

Department of Electrical and Computer Engineering, McGill University, Montreal, Canada

## ABSTRACT

Multi-task learning (MTL) for deep neural network (DNN) multi-lingual acoustic models has been shown to be effective for learning parameters that are common or shared between multiple languages[1, 2]. In the MTL paradigm, the number of parameters in the output layer is large and scales with the number of languages used in training. This output layer becomes a computational bottleneck. For mono-lingual DNNs, low-rank matrix factorization (LRMF) of weight matrices have yielded large computational savings[3, 4]. The LRMF proposed in this work for MTL, is for the original language-specific block matrices to "share" a common matrix, with resulting low-rank language specific block matrices. The impact of LRMF is presented in two scenarios, namely : (a) improving performance in a target language when auxiliary languages are included during multi-lingual training; and (b) cross-language transfer to an unseen language with only 1 hour of transcribed training data. A 44% parameter reduction in the final layer, manifests itself in providing a lower memory footprint and faster training times. An experimental study shows that the LRMF multi-lingual DNN provides competitive performance compared to a full-rank multi-lingual DNN in both scenarios.

***Index Terms***— Low-resource speech recognition, Multi-lingual speech recognition, Neural Networks for speech recognition, Multi-task Learning

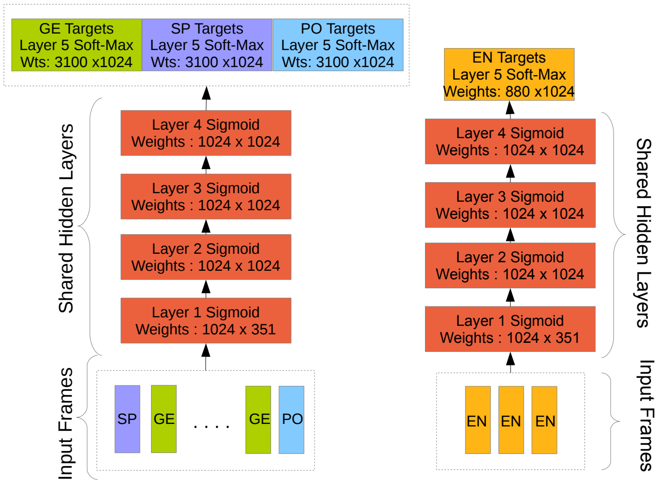
## 1. INTRODUCTION

In many automatic speech recognition (ASR) applications it is a challenge to configure acoustic models in a new language, especially when the amount of transcribed data required to do so is limited. This problem is often addressed by including auxiliary languages during training acoustic models with the goal of improving ASR performance in a target language [5, 6]. When the new language is not seen during training, existing acoustic models obtained using multi-lingual training are used either to initialize training in a new language, or adapt the model-parameters from this initial acoustic model to the new target language [1, 2, 7, 8, 9, 10]. This process is often termed as *cross-language transfer* [5].

Recently, hybrid deep neural network - hidden Markov model (DNN-HMM) acoustic models have yielded state-of-the-art results on many well-known speech tasks [11, 12]. Further, it has been shown that hidden layers in DNNs, trained in a multi-lingual manner are transferable across languages [9, 13, 1]. The work in this paper is applied to two paradigms associated with multi-lingual training of hybrid DNN-HMM acoustic models for ASR. The first paradigm is multi-task learning (MTL) [14] where the parameters of a single DNN are trained using utterances from multiple languages. The interest here is in the multi-lingual DNN architecture depicted in the block diagram of Figure 1a. The goal in this scenario is to improve ASR performance in a target language by including auxiliary languages during training. The input and hidden layers of this network are shared across multiple languages and the output layer consists of separate activations for each language. The second paradigm is a cross language transfer (XLT) scenario depicted in Figure 1b where a hybrid DNN trained from a set of well-resourced languages is leveraged in training an acoustic model for a new under-resourced language. This is done by removing the final layer of the multi-lingual network in Figure 1a, and replacing it with a new output layer whose weights are trained from available data obtained from the new language as shown in Figure 1b.

Techniques are investigated for minimizing language-specific cost functions jointly for all languages in the multi-task learning paradigm and in the unseen target language for the cross-language transfer paradigm. Techniques are also investigated for minimizing computational complexity associated with the networks in Figure 1. This is accomplished through the use of low rank matrix factorization (LRMF) to reduce the total number of weights in the last layer of these networks. The impact of this low-rank matrix factorization is investigated on multi-task DNN training and cross-language transfer.

The paper is organized as follows. First, an overview of DNN-HMM hybrid acoustic models is presented in Section 2. A description of multi-task learning for multi-lingual ASR is then presented in Section 3. Section 4, introduces low-rank factorization for the final soft-max layer. Section 5 presents an experimental study investigating the impact of low-rank factorization of the soft-max layer on multi-task DNN training and cross-language transfer. Finally, the paper concludes with a discussion regarding our experimental study and detailing directions of future work.



(a) Multi-task DNN Training (b)Cross-Language Transfer

**Fig. 1**: Multi-lingual DNN Training

This work was supported by the Nuance Foundation.

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

## 2. HYBRID DNN-HMM ACOUSTIC MODELS

In hybrid DNN-HMM[15, 11, 12] acoustic models, a DNN is used directly to model the posterior probability $p(j|\mathbf{x}_t)$, of an HMM context-dependent state $j \in 1, \ldots, J$, given an input speech frame $\mathbf{x}_t$. Here $J$ denotes the total number of context-dependent HMM states. These frame dependent posteriors are converted to likelihoods using HMM state prior probabilities. HMM state prior probabilities are obtained using the training data and an initial continuous-density HMM (CD-HMM) system.

A DNN with $L$ layers has parameters $\Theta = \{\mathbf{W}^l, \mathbf{b}^l\}_{l=1}^{L}$, where $\mathbf{W}^l$ are referred to as weights for a layer, and $\mathbf{b}^l$ are referred to as biases for that layer. Here $\Theta$ is used to denote the collective set of parameters for a DNN. The input to each layer, after having undergone an affine transformation due to the weights $\mathbf{W}^l$ and biases $\mathbf{b}^l$ in that layer, produces a signal $\mathbf{h}^l$ that is propagated through an element-wise activation function to yield an input to the next successive layer. A popular choice of the activation function for each unit $i \in 1, \ldots, I$, in layer $l \in 1, \ldots, L-1$ is the non-linear sigmoid activation function $\sigma_i(h_i) = 1/(1 + \exp^{-h_i})$. The final layer $L$ is a classification layer where the units (and hence training targets) are context-dependent HMM states $j \in 1, \ldots, J$. This layer has a soft-max non-linearity $soft-max(h_j^L) = \exp(h_j^L)/\sum_j \exp(h_j^L)$ that provides an approximation to the required posterior probability $p(j|\mathbf{x}_t)$.

An $L$-layer DNN, with parameters $\Theta$ is trained using mini-batch stochastic gradient descent (SGD) to minimize a suitable error criterion $J(\Theta)$. The error criterion used in this work is the cross-entropy error criterion given by $J(\Theta) = -\sum_t \sum_j \hat{p}(j|\mathbf{x}_t) \log(p(j|\mathbf{x}_t))$. The quantity $\hat{p}(j|\mathbf{x}_t)$ treated as the ground-truth classification for the frame $\mathbf{x}_t$ is obtained from a Viterbi-alignment using the CD-HMM. The SGD is carried out over a mini-batch of randomly selected $m$ speech frames. The gradients for the parameters $\Theta$ are computed and updated using the back-propagation algorithm[16]. Recently, unsupervised and the supervised pre-training procedures have been motivated for initializing DNN fine-tuning using back-propagation[12, 11]. Without preference for one method over another we chose unspervised Restricted Boltzman Machine (RBM) pre-training as motivated by the authors of [12, 11] to initialize our DNNs before fine-tuning.

## 3. MULTI-TASK LEARNING FOR MULTI-LINGUAL ASR

The challenging task of building an ASR system in a low-resource language is often alleviated by borrowing information from a so-called resource-rich language ASR system. Irrespective of the modelling approach used, the idea of *transferring* or *sharing* statistical model parameters that characterize acoustic-phonetic knowledge across languages is a well-known approach to this problem. Multi-task learning (MTL) focuses on learning different yet related tasks simultaneously[17, 14] with a common classifier. The global cost function for MTL is a sum of costs of each of the individual tasks. The data from each of the individual tasks during training is presented in no particular manner. Huang et al.[1], in their work employ MTL for cross-language transfer in DNN acoustic modelling. Previous work by Ghoshal et al.[9] is an example of adaptive learning for knowledge transfer between languages. Both adaptive learning and MTL are sub-instances of transfer learning[14]. For multi-lingual training, it has been acknowledged by the authors in [9] that the language sequential manner of training multi-lingual DNNs could be suboptimal. In this work we decided to adopt a multi-task learning approach to avoid ambiguity regarding the best language-sequential order for multi-lingual DNN training.

In training a DNN in a multi-task fashion, each mini-batch contains data from all the tasks or in this context all the languages in question. The back-propagation algorithm needs to be modified in order to train a DNN multi-task fashion. The parameters of all the shared hidden layers are updated with the data from all the languages in a mini-batch. On the other hand, the parameter update for the each language specific soft-max or the final classification layer is computed only from the language specific examples in the mini-batch. During decoding the posteriors are taken from the language-specific outputs in the final layer. Multi-task DNN training and its use in decoding is as illustrated in Figure 1a.

Cross-language transfer is illustrated in Figure 1b. After multi-task DNN training, the shared layers are kept intact and a new randomly initialized soft-max layer is placed on top. With a limited amount of training data only the parameters of the new soft-max layer are updated[1, 18]. The impact of LRMF on cross-language transfer is investigated in Section 5.3.

## 4. LOW-RANK FACTORIZATION

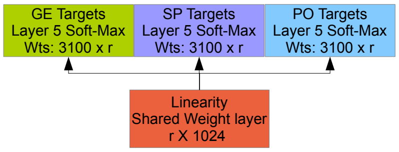


**Fig. 2**: Soft-max layer after factorization

From Figure 1a, it is apparent that the weight matrix for this final layer is composed of three block matrices corresponding to each of the three languages used for multi-task DNN training. The number of parameters in this layer scale with the number of languages used in training. This slows down multi-task training considerably, and necessitates the use of large amounts of memory and disk space.

Recently Sainath et. al[3] and Xue at al.[4], have shown low-rank matrix factorization(LRMF) to be an effective method reducing computational and space complexity for mono-lingual DNNs. In [3], the network is restructured replacing the weight matrix for the final soft-max layer by two matrices with a linear "bottleneck" layer in-between. The DNN in this case implicitly learns the factorization during DNN training. On the other hand in [4] is it shown that considerable savings is possible by using Singular Value Decomposition-based LRMFs for the weight matrices in all layers of an initially well trained DNN. Their temporary loss in performance is recovered through further fine tuning.

LRMFs have been explored for building a robust language-independent front end feature extractor for low-resource Tandem DNN based ASR [19]. The goal in this work on the other hand is to use LRMFs to reduce computational complexity of multi-task DNN training for hybrid DNN-HMM systems. Figure 2, illustrates how the three language-specific block matrices in the final weight layer are replaced by a shared matrix of size $r \times 1024$ and three language specific matrices of size $3100 \times r$. Here $r$ is a parameter that determines the rank of the shared matrix and the language-specific matrices. In this work a value of $r = 512$ is used. The number of parameters in output layer of the original network is $3 \times 3100 \times 1024 = 9523200$. In a LRMF with $r = 512$ the number of parameters in the final layer is $3 \times 3100 \times 512 + 512 \times 1024 = 5285888$. This simple change amounts to a 44% reduction in the number of parameters in the output layer.

## 5. EXPERIMENTS

This section presents the results of an experimental study performed to evaluate the impact of the multi-task learning (MTL) for DNNs and cross-language transfer (XLT) paradigms displayed in Figure 1. The impact of LRMF is also investigated for both training paradigms. Multi-task training is performed using the German as the target language and Spanish and Portuguese auxiliary languages taken from the Globalphone speech corpus [20, 21]. MTL is evaluated by observing ASR performance on German when compared to training a mono-lingual DNN system on German alone. XLT is evaluated using a 1 hour subset the English language Wall Street Journal (WSJ) corpus as data representing the "low resource" language. The multi-lingual DNN trained from the three high resource languages is used to initialize DNN training for XLT using a 1 hour subset of the WSJ corpus consisting of utterances selected randomly from the 15-hour WSJ-SI-84 training set.

### 5.1. Baseline Monolingual Systems

In this section, baseline ASR word error rates (WERs) are presented for monolingual-trained CD-HMM and DNN-HMM German systems. Baseline ASR WERs are also presented for the 1h monolingual CD-HMM and DNN-HMM English systems. Results obtained using the full English training set have also been included for comparison.

#### *5.1.1. Speech Corpora*

The statistics for the three languages from the Globalphone corpus are presented in Table 1. Tri-gram language models for German, Spanish and Portuguese were obtained from Karlsruhe University. All of the English training is carried out using the WSJ0-SI-84 training set which consists of about 15 hours of audio data (7138 utterances). To simulate a low-resource scenario a 1-hour (483 utterances) subset of the WSJ0-SI-84 training set is created by randomly selecting utterances from the 15 hour full-training set. The evaluations for WSJ-English are provided on the open-vocabulary 20k word test condition similar to [22]. Pruned tri-gram language models are used during decoding.

**Table 1**: Globalphone speech data used for experimental study

| | Amount of data (Hrs) | | | Characteristics | | |
|---|---|---|---|---|---|---|
| Language | Train | Dev. | Test | # Dict. Entries | Phones | # Speakers |
| German | 14.9 | 2.00 | 1.5 | 41k | 43 | 77 |
| Spanish | 17.5 | 1.66 | 2.0 | 39k | 42 | 100 |
| Portuguese | 22.75 | 1.65 | 1.75 | 59k | 45 | 101 |

#### *5.1.2. Mono-lingual CDHMM Models*

This sub-section describes the details of the Globalphone and the WSJ baseline systems. The systems were built using 39-dimensional MFCC features [23] with 13 cepstral coefficients, and their first and second derivatives. Speaker based cepstral mean and variance normalization (CMVN) was applied to the extracted features. In the Globalphone Spanish, Portuguese and German systems the number of target context-dependent states were set to 3100 for each language following the setup of the authors in [24]. The approximate total number of Gaussians in the each of the Globalphone CDHMM models was 50k with an average of 16 Gaussians per context-dependent state. On the other hand, the 15-hour (WSJ0-SI-84 with 7138 utterances) and 1-hour (WSJ0-SI-84 with 483 utterances) English systems have 1968 and 880 context-dependent states respectively with an average of 8 Gaussians per CDHMM state. For this study cross-word tri-phone CDHMM systems were built using the Kaldi speech recognition toolkit[22].

The performance metric used in this study is the word error rate (WER) expressed as a percentage(%). The performance of the baseline German CDHMM system on the Globalphone development evaluation set are presented in the first row of Table 2. The results for German are in the range of previously published numbers [13]. For the 15h and 1h English WSJ systems results in WER are presented in the first row Table 3 on the WSJ Nov93 development (consisting of 503 utterances) and the WSJ Nov92 (consisting of 333 utterances) evaluation sets.

**Table 2**: Results for German on the dev. and eval. sets

| System | Dev. | Eval. |
|---|---|---|
| CDHMM | 14.98 | 22.62 |
| DNN-HMM | 12.54 | 19.92 |
| ML-DNN-HMM | 11.52 | 18.14 |
| ML-DNN-HMM-FACT | 11.72 | 18.50 |

**Table 3**: Results for WSJ systems Nov93 dev. set and Nov92 test set

| System | 15-h | | 1-hr | |
|---|---|---|---|---|
| | Dev. | Eval. | Dev. | Eval. |
| CDHMM | 17.29 | 12.12 | 28.71 | 21.41 |
| DNN-HMM | 14.08 | 9.30 | 27.95 | 18.47 |

#### *5.1.3. Mono-lingual DNN-HMM Models*

This section describes the details of our mono-lingual DNN training setup. All DNNs used in this study consist of 4 hidden layers of 1024 units each. These DNN parameters were chosen to be consistent with previous studies [13, 25] that have used the Globalphone corpus for hybrid DNN-HMM ASR. All of the DNNs were initialized using RBM pre-training with a learning rate of $\eta_p = 0.005$. A mini-batch size of $m = 256$, a learning rate of $\eta_b = 0.01$ and a first order low-pass momentum of 0.9 was used for DNN fine-tuning in all of the languages. Training was halted when the cross-entropy cost was seen to plateau over multiple epochs on the validation sets [11]. Input to the DNN consists of 39-dimensional CMVN MFCC features taken over a $c = 4$-frame context window to yield a final input feature vector size of 351. State labels were obtained by a Viterbi-alignment of the language-specific MFCC features against the their respective CDHMM models. The German mono-lingual DNN was trained with 3100 CDHMM states as targets. For the English 15h and 1h systems, 1968 and 880 CDHMM states respectively were used as targets. All of the DNN training was carried out on NVIDIA CUDA-capable GPU cards. A Python based DNN trainer[1] based on Gnumpy[26] was adapted for this work and interfaced with the Kaldi decoder to yield hybrid DNN-HMM systems.

The second row of Table 2 lists the WER obtained with the mono-lingual German DNN-HMM system on the development and evaluation sets. The second row of Table 3 lists the results for the 15h and 1h mono-lingual English DNN-HMM systems.

### 5.2. Multi-lingual DNN Training - Multi-task learning

In this section experimental results are reported on German when two auxiliary languages namely Spanish and Portuguese along with German are used for a multi-lingual training. As mentioned in Section 3, MTL DNN training is used to build a single multi-lingual DNN. There is a dedicated soft-max layer for each language, while the hidden and input layers are kept shared. We implemented the MTL DNN training algorithm as an additional library to the Python based DNN trainer. The configuration (input and hidden layers) and hyperparameters (learning rates, mini-batch size, momentum) used for multi-lingual DNN training is similar to the mono-lingual setup.

The performance of the multi-lingual system on the German development and evaluation sets is listed in the third row of Table 2 indicated as ML-DNN-HMM. By looking at results from Table 2,

[1] http://www.cs.toronto.edu/˜gdahl/gdbn.tar.gz

a relative improvement of 8.9% and 8.1% WER over the German mono-lingual DNN baseline is observed on the evaluation and development sets respectively. Interestingly similar to observations in [1], improvements in performance were also observed for Spanish and Portuguese by the MTL multi-lingual DNN over their respective mono-lingual baselines on the development sets.

Next, as mentioned in Section 4, the configuration of the multi-lingual DNN is changed to reduce the number of parameters in the final weight layer. As noted earlier LRMF with a rank of $r = 512$ reduces the parameters by 44% in the final layer of the MTL multi-lingual DNN. This results in faster training times, and a lower memory footprint. With our training setup a relative speed-up of about 27.7% was observed when training on an NVIDIA K20 GPU. Further, there is a relative reduction of 28% in the amount of memory used due to fewer parameters in the LRMF-MTL-DNN compared to the full-rank MTL-DNN. $L2$ regularization was used for the language-specific layers with a weight decay coefficient of value $\lambda = 10^{-5}$ for German and Portuguese, and $\lambda = 10^{-4}$ for Spanish. The values were selected by observing the WER on the language specific development sets. It was observed that the WER performance is sensitive to the value of the weight decay parameter $\lambda$. The results for the performance of the MTL mutli-lingual DNN with LRMF on the soft-max layer is reported on the German development and evaluation set in the fourth row of Table 2. The factorized model is referred to as ML-DNN-HMM-FACT. It can be seen that the LRMF multi-lingual DNN is still able to provide a 7% relative improvement over the mono-lingual DNN baseline on the evaluation set. A possible reason for the slight degradation in WER compared to the full-rank multi-lingual DNN is that the weight decay parameters are probably not optimal. Future work will investigate better values for the weight decay parameters, and compare the performance of weight decay to other regularization methods in the language specific layers.

### 5.3. Multi-lingual DNN Training - Cross Language Transfer

To investigate the impact of LRMF on cross-language adaptation in a low-resource situation, XLT experiments are presented with training on the and 1h subset of the WSJ corpus. The results of the 1h mono-lingual DNN system from the first row of Table 3 forms the baseline for XLT experiments are reproduced in the first row of Table 4 for reference. For effective XLT in a low-resource situation, the soft-max layer of a multi-lingual DNN is replaced for the new target language as illustrated in Figure 1b. It must be noted that in the case of the LRMF multi-lingual DNN model, only the language specific component of the top-layer needs to be replaced. The "shared" component of the factorization still remains intact. After replacing the soft-max layer for the new target language, only the parameters of the soft-max layer are updated. For the LRMF multi-lingual DNN only the the language specific weight matrix is updated. The shared component is left untouched. An $L2$ regularization coefficient of value of $\lambda = 10^{-3}$ was used for the language-specific factorization component for successful XLT of the LRMF multi-lingual DNN. ML-DNN HMM in Table 4 denotes the multi-lingual DNN models trained without LRMF, and ML-DNN-HMM-FACT denotes the model with LRMF.

**Table 4**: Results for cross-language transfer to English

| System | 1-hr | |
|---|---|---|
| | Dev. | Eval. |
| DNN-HMM | 27.95 | 18.47 |
| ML-DNN-HMM | 23.83 | 16.27 |
| ML-DNN-HMM-FACT | 21.64 | 15.97 |

The first conclusion from Table 4 is that cross-language transfer with shared layers taken from the multi-lingual DNN clearly is better than building a DNN system in the low-resource language from scratch. Row 2 of Table 4, shows the results for updating just the soft-max layer when initializing training with the multi-lingual DNN that has not undergone any factorization. Row 3 of Table 4, shows the results for updating just the soft-max layer when initializing training with the multi-lingual DNN with LRMF. In this case the LRMF mutli-lingual DNN is able to provide better performance compared to the full-rank DNN when only 1-hr of training data is used. The improvement seen here could be perhaps due to the extra "shared" component in the soft-max layer. To make a definite conclusion requires detailed experimentation, and is left as a direction for future work.

## 6. CONCLUSION

In this work first, multi-lingual DNN training using multi-task learning (MTL) was studied for improving ASR performance in German which was set as the target language. Spanish and Portuguese were used as auxiliary languages. A relative improvement of 8.9% was observed with the multi-lingual DNN system over the mono-lingual hybrid DNN-HMM system on the German evaluation set. To reduce computational complexity, a low-rank matrix factorization (LRMF) of the weight matrices in the final layer was proposed. The factorization yields a sizeable reduction in the number of parameters in the final layer which manifests itself in faster training times, and a lower memory footprint. The resulting LRMF system still maintained a 7% improvement over the mono-lingual baseline DNN system. Weight decay based $L2$ regularization was found to play an important role in the performance of the LRMF multi-lingual DNN. A possible reason for the slight degradation in WER compared to the full-rank multi-lingual DNN is that the setting of the weight decay parameters were probably not optimal.

Cross-language transfer was also studied for building acoustic models for a new low-resource target language that is not seen in training. The existing multi-lingual DNN trained using MTL was used for initializing cross-language transfer on 1h of English data from the WSJ corpus. It was concluded that better performance was observed during the process of cross-lingual transfer due to LRMF.

Future work will investigate better values for the weight decay parameters LRMF multi-lingual DNN, and compare the performance of weight decay to other regularization methods to prevent over-fitting in the language specific layers. Further, the impact of choosing various values of the rank $r$ on the ASR WER will be studied. Another direction of future work is to investigate if the extra "shared" component provides an added advantage during cross-lingual transfer. Lastly, following the work in [4], explicit factorization of the weight matrices could also be compared to the approach presented in this work.

## 7. REFERENCES

[1] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. of ICASSP*. IEEE, 2013.

[2] Georg Heigold, Vincent Vanhoucke, Andrew Senior, Patrick Nguyen, M Ranzato, Matthieu Devin, and Jeffrey Dean, "Multilingual acoustic models using distributed deep neural networks," in *Proc. of ICASSP*. IEEE, 2013.

[3] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. of ICASSP*. IEEE, 2013.

[4] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *Proc. of INTERSPEECH*, 2013.

[5] Tanja Schultz and Katrin Kirchhoff, *Multilingual speech processing*, Academic Press, 2006.

[6] Lukas Burget, Petr Schwarz, Mohit Agarwal, Pinar Akyazi, Kai Feng, Arnab Ghoshal, Ondrej Glembek, Nagendra Goel, Martin Karafiát, Daniel Povey, et al., "Multilingual acoustic modeling for speech recognition based on subspace gaussian mixture models," in *Proc. of ICASSP*. IEEE, 2010.

[7] Tanja Schultz and Alex Waibel, "Language independent and language adaptive acoustic modeling for speech recognition," *Speech Communication*, vol. 35, pp. 31–51, 2001.

[8] Hui Lin et al., "A study on multilingual acoustic modeling for large vocabulary ASR," in *Proc. of ICASSP*, april 2009.

[9] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *Proc. of ICASSP*, 2013.

[10] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Cross-lingual and multi-stream posterior features for low resource LVCSR systems.," in *Proc. of INTERSPEECH*, 2010.

[11] Geoffrey Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[12] George Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, , no. 99, pp. 1–1, 2010.

[13] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc. IEEE Workshop on Spoken Language Technology*, Miami, Florida, USA, December 2012, pp. 246–251.

[14] Li Deng and Xiao Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.

[15] Steve Renals, Nelson Morgan, Hervé Bourlard, Michael Cohen, and Horacio Franco, "Connectionist probability estimators in HMM speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 2, no. 1, pp. 161–174, 1994.

[16] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, 1988.

[17] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[18] Xiao Li and Jeff Bilmes, "Regularized adaptation of discriminative classifiers," in *Proc. of ICASSP*. IEEE, 2006.

[19] Yu Zhang, E. Chuangsuwanich, and J. Glass, "Extracting deep neural network bottleneck features using low-rank matrix factorization," in *Proc. of ICASSP*, May 2014, pp. 185–189.

[20] Tanja Schultz, "Globalphone: a multilingual speech and text database developed at Karlsruhe university.," in *Proc. of INTERSPEECH*, 2002.

[21] Tanja Schultz, Ngoc Thang Vu, and Tim Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *Proc. of ICASSP*. IEEE, 2013.

[22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proc. of IEEE ASRU*, 2011.

[23] Steven Davis and Paul Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 4, pp. 357–366, 1980.

[24] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *Proc. of ICASSP*. IEEE, 2013.

[25] Yajie Miao and Florian Metze, "Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training.," in *Proc. of INTERSPEECH*, 2013.

[26] Tijmen Tieleman, "Gnumpy: an easy way to use GPU boards in Python," *Department of Computer Science, University of Toronto*, 2010.