SEMI-SUPERVISED TRAINING OF DEEP NEURAL NETWORKS

Karel Veselý, Mirko Hannemann, Lukáš Burget

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Czech Republic

ABSTRACT

In this paper we search for an optimal strategy for semi-supervised Deep Neural Network (DNN) training. We assume that a small part of the data is transcribed, while the majority of the data is untranscribed. We explore self-training strategies with data selection based on both the utterance-level and frame-level confidences. Further on, we study the interactions between semi-supervised frame-discriminative training and sequence-discriminative sMBR training. We found it beneficial to reduce the disproportion in amounts of transcribed and untranscribed data by including the transcribed data several times, as well as to do a frame-selection based on per-frame confidences derived from confusion in a lattice.

For the experiments, we used the Limited language pack condition for the Surprise language task (Vietnamese) from the IARPA Babel program. The absolute Word Error Rate (WER) improvement for frame cross-entropy training is 2.2%, this corresponds to WER recovery of 36% when compared to the identical system, where the DNN is built on the fully transcribed data.

Index Terms— semi-supervised training, self-training, deep network, DNN, Babel program

1. INTRODUCTION

The current state-of-the-art ASR systems require a relatively large database to be trained on. This needs to be recorded and manually transcribed, along with a collection of linguistic data resources for lexicon and language modeling. However, the data preparation is slow and costly, which can be prohibitive especially in case of languages with low number of speakers. On the other hand, the data preparation time and cost can be reduced by transcribing only a subset of the data and using the rest for semi-supervised training.

Very practical are the self-training methods [1] [2][3], in which the transcribed data are used to build a seed model. This is later used to decode untranscribed data and the resulting hypotheses are used as ground-truth transcripts in further training. Typically, the data are selected according to some form of a confidence measure.

Another family of semi-supervised methods aim to systematically incorporate the uncertainty of unlabeled data into the objective functions and minimize its entropy [4] [5] [6]. Other possible method is based on the feature-space manifold assumption using a graph-based framework [7], where the unsupervised datapoints are assigned to the class of the nearest supervised datapoints.

In our work, we search for an optimal self-training strategy for DNN, which is trained in three stages:

1. pre-training [8][9][10] – layer-wise training of Restricted Boltzmann Machines (RBM) by Contrastive Divergence algorithm,
2. frame-classification training [11][12] – mini-batch Stochastic Gradient Descent (SGD), optimizing frame cross-entropy,
3. sequence-discriminative training [13][14][15] – SGD with per-sentence updates, optimizing state Minimum Bayes Risk (sMBR).

For unsupervised RBM training, adding more data is trivial. In case of supervised training, we experiment with data selection based on both sentence-level and frame-level confidences. We also experiment with SGD using frame-wise confidence-weighted gradients. Finally we study interactions between semi-supervised frame-classification training and sequence-discriminative training.

A parallel work, which focuses on semi-supervised training of bottleneck-feature extractor, and which is trained on the same dataset is described in [16]. The same dataset is also used in [17], where the authors focus on semi-supervised sequence-discriminative training of a GMM-HMM system.

This paper is organized as follows: section 2 describes experimental setup and seed model, in section 3 we compare performance of supervised systems trained with low/large amounts of data. In section 4 we introduce the sentence-level and frame-level confidence measures, while in section 5 we explore different strategies to semi-supervised training. Finally, in section 6 we summarize the observations and discuss future directions.

This work was partly supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013, and the IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

Table 1. *Data analysis, numbers of speakers, amounts of annotated speech data after resegmentation*

Dataset	FullLP	LimitedLP	dev
speakers	991	121	120
size in hours (reseg.)	84.8	10.8	9.8

2. EXPERIMENTAL SETUP

In this paper, we report experiments on the Vietnamese dataset[1] as provided within the IARPA Babel program, release babel107b-v0.7. The training data consist of a large portion of conversational telephone speech and a small part of prompted speech. For training, we used both types of data. The development set consists of conversational speech only. The data come from various telephone channels: landlines, different kinds of cellphones, or phones embedded in vehicles. The sampling rate is 8000 Hz.

Two scenarios are defined – Full Language Pack (FullLP), in which all the collected data is transcribed; and Limited Language Pack (LimitedLP), in which only a subset of the data is transcribed, while the remaining part of the FullLP data can be used as untranscribed data.

The overview of the data (i.e. numbers of speakers and amounts of speech data after resegmenting) is in Tab. 1. We generated our segmentation, using MLP-based VAD with Viterbi smoothing [18]. The segments were extended by +/- 300 milliseconds.

The Vietnamese phone set consists of 29 phonemes, which are marked with six different tones. We manually merged the under-represented phones. For the triphone-tree clustering, we introduced a "position in a word" feature, which leads to final phoneset with 350 items. We allow state sharing across phonemes.

The original lexicon provided by Appen was modified by reducing the number of pronunciation variants. The FullLP lexicon contains 6k words and the LimitedLP lexicon contains 3k words. Due to the syllabic nature of Vietnamese (syllables are considered as words), the OOV rate on the dev-set is low: 0.21% for the FullLP and 1.19% for the LimitedLP lexicons.

We used a trigram language model with Kneser-Ney smoothing built on the training transcripts, with 100k 3-grams and 200k 2-grams for FullLP, and with 12k 3-grams and 47k 2-grams for LimitedLP.

2.1. Feature extraction, auxiliary GMM-HMM system

The acoustic models (both GMM-HMM and DNNs) are trained on VTLN-warped 12-dimensional PLPs augmented by C0, pitch (F0) [19], and probability of voicing with logit transformation. The F0 is divided by mean pitch over speaker data, the non-voiced parts are bridged-over by linear interpolation. These features are then mean/variance normalized, spliced by +/- 4 frames next to the central frame and projected down to 40 dimensions using linear discriminant analysis (LDA) and single semi-tied covariance (STC) transform [20]. Moreover, speaker adaptive training (SAT) is done using a single feature-space maximum likelihood linear regression (fMLLR) [21] transform estimated per speaker. The baseline GMM-HMM system with 2300 cross-word triphone tied states and 10 Gaussians per state is used to prepare LDA+STC+fMLLR features. For the supervised data, we compute fMLLR transforms from force-alignments. For the unsupervised data we compute fMLLR from lattices by using 2 passes of decoding. The GMM-HMM system is also used to produce DNN training targets by forced-alignment to the transcription, these triphone-state targets are used for the frame-classification training. The DNN triphone tree is inherited from the baseline GMM-HMM system.

2.2. DNN-HMM training

The DNNs are trained similarly as in [15]: the fMLLR features are spliced using context of +/- 5 frames, and are shifted / rescaled in order to have zero mean and unit variance on the DNN input. For all the experiments, we used the same DNN topology: 6 hidden layers, where each hidden layer has 2048 neurons with sigmoids, 440 inputs (i.e. the context of 11 fMLLR frames) and 2300 dimensional output layer with softmax. The hidden layers of DNN are initialized with stacked Restricted Boltzmann Machines (RBMs) that are pre-trained in a greedy layer-wise fashion [8], using Contrastive Divergence algorithm (CD-1) with one step of Markov-chain Monte-Carlo sampling. The Gaussian-Bernoulli RBM is trained with an initial learning rate of 0.01 and the Bernoulli-Bernoulli RBMs with a rate of 0.4. The initial RBM weights are randomly drawn from a Gaussian $\mathcal{N}(0, 0.01)$; the hidden biases of Bernoulli units as well as the visible biases of the Gaussian units are initialized to zero, while the visible biases of the Bernoulli units are initialized as $b_v = \log(p/1-p)$, where p is the mean output of a Bernoulli unit from previous layer. During pre-training, the momentum m is linearly increased from 0.5 to 0.9 on the initial 50 hours of data, this is accompanied by a rescaling of the learning rate using $1 - m$. Also the L2 regularization is applied to the weights, with a penalty factor of 0.0002. On each layer we swipe through more than 100h of data, so with the LimitedLP set (10h) we pre-train using 10 epochs, while with the extended set (84.8h) we use 3 epochs.

After pre-training, we add the output layer with random weights drawn from $\mathcal{N}(0, 0.01)$ and zero biases, and we perform frame-classification training (we classify frames into triphone tied-states). We use mini-batch Stochastic Gradient Descent (SGD) to minimize per-frame cross-entropy between the labels and network output. The utterances and frames are

[1]Collected by Appen Butler Hill: `http://www.appenbutlerhill.com`

presented in a randomized order, the SGD uses mini-batches of 256 frames, and an exponentially decaying schedule that starts with an initial learning rate of 0.008 and halves the rate when the improvement in frame accuracy on a cross-validation set between two successive epochs falls below 0.5%. The optimization terminates when the frame accuracy increases by less than 0.1%. Cross-validation is done on held-out set which corresponds to 10% of training data. To build the seed DNN, we first run the training to the end, then we re-align using the DNN, and we train again using the new alignments, while re-using the pre-trained stack of RBMs and randomly initialized output layer.

Finally, the seed network is re-trained by sequence-discriminative training by optimizing sMBR objective. This aims to maximize expected frame accuracy of being in a correct state. The expectation is calculated over the possible state sequences represented by lattices. The reference sequences are obtained by force-alignment to transcription. For semi-supervised training, the reference sequence is replaced by the best hypothesis. We re-generate reference alignment and lattices after 1st epoch [15].

In decoding, we subtract the log-priors computed by counting states in alignments, to convert log-posteriors into log-likelihoods, which are more suitable for standard Maximum a Posteriori (MAP) decoding. The training is accelerated by computing on general-purpose graphics processing unit (GPGPU), we use single GPU programmed with CUDA. For all the experiments, we used the Kaldi toolkit [22].

3. SUPERVISED EXPERIMENTS

As described in previous section, the seed model is trained in several stages. The auxiliary GMM-HMM model is trained by mixing-up maximum likelihood training, where the last stage produces LDA+STC+fMLLR features. Then a DNN is built on top of fMLLR features by using layer-wise pre-training, 2 runs of frame-classification training and 1+4 iterations of sequence-discriminative training.

During the evaluations, we have discovered that part of the training data is incorrectly transcribed. We removed 2% of least confident sentences, which resulted in 0.4% WER improvement on the sMBR level. The sentence confidences were calculated as follows: We used forward-backward over lattices to compute per-frame state posteriors. The posterior probability of correct state as given by reference alignment is taken as per-frame confidence. The sentence confidence is the average of per-frame confidences.

We applied the above described training procedure to both the LimitedLP and FullLP conditions. By looking at Tab. 2, we see system performance at individual stages of the training. In the last stage, the LimitedLP system (that is later used as seed model) has 13.6% worse WER than the FullLP system. This large difference is not solely due to lower amount of acoustic model training data, but also due to smaller lexicon, language model trained on smaller set of transcripts, different segmentation and smaller number of triphone states. In FullLP condition the optimum was 4800 states, while for LimitedLP 2300.

Table 2. *Baseline performance of LimitedLP system trained on 10h data (2% of low-confident segments removed); upper bound FullLP performance*

Dataset [WER]	LimitedLP	FullLP
GMM (fMLLR)	69.0	58.6
DNN	63.1	50.4
DNN-sMBR	60.6	47.0

In order to get an idea how much a DNN can improve by unsupervised self-training from the seed model, we made an experiment where we treated the untranscribed data as if we knew the correct transcription. The rest of the LimitedLP system was the same as before (LM, lexicon, GMMs, fMLLR features). The WER we obtained is 57.0%, we will consider this to be the upper bound performance for the semi-supervised training.

4. CONFIDENCE MEASURES

In general, it is necessary to incorporate some form of confidence measure into the semi-supervised training. In self-training, we decode using the seed system and treat the hypothesis as a reference. If we get a number that tells us how certain the decoder was about the decoded hypothesis, we can use it to pre-select the data and remove hypotheses which are more likely to contain error. In our experiments, we use confidence measures of two levels, the sentence-level and frame-level.

The sentence-level confidence c_{sent} is calculated as the average word confidence in a sentence, eq. (1). The word confidence c_{w_i} is the posterior probability of word w_i in i-th bin of a confusion network. The word sequence and posteriors are obtained by Minimum Bayes Risk decoding [23][24], which minimize expected word error rate.

$$c_{sent} = \frac{1}{N} \sum_{i=1}^{N} c_{w_i} \tag{1}$$

The frame-level confidence c_{frame_i}, eq. (2), is extracted from lattice posteriors $\gamma(i, s)$, which express the probability of being in state s at time i. The frame confidence is the posterior value under the state from the best path $s_{i,1best}$. The posteriors are computed using forward-backward.

$$c_{frame_i} = \gamma(i, s_{i,1best}) \tag{2}$$

Both confidence measure values reside in the interval $(0, 1)$, so that they can be used either for threshold-driven data selection or training with weighted data.

5. SEMI-SUPERVISED EXPERIMENTS

The main objective of this work is to make such use of unannotated in-domain speech data, that the WER performance of DNN ASR system improves. In this section, we will search for an optimal strategy to achieve this goal.

5.1. RBM pre-training

As the first experiment, we tried to add data to the RBM pre-training; this is trivial since the Contrastive Divergence algorithm does not need any labels. As can be seen in Tab. 3,

Table 3. *Adding more data to unsupervised RBM pre-training*

Pre-training data [h]	10.8	84.8
Pre-training iterations	10	3
Fine-tuning data [h]	10.8	10.8
WER	63.8	63.8

we tried pre-training with more iterations on smaller set and less iterations on larger set, which contains both the annotated and unannotated data. In both cases, the fine-tuning (frame-classification training) was performed on the annotated dataset. However, there is no WER difference between the two systems, this is consistent with observations previously published in [25][26].

5.2. Frame-classification training (cross-entropy)

As the pre-training is not promising, we focus on frame-classification training. In the first experiments, we add whole sentences of unannotated data, according to their sentence-level confidences. Due to mini-batch SGD training, the annotated and unannotated data are mixed together. As can

Table 4. *Adding unannotated segments (sorted according to per-sentence confidence)*

Added segments	0%	50%	70%	90%	95%	100%
WER	63.1	62.4	62.1	62.1	62.1	**62.0**

be seen in Tab. 4, we get significantly better results by adding unlabeled segments. Interestingly, the WER seems to be relatively insensitive to the amount of added segments, this may indicate that the per-sentence confidence is not crucial for semi-supervised DNN training.

Due to large disproportion between the amount of annotated and unannotated data (≈1:7), we tried to include the annotated data several times to the training set. This leads to stronger focus on transcribed data during the SGD training. In Tab. 5, we see that a slight WER improvement can be achieved by including the annotated data 3x.

Table 5. *Including several copies of annotated data, while using 100% unannotated segments*

No. copies	1x	2x	3x	4x	5x
WER	62.0	62.0	**61.7**	61.8	61.9

Previously, we have observed that the semi-supervised training is insensitive to per-sentence confidence. Nevertheless, we can still think of using data selection, that will be based on per-frame confidences, this will help us to remove frames, where the decoder was uncertain. The results in

Table 6. *Dropping frames from unannotated part according to threshold on per-frame confidence, while using 100% unannotated segments and including annotated part 3x*

Threshold	0.0	0.5	0.7	0.8	0.9	0.95
Removed frm.	0%	11%	18%	23%	28%	32%
WER	61.7	61.2	**60.9**	60.9	61.0	61.0

Tab. 6 indicate that we can achieve significant WER improvement by using frame-selection. Again, the WER is relatively insensitive within the interval of high thresholds.

In the last experiment, we performed frame-weighted training. We used the per-frame confidences, i.e. the posteriors of being in the correct state, to re-scale the vectors with error derivatives that are used for backpropagation. As the gradient depends linearly on error derivative through Jacobians, scaling the derivatives is equivalent to scaling of gradients. We combine the frame-weighting and frame-selection. The results in Tab. 7 show, that with threshold 0.5,

Table 7. *Frame-weighting by a confidence, while using 100% unannotated segments, including annotated part 3x and using thresholded frame-selection*

Frame-selection threshold	WER [%]
0.0	61.3
0.5	60.9
0.7	60.9

there is a small WER improvement. However, with threshold 0.7, which corresponds to the best system, there was no WER difference, therefore frame-weighting and frame-selection do not seem to be complementary.

The overall absolute WER improvement coming from semi-supervised frame-classification training is 2.2%, from which 1.1% is caused by adding all the unannotated segments and 1.1% comes from thresholded frame-selection. This corresponds to WER recovery [2] of 36%. In the previous work [1], WER recovery was defined as the ratio of semi-supervised WER improvement and the upper-bound WER improvement, obtained by using ground-truth reference.

5.3. Sequence-discriminative training (sMBR)

So far, we have observed WER improvements from semi-supervised frame cross-entropy training, on the other hand our seed system was trained using sequence-discriminative criterion (sMBR). To outperform the seed system we need to apply sequence-discriminative training as well.

It is not clear whether the strategy based on thresholded frame-selection will be efficient also for the sMBR training. At first, we try to outperform the seed system by applying supervised sequence-discriminative training to the best self-trained DNN. As can be seen in Tab. 8, the WER difference

Table 8. *Supervised sequence-discriminative training of the best self-trained DNN*

	baseline	semi-supervised	
cross-entropy data	10.8h	84.8h	
sMBR data	10.8h	10.8h	
	WER	WER	Δ
cross-entropy training	63.1	60.9	-2.2
sMBR training	60.6	**58.8**	-1.8

slightly lowered from the cross-entropy level 2.2%, to the sMBR level 1.8%.

Our intuition is that further improvements are possible by using semi-supervised sequence-discriminative training. We tried to start from the best model, which is already trained by supervised sMBR on the annotated data. We used the model to re-generate reference sequences, per-frame confidences and the lattices with the population of other possible sequences. Then we performed experiments with semi-supervised sMBR training with confidence-based frame-selection, however so far, these experiments were not successful. We also tried to depart from the self-trained DNN without sMBR training. As expected, we observed WER improvements, but they did not outperform the best model with the supervised sMBR training. We plan to do a more detailed analysis of this interesting problem in the future.

6. CONCLUSIONS AND DISCUSSION

Our quest in this paper is to search for an optimal data-selection strategy for the semi-supervised DNN training. We performed an analysis at all the three stages of DNN training. The RBM pre-training has been found insensitive to adding more data. When experimenting with the frame-classification training, we obtained 2.2% absolute WER improvement, while for the last stage, the sequence-discriminative self-training we did not observe performance improvement.

In the experiments with the frame-classification training we converged to a self-training setup, where we use all the untranscribed segments, we select frames with confidence larger than 0.7, and we include the transcribed data 3x. The frame-selection uses per-frame confidence measure, which is extracted from lattice posteriors, where we select the posterior probabilities of states on the best path in a lattice. This confidence measure expresses how certain the decoder was at a particular frame, and we assume that low-confidence coincides with errors in hypothesis. The transcribed data are added several times in order to better balance the amount of transcribed and untranscribed data for the mini-batch SGD training.

From the machine-learning point of view, it is surprising that even the basic strategy of self-training improves WER performance of a DNN, which is inherently a discriminative model, and therefore should be more sensitive to errors in the training data than generative models.

Another interesting point is the interaction between frame-classification training and sequence-discriminative sMBR training. Although we do not use sMBR self-training, most of the improvement from frame-classification self-training is preserved even if we do a supervised sMBR training on the small transcribed dataset.

As ongoing work, we intend to investigate more into semi-supervised sequence-discriminative training, and interactions of semi-supervised acoustic modeling with language modeling. Semi-supervised training where we iteratively re-generate reference and retrain is also promising, and is likely to lead to even better results.

7. REFERENCES

[1] S. Novotney, R. M. Schwartz, and J. Z. Ma, "Unsupervised acoustic and language model training with small amounts of labelled data," in *Proc. of IEEE ICASSP*, 2009, pp. 4297–4300.

[2] S. Novotney and R. M. Schwartz, "Analysis of low-resource acoustic model self-training," in *Proc. of INTERSPEECH*, 2009, pp. 244–247.

[3] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration," in *Proc. of INTERSPEECH*, 2013.

[4] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. of NIPS*, 2004.

[5] J.-T. Huang and M. Hasegawa-Johnson, "Semi-supervised training of gaussian mixture models by conditional entropy minimization," in *Proc. of INTERSPEECH*, 2010, pp. 1353–1356.

[6] D. Yu, B. Varadarajan, L. Deng, and A. Acero, "Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy

reduction maximization criterion," *Computer Speech & Language*, vol. 24, no. 3, pp. 433–444, 2010.

[7] J. Malkin, A. Subramanya, and J. Bilmes, "On the semi-supervised learning of multi-layered perceptrons," in *Proc. of INTERSPEECH*, 2009, pp. 660–663.

[8] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[9] A. R. Mohamed, T. N. Sainath, G. E. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny, "Deep belief networks using discriminative features for phone recognition," in *Proc. of IEEE ICASSP*, 2011.

[10] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. IEEE ASRU*, December 2011, pp. 30–35.

[11] H. Bourlard and N. Morgan, *Connectionist Speech Recognition—A Hybrid Approach*. Kluwer Academic, 1994.

[12] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. of INTERSPEECH*, 2011.

[13] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Proc. of INTERSPEECH*, 2006.

[14] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proc. IEEE ICASSP*, April 2009, pp. 3761–3764.

[15] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. of INTERSPEECH 2013*, 2013.

[16] F. Grezl and M. Karafiat, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Proc. of ASRU*, 2013.

[17] R. Hsiao, T. Ng, F. Grezl, S. Tsakalidis, L. Nguyen, and R. Schwarz, "Discriminative semi-supervised training for keyword search in low resource languages," in *Proc. of ASRU*, 2013.

[18] T. Ng, B. Zhang, L. Nguyen, S. Matsoukas, X. Zhou, N. Mesgarani, K. Veselý, and P. Matejka, "Developing a speech activity detection system for the darpa rats program," in *Proc. of INTERSPEECH*, 2012.

[19] D. Talkin, "A robust algorithm for pitch tracking (rapt)," in *Speech Coding and Synthesis*. Elsevier, 1995.

[20] 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, May 1999.

[21] D. Povey and G. Saon, "Feature and model space speaker adaptation with full covariance gaussians," in *Proc. of INTERSPEECH*, 2006.

[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 *Proc. of IEEE ASRU*, 2011.

[23] H. Xu, D. Povey, L. Mangu, and J. Zhu, "An improved consensus-like method for minimum bayes risk decoding and lattice combination," in *Proc. of IEEE ICASSP*, 2010, pp. 4938–4941.

[24] P. Swietojanski, A. Ghoshal, and S. Renals, "Revisiting hybrid and gmm-hmm system combination techniques," in *Proc. of IEEE ICASSP*, 2013.

[25] D. Yu, L. Deng, and G. Dahl, "Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition," in *Proc. of NIPS*, 2010.

[26] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in dnn-based lvcsr," in *Proc. of IEEE SLT*, 2012, pp. 246–251.