Deep Auto-encoder Based Multi-task Learning Using Probabilistic Transcriptions

Amit Das[1], *Mark Hasegawa-Johnson*[1], *Karel Veselý*[2]

[1]University of Illinois, USA
[2]Brno University of Technology, Czech Republic

amitdas@illinois.edu, jhasegaw@illinois.edu, iveselyk@fit.vutbr.cz

Abstract

We examine a scenario where we have no access to native transcribers in the target language. This is typical of language communities that are under-resourced. However, turkers (online crowd workers) available in online marketplaces can serve as valuable alternative resources for providing transcripts in the target language. We assume that the turkers neither speak nor have any familiarity with the target language. Thus, they are unable to distinguish all phone pairs in the target language; their transcripts therefore specify, at best, a probability distribution called a probabilistic transcript (PT). Standard deep neural network (DNN) training using PTs do not necessarily improve error rates. Previously reported results have demonstrated some success by adopting the multi-task learning (MTL) approach. In this study, we report further improvements by introducing a deep auto-encoder based MTL. This method leverages large amounts of untranscribed data in the target language in addition to the PTs obtained from turkers. Furthermore, to encourage transfer learning in the feature space, we also examine the effect of using monophones from transcripts in well-resourced languages. We report consistent improvement in phone error rates (PER) for Swahili, Amharic, Dinka, and Mandarin.

Index Terms: cross-lingual speech recognition, probabilistic transcription, deep neural networks, multi-task learning

1. Introduction

Recent advances in speech recognition have been mostly due to the advent of deep learning algorithms such as deep neural networks (DNN), convolutional neural networks (CNNs) and recurrent neural networks (RNN). Their popularity is mostly attributed to the fact that neural networks achieve much lower error rates than Gaussian mixture models (GMMs), especially with large training corpora. However, these systems are manifest only in a few countries where languages are well-resourced. A well-resourced language is a language (e.g. English) with an abundance of resources to support development of speech technology. Those resources are usually defined in terms of 100+ hours of speech data, corresponding transcripts, pronunciation dictionaries, and language models. Among these, the most expensive and time consuming resource is the acquisition of transcripts. Primarily because of this reason, more than 99% of 6900 languages in the world are still under-resourced [1].

In [2], Hasegawa-Johnson *et al.* were able to show that automatic speech recognition (ASR) systems can be built from transcripts collected from non-native speakers who neither spoke the target language nor had any familiarity with it. This circumvents the difficult task of finding native speakers in the target language. Briefly, a single utterance in some target language L is transcribed by multiple turkers who do not speak L. Due to this, no single turker can generate the correct transcript. Instead, a collection of transcripts from multiple turkers is constructed for a single utterance in L. This collection can be represented as a confusion network. We will refer to such a network as a *probabilistic transcript* (PT). On the contrary, a native speaker will be able to generate the correct transcript which can be represented as a single sequence of labels. We will refer to this sequence as a *deterministic transcript* (DT). DTs are simply conventional transcripts that are used for building large corpora ASR systems.

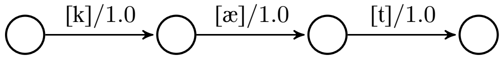

Figure 1: *A deterministic transcription (DT) for the word* cat.

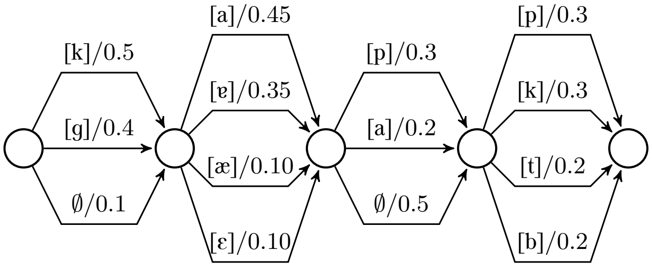

Figure 2: *A probabilistic transcription (PT) for the word* cat.

As an example, consider the DT for the word "cat" as shown in Fig. 1. Each arc represents a label and a probability value which is 1.0 always. On the other hand, a PT will look like the network in Fig. 2. The arc weight specifies the conditional probability that the phoneme was spoken, given the evidence in the transcripts. Because workers cannot distinguish all phone pairs in the utterance language, these weights are usually less than 1.0. In terms of training a DNN, the DTs are simply 1-hot alignments that are frequently observed in conventional transcripts. However, for PTs, the alignments are soft since a single frame could have multiple labels with non-zero probabilities.

2. Background

The following low resource conditions outline the nature of the data used in this study:

- PTs in target language: PTs in the target language L, written as English nonsense syllables, are collected from turkers who do not speak L.
- PTs are limited: The amount of PTs available from the turkers is limited to only 40 minutes of audio.
- Zero DTs in target language: There are no DTs in L.

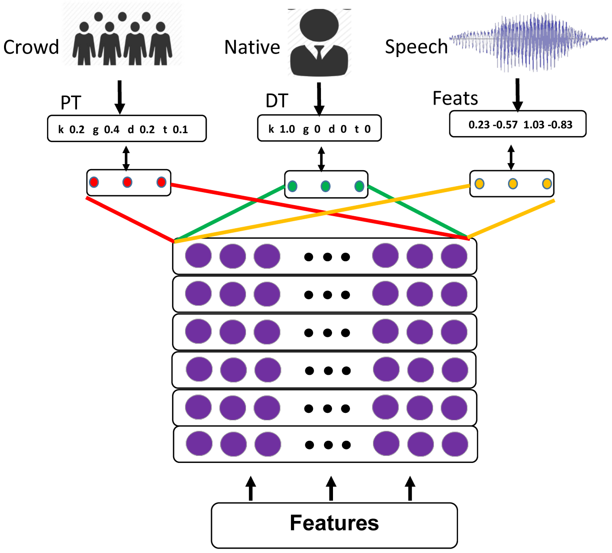

Figure 3: *Multi-task Learning for probabilistic transcripts (PT).*

- DTs only in source languages: There are DTs from 6 other source languages ($\neq L$) which are well-resourced.
- DTs are limited: There are about 40 minutes of DTs in each source language. Hence, the total amount of multilingual DTs available for training is $\approx$ 4 hours. (40 minutes/language $\times$ 6 languages).
- Unsupervised data: There are about 5.5 hours of unsupervised data in the target language.

Training an ASR system in such a scenario is challenging mainly for two reasons: (a) Zero DTs in the target language (lacking in quantity), (b) Unreliable PTs (lacking in quality). However, it is still possible to train a hidden Markov model (HMM) using PTs [3] and achieve significant improvements over a multilingual HMM that does not use PTs. Later, in [4], it was demonstrated that a DNN is better trained when integrated with the MTL framework. In this study, we show that we are able to achieve further improvements by making use of unsupervised data. More specifically, we explore using a deep auto-encoder as part of the MTL framework.

3. Deep Auto-encoder based MTL

Auto-encoders have been previously used for noise reduction using single layer networks in [5] and deeper recurrent layers in [6].

More recently, deep denoising auto-encoders in the MTL framework have been used in the problem of far-field speech recognition [7]. An auto-encoder was used to learn the mapping between the noisy signals of distant microphones and the relatively clean signals of close-talk microphones. Since the primary objective is to improve the classification performance of an ASR system, the authors in [7] integrate the auto-encoder into an MTL framework. Thus, the unified network optimizes two tasks simultaneously - the denoising task and the recognition task.

Auto-encoders have also been used to generate bottleneck features in under-resourced scenarios when little training data are available. For example, in [8], the authors train a stack of deep auto-encoders (DAEs) in a layer-wise unsupervised manner to predict clean speech from artificially corrupted noisy speech. Then a bottleneck layer, an additional hidden layer, and a final softmax layer are added to the stack of DAEs before fine-tuning the entire network using backpropagation.

More relevant to our work is the study in [9]. The authors train a neural network to recognize digits from inaccurately labeled images in the MNIST dataset. To incorporate a notion of perceptual consistency in the training, they train an auto-encoder in parallel to promote top-down consistency of model predictions with the observations. This allowed the model to discover the noise pattern in the data.

Similarly, in the PT scenario, the labels are inaccurate and hence noisy. Consequently, the backpropagated errors generated from cross-entropy training are also noisy. This will be evident from the results in Section 4.5. Briefly, the results show that DNNs trained using PTs do not necessarily outperform GMM-HMMs. This is because the label noise in PTs corrupt the hidden layer feature representations during training. Thus, in this context, we use DAE as a means to fix these errors and thus discover more useful hidden layer representations. This is illustrated in Fig. 3. Similar to [7, 9], the primary objective is to improve phone error rates (PERs) in the target language. Thus, the objective function of the MTL framework is represented by,

$$\mathcal{L}(\mathbf{W}) = \mathcal{L}_{\text{CE-PT}}(\mathbf{W}) + \lambda_{\text{DT}}\mathcal{L}_{\text{CE-DT}}(\mathbf{W}) + \lambda_{\text{DAE}}\mathcal{L}_{\text{DAE}}(\mathbf{W}) \quad (1)$$

where $\mathcal{L}_{\text{CE-PT}}$ is the cross-entropy loss of the first (primary) task where the ground truth labels (senones) are based on PTs of the target language. $\mathcal{L}_{\text{CE-DT}}(\mathbf{W})$ is the cross-entropy loss of the second task where the ground truth labels are based on DTs of the other well-resourced languages. The ground truth labels of DTs can either be context-dependent senones or context-independent phones. In the third task, the loss function $\mathcal{L}_{\text{DAE}}$ is the MSE (mean square error) between the reconstructed features and the input features. $\lambda_{\text{DT}}, \lambda_{\text{DAE}}$ are the weights of the loss functions associated with the secondary tasks in the MTL. $\mathbf{W}$ are the weights of the neural network.

4. Experiments and Results

4.1. Data

Multilingual audio files were obtained from the Special Broadcasting Service (SBS) network which publishes multilingual radio podcasts in Australia. Natively transcribed DTs available for Arabic (arb), Cantonese (yue), and Hungarian (hun) were always used as training languages. PTs (from turkers unfamiliar with the target language) were used as additional training data for the target language. We experimented with four target languages - Swahili (swh), Amharic (amh), Dinka (din), and Mandarin (cmn) - in a round-robin fashion. For example, if swh is the target language, then the training set consists of PTs in swh and DTs in the remaining six languages (arb, yue, hun, amh, din, cmn). Thus, it excludes having swh DTs in the training set.

More than 2500 turkers participated in transcribing, with roughly 30% of them claiming to know only English. The remaining turkers claimed knowing other languages such as Spanish, French, German, Japanese, and Mandarin. The utterances were limited to a length of 5 seconds. This is because the turkers did not understand the utterance language and it was easier for them to annotate short utterances than long. Since English was the most common language among the turkers, they were asked to annotate the sounds using English letters. The sequence of letters was not meant to be meaningful English words or sentences since this would be detrimental to the final performance. The important criterion was that the annotated letters represent sounds they heard from the utterances as if they were listening

Table 1: *SBS Multilingual Corpus.*

Language	Utterances		Phones
	Train	Test	
Swahili (swh)	462	123	48
Amharic (amh)	516	127	37
Dinka (din)	248	53	27
Mandarin (cmn)	467	113	52
Arabic (arb)	468	112	46
Cantonese (yue)	544	148	32
Hungarian (hun)	459	117	65
All	-	-	82

to a sequence of nonsense syllables in some exotic language. Since no turker is likely to generate the perfect transcript, each utterance was transcribed by ten turkers creating ten different transcripts per utterance. These transcripts were converted to phones and merged into a PT using [10]. Turkers were typically paid $500 per ten turkers for transcribing an hour of audio. As for DTs, the same set of utterances were transcribed by native speakers of the utterance language. The DTs in the target language were used only to know the ground truth hypotheses which are necessary for evaluating the ASR performance on the test set.

The training set consists of a) about 40 minutes of PTs in the target language and, b) about 40 minutes of DTs in other source languages which exclude the target language. The development and test sets were worth 10 minutes each. Going back to our previous example, if swh is the target language, then the training set consists of 40 minutes of PTs in swh and 40 minutes of DTs each in amh, din, cmn, arb, yue, and hun.

Finally, phone based language models (LMs) were built from the text in the target language mined from Wikipedia. The corpus is summarized in Table 1. The test utterances were selected to avoid speaker or gender bias. All experiments were conducted using the Kaldi toolkit [11].

4.2. Monolingual GMM-HMM and DNN-HMM

In the first baseline, monolingual GMM-HMM and DNN-HMM models were trained and tested using DTs in the target language. This is the oracle scenario if we assume DTs were to be available in the target language. Context-dependent GMM monolingual acoustic models were trained using 39-dimensional Mel Frequency Cepstral Coefficients (MFCC) which include the delta and acceleration coefficients. Temporal context was included by splicing 7 successive frames (current +/- 3 frames) 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) [12] was computed to transform the means of the existing model. 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 [13]. The LDA+MLLT+SAT model is the final GMM model. The forced aligned senones obtained from this model were treated as the ground truth labels for DNN training.

For DNN training, we start with greedy layer-wise Restricted Boltzmann Machines (RBMs) unsupervised pre-training since this leads to better initialization [14]. Then the DNNs were fine-tuned using supervised cross-entropy training. The monolingual PERs over a total of about 7K-8K phones are given in Table 2. This gives us an estimate about the approximate lower bound PERs thereby indicating that this is possibly the best we can achieve.

Table 2: *Phone Error Rates (PERs) of monolingual GMM-HMM and DNN-HMM models. Dev set in parentheses.*

Lang	PER (%)	
	GMM-HMM	DNN-HMM
swh	35.13 (45.78)	34.25 (39.64)
amh	51.90 (48.68)	46.69 (44.07)
din	51.56 (47.03)	48.37 (48.00)
cmn	31.80 (26.14)	28.26 (25.16)

Table 3: *Phone Error Rates (PERs) of multilingual GMM-HMM and DNN-HMM models. Dev set in parentheses.*

Lang	PER (%)		
	GMM-HMM	DNN-HMM	# Senones
swh	63.02 (66.00)	60.40 (61.62)	950
amh	68.65 (68.47)	65.56 (64.82)	1008
din	67.93 (66.79)	63.81 (65.44)	1012
cmn	69.55 (67.08)	59.50 (59.50)	985

4.3. Multilingual GMM-HMM and DNN-HMM

As mentioned earlier, we assume DTs in the target language are not available to us during training. Thus, in the second baseline, DTs from the six training languages were pooled together to train multilingual GMM-HMMs and multilingual DNN-HMMs. The training procedure was the same as the one outlined in Section 4.2. Decision tree clustering of the multilingual data resulted in about 1000 senones. The multilingual DNNs were trained using 6 hidden layers with 1024 nodes per layer and a final softmax layer with about 1000 output nodes representing the senones.

The PERs are given in Table 3. Expectedly, due to lack of DTs in the target language, the PERs are much higher than the ideal case in Table 2. Hence, the PERs in Table 3 establish the upper bound of PERs. In all subsequent experiments, we start from the upper bound of PERs in Table 3 and attempt to approach the lower bound PERs in Table 2 by including PTs during training.

4.4. PT Adapted MAP GMM-HMM

In this step, the multilingual GMM-HMM model in Section 4.3 is adapted using the PTs in the target language since DTs are not available for adaptation. The multilingual GMM-HMM acoustic model is adapted using maximum aposteriori (MAP) adaptation described in more detail in [3]. The main component in this step is that the ASR search graph, represented as a WFST mapping from the acoustic signal to a sentence, is defined by the composition $H \circ C \circ L \circ G \circ PT$ instead of the usual $H \circ C \circ L \circ G$. Here, PT is the confusion network of phones obtained from turkers as was described in Section 1.

The PER results for the MAP adapted GMM are under the column MAP-GMM in Table 4. The PER results for the multilingual DNN-HMM (column MULTI-DNN in Table 4) are replicated from Table 3 for purposes of comparison.

4.5. PT Adapted DNN-HMM

Here, we follow the conventional procedure of adapting a multilingual DNN-HMM to the target language. The softmax layer of the multilingual DNN-HMM in Section 4.3 is replaced by a randomly initialized softmax layer while retaining the shared hidden layers (SHLs) of the multilingual DNN-HMM [15]. The resulting DNN is fine tuned using the PT alignments generated by the MAP adapted model from Section 4.4. This is the con-

Table 4: *Phone Error Rates (PERs) of different ASR systems: (a) Unadapted system: Multilingual DNN (MULTI-DNN, Sec. 4.3), (b) Adapted systems: MAP adapted GMM (MAP-GMM, Sec. 4.4), Conventional DNN trained with PTs (DNN, Sec. 4.5), MTL trained with PT senones and DT senones (MTL-SS, Sec. 4.6), MTL trained with PT senones and DT monophones (MTL-SP, Sec. 4.6), MTL trained with PT senones, DT senones, and Auto-encoder (MTL-SSA, Sec. 4.7), MTL trained with PT senones, DT monophones, and Auto-encoder (MTL-SPA, Sec. 4.7), and (c) Oracle system: Monolingual DNN (MONO-DNN, Sec. 4.2) trained with DTs in the target language. The number in the parentheses is the absolute improvement of PER over the MULTI-DNN system.*

Lang	**PER (%)**							
	Unadapted	**PT Adapted**						**Oracle**
	MULTI-DNN	**MAP-GMM**	**DNN**	**MTL-SS**	**MTL-SP**	**MTL-SSA**	**MTL-SPA**	**MONO-DNN**
swh	60.40 (0.0)	45.32 (15.08)	45.89 (14.51)	44.89 (15.51)	44.51 (15.89)	44.03 (16.37)	**43.72 (16.68)**	34.25 (26.15)
amh	65.56 (0.0)	61.98 (3.58)	61.72 (3.84)	60.79 (4.77)	60.44 (5.12)	59.40 (6.16)	**59.32 (6.24)**	46.69 (18.87)
din	63.81 (0.0)	59.48 (4.33)	59.64 (4.17)	58.37 (5.44)	58.44 (5.37)	58.20 (5.61)	**57.65 (6.16)**	48.37 (15.44)
cmn	59.50 (0.0)	56.38 (3.12)	55.03 (4.47)	53.53 (5.97)	53.15 (6.35)	**51.97 (7.53)**	52.54 (6.96)	28.26 (31.24)

ventional way to adapt a DNN using DTs [16]. However, it was shown in [4] that this approach does not work very well for PTs largely due to the presence of incorrect labels in PTs.

The results are shown under the column DNN in Table 4. Clearly, the performance of DNN is worse than MAP-GMM for Swahili and Dinka and only marginally better for Amharic. On an average, the DNN system marginally outperforms the MAP-GMM system by only about 0.22% absolute.

4.6. Multi-Task Learning With Senones and Phones

To alleviate the effect of incorrect labels, an MTL system [4] using two separate softmax layers are used. The first softmax layer is trained with senones of the target language PTs whereas the second softmax layer is trained with senones of DTs in all the remaining six languages. In Table 4, the column under MTL-SS (SS = Senones in PT + Senones in DT) represents the PERs over various languages. The absolute decrease in PER compared to the conventional DNN is consistent across all languages and is in the range 1.00%-1.5%. Comparing the PT adapted MTL-SS with the unadapted MULTI-DNN, the absolute decrease in PER is in the range 4.77%-15.51%.

In this study, we also investigate MTL training using context-independent monophone targets for the DTs as opposed to using only context-dependent senones in [4]. The motivation behind this is to encourage more sharing of the feature space between PTs and DTs. It is possible that some of the contexts (senones) in the target language are unique to the target language although the center phone might be common between the target language and the well-resourced languages. The unique context discourages transfer learning. To alleviate this effect, we train the second task using monophones obtained from the DTs. The results are outlined in Table 4 under the column MTL-SP (SP = Senones in PT + Phones in DT). The net effect of using monophone targets is a slight improvement in PER by about 0.4% absolute over MTL-SS except for Dinka.

4.7. Multi-Task Learning With Deep Auto-encoder

Although MTL in Section 4.6 improves PER over the conventional DNN, it does not make use of large amounts of untranscribed audio data that are available in the target language. In this study, we make use of the DAE, as illustrated in Figure 3, as an additional task in the MTL framework. The structure of the DAE is simple. It uses the SHLs as those in the MTL framework. In addition, it has a distinct output layer which is simply an affine transform layer added on top of the final SHL of the MTL. Thus, the SHL acts as the encoder and the affine transform layer acts as the decoder. The DAE is trained using the MSE loss computed between the input features and output of the decoder.

We used about 4000 untranscribed utterances from the target language for training the DAE. First, fMLLR features were generated for the untranscribed utterances through a two-pass estimation of the fMLLR transforms. The PT adapted MAP GMM-HMM model was treated as the alignment model. Following this, the fMLLR features were spliced across +/- 5 frames and then normalized to zero mean and unit variance. In an identical fashion, the input features for all tasks in the MTL are also spliced and normalized. This helps avoid the possibility of generating large MSE errors at the DAE output. In addition, we keep the value of the weighting term λ_{DAE} in (1) to low values between 0.001-0.005.

The number of frames used to train the DAE far outnumbers the number of frames for other tasks. This would result in SGD minibatches getting biased toward the auto-encoder task. In order to maintain a balance of frames across all tasks in the minibatch, we create duplicate copies of frames for both the PT and DT tasks. For the PT tasks we used 4-6 copies and 1-2 copies for the DT tasks. The number of copies and acceptable values of λ_{DT} in (1) were found from the development set.

In Table 4, the columns under MTL-SSA (SSA = Senones in PT + Senones in DT + Autoencoder) and MTL-SPA (SPA = Senones in PT + Phones in DT + Autoencoder) report the PER results when DAE was added as an additional task to MTL-SS and MTL-SP respectively. Comparing MTL-SSA with MTL-SS, the absolute decrease in PER is about 0.17-1.56%. Similarly, comparing MTL-SPA with MTL-SP, the absolute decrease in PER is about 0.6-1.12%. The best systems for each language are highlighted in bold. Comparing the best systems with our previous work on MTL-SS [4], we observe an absolute decrease in PER in the range 1.17-1.56%. According to our knowledge, these are the best reported results for systems trained with PTs and zero DTs in all the four target languages.

Despite these improvements, the PERs are still about 10-25% (absolute) above the oracle system (MONO-DNN). Future work includes compensating label noise by interpolating PT labels with neural network predictions and estimating noisy channel (misperception) models of the non-native turkers using DNNs.

5. Conclusions

In this study, we report further improvements in training DNNs with noisy probabilistic transcripts while having no access to native transcripts in the target language. The central idea in this study is to make use of large amounts of untranscribed data to train a deep auto-encoder as an auxiliary task integrated with the multi-task learning framework. We reported PER improvements in the range 1.17-1.56% over the MTL system that does not use untranscribed data. These improvements were found to be consistent across all the four target languages.

6. References

[1] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, "Automatic speech recognition for under-resourced languages: A survey," vol. 56, pp. 85–100, Jan 2014.

[2] M. Hasegawa-Johnson, P. Jyothi, D. McCloy, M. Mirbagheri, G. Liberto, A. Das, B. Ekin, C. Liu, V. Manohar, H. Tang, E. Lalor, N. Chen, P. Hager, T. Kekona, R. Sloan, and A. K. C. Lee, "ASR for under-resourced languages from probabilistic transcription," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 1, pp. 46 – 59, 2017.

[3] C. Liu, P. Jyothi, H. Tang, V. Manohar, R. Sloan, T. Kekona, M. Hasegawa-Johnson, and S. Khudanpur, "Adapting ASR for under-resourced languages using mismatched transcriptions," in *ICASSP*, 2016, pp. 5840–5844.

[4] A. Das and M. Hasegawa-Johnson, "An investigation on training deep neural networks using probabilistic transcriptions," in *Interspeech*, 2016, pp. 3858–3862.

[5] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *ICML*, 2008, pp. 1096–1103.

[6] A. L. Mass, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Interspeech*, 2012, pp. 22–25.

[7] Y. Qian, T. Tan, and D. Yu, "An investigation into using parallel data for far-field speech recognition," in *ICASSP*, 2016, pp. 5725–5729.

[8] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked autoencoders," in *ICASSP*, 2013, pp. 3377–3381.

[9] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *arXiv:1412.6596*, 2014.

[10] P. Jyothi and M. Hasegawa-Johnson, "Transcribing continuous speech using mismatched crowdsourcing," in *Interspeech*, 2015, pp. 2774–2778.

[11] 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.

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

[13] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language.*, vol. 12, pp. 75–98, 1997.

[14] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Adv. in Neural Information Processing Systems*, 2006, pp. 153–160.

[15] 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," in *Interspeech*, 2012, pp. 2586–2589.

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