INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

An investigation on training deep neural networks using probabilistic transcriptions

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, a transfer learning technique is presented for cross-lingual speech recognition in an adverse scenario where there are no natively transcribed transcriptions in the target language. The transcriptions that are available during training are transcribed by crowd workers who neither speak nor have any familiarity with the target language. Hence, such transcriptions are likely to be inaccurate. Training a deep neural network (DNN) in such a scenario is challenging; previously reported results have described DNN error rates exceeding the error rate of an adapted Gaussian Mixture Model (GMM). This paper investigates multi-task learning techniques using deep neural networks which are suitable for this scenario. We report, for the first time, absolute improvement in phone error rates (PER) in the range 1.3-6.2% over GMMs adapted to probabilistic transcriptions. Results are reported for Swahili, Hungarian, and Mandarin.

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

1. Introduction

We explore training deep neural networks using probabilistic transcripts (PT) but no deterministic transcripts (DT) in the target language. DT means the transcript was collected from native speakers of a language. Since there is no ambiguity in such ground truth labels, the labels are deterministic in nature. The labels are then converted to IPA phone symbols. As an example the DT for the word "cat" can be represented as shown in Fig. 1 with each arc representing a symbol and a probability value. Here, each symbol occurs with probability 1.0. On the other hand, PT means that the transcript was probabilistic or ambiguous in nature. Such transcripts frequently occur, for example, when collected from crowd workers who do not speak the language they are transcribing [1]. Usually a training audio clip (in some language L) is presented to a set of crowd workers who neither speak L nor have any familiarity with it. Thus, due to their lack of knowledge about L, the labels provided by such workers are inconsistent, i.e., a given segment of speech can be transcribed by a variety of labels. This inconsistency can be modeled as a probability mass function (pmf) over the set of labels transcribed by crowd workers. Such a pmf can be graphically represented by a confusion network as shown in Fig. 2. Unlike the DT in Fig. 1 which has a single sequence of symbols, the PT has 3×4×3×4 = 144 possible sequences, one of which could be the right sequence. In this case, it is "k æ ∅ t".

Collecting and processing PTs for audio data in the target language L from crowd workers who do not understand L is called *mismatched crowdsourcing* [1]. The language L is the language we want to recognize using an automatic speech recognition (ASR) system trained using PTs. The objective of this study is to train a deep neural network using PTs in language L while transfering knowledge from DTs in other languages excluding L. An ASR system trained this way is particularly useful for low-resourced languages where it is difficult to find native transcribers in L but easy to find non-native crowd workers through online sources like Amazon's Mechanical Turk or Upwork. The following five low resource conditions outine the nature of the data used in this study:

- PTs in Target Language: PTs in the target language L are collected from crowd workers who do not speak L.
- PTs are limited: The amount of PTs available from the crowd workers is limited to only 40 minutes of audio.
- Zero DT in Target Language: There are no DTs in L.
- DTs only in Source Languages: There are DTs from 5 other languages ($\neq L$).
- DTs are limited: The DTs are worth about 40 minutes of audio per language. Hence, the total amount of multilingual DTs available for training is $\approx$ 3.3 hours. (40 minutes/language × 5 languages = 200 minutes)
- Unsupervised data in Target Language: There are at least 5 hours of unlabeled data in L.

The objective of this paper is to explore DNN techniques that can adapt using PTs. DNNs have been used in cross-lingual speech recognition either through tandem or hybrid approaches. In tandem approaches, either a) posteriors of the DNNs are Gaussianized [2, 3], or b) the outputs of an intermediate layer (bottleneck extractions) [4, 5], followed by dimensionality reduction using principal component analysis (PCA) are used as distinctive features for training GMM-HMM classifiers. In the class of hybrid approaches, a front-end GMM-HMM system generates alignments (usually shared context-dependent GMM states known as senones) which are used to train DNNs. DNNs have also been earlier used for knowledge transfer with zero labeled training data using an "open-target MLP" [6] or by adaptation using self-training and unsupervised pre-training [7]. Previously presented results [8] showed that DNNs can be adapted to PTs with resulting error rates exceeding or almost same as those of adapted GMMs. This paper is the first to report DNN adaptation to PTs with error rates consistently below those of adapted GMMs.

2. Algorithm

2.1. Mismatched crowdsourcing

We briefly review mismatched crowdsourcing which is used to post-process raw transcriptions obtained from crowd workers. A single audio file is transcribed by multiple workers since no

http://dx.doi.org/10.21437/Interspeech.2016-655

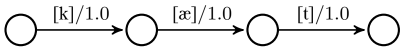

Figure 1: A deterministic transcription (DT) for the word *cat*.

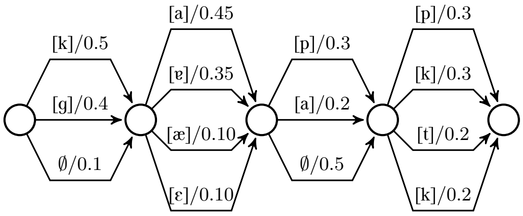

Figure 2: A probabilistic transcription (PT) for the word *cat*.

individual worker is entirely reliable. First the letters in the transcripts are converted to IPA symbols. To remove the most erroneous transcripts, each symbol in a transcript was assigned a score which is the sum of context independent agreements and context dependent agreements with other transcrips. Following this, the multiple transcripts are merged using a ROVER technique applied on equivalence classes (symbols belonging to the same class). More details of these steps are given in [1].

2.2. DNN Training using Probabilistic Transcripts

The focus of this paper is to study DNN techniques that can adapt to the target language given the five low resource conditions outlined in Section 1. At this point, assume that frame level alignments from a HMM are available as ground truth labels for DNN training. Since these are alignments based on PTs and not DTs, the ground labels are soft rather than 1-hot. From the illustrated example, the ground truth labels for a frame representing "æ" in the word "cat" could be a vector of soft labels such as $[0.35\ \text{a}, 0.45\ \text{ɐ}, 0.1\ \text{æ}, 0.1\ \text{ɛ}]$ instead of the 1-hot label $[1.0\ \text{æ}]$.

One possibility is to ignore the soft labels in PTs since they are noisy and instead use a self-training method. Here, a trained ASR system decodes the unsupervised data and then uses the confidence sampled decoded labels to retrain itself. This was earlier used in monolingual [9] and multilingual scenarios [7]. In [7], the multilingual ASR system was used to decode the unsupervised data in an unseen target language and then retrained using the decoded labels to adapt to the target language. However, this method does not leverage the available PTs.

Another possibility is to use the conventional approach to adapt a multilingual DNN to a new language. This is achieved by retaining the shared hidden layers (SHLs) [10] of an existing multilingual DNN and then replace the multilingual trained softmax layer with a new softmax layer which is fine tuned using the labels of only the target language [11]. In the current scenario, there are no DTs. Hence, an obvious step is to use the PTs to fine tune the softmax layer. This is illustrated as the DNN-1 system in Fig. 3(a). Since cross-entropy training of DNN attempts to minimize the Kullback-Leibler divergence between the distributions of ground truth labels (which are noisy for PTs) and DNN posterior outputs, the posteriors simply learn the noisy distribution of the PTs. This degrades the performance of the DNN, sometimes even worse than a GMM-HMM system, as will be reflected later in the experiments in Section 3.5. This also reaffirms the fact that DNNs do not generalize well if the training and test data are generated from two different joint distributions of acoustic data and labels. In [12], this was shown for the case when a DNN was trained using wideband data but tested on narrowband data. In our case, the training data are based on PT distributed labels whereas during test time the network outputs are compared against DT distributed labels to measure accuracy.

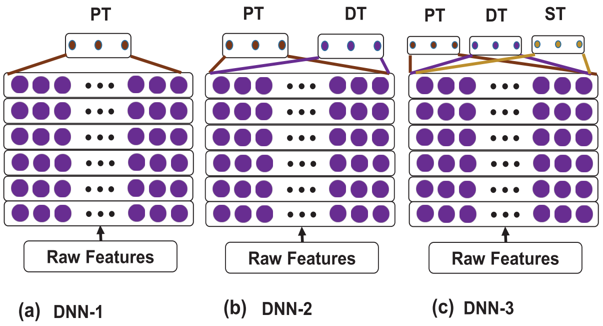

Figure 3: DNN adaptation to probabilistic transcripts (PT).

To take advantage of the PTs while at the same time alleviate the effect of noisy labels, we explore another DNN based on multi-task learning with multiple softmax layers [13]. Here, each layer is trained using a different set of transcripts. The first softmax layer is trained using PTs of the target language whereas the second layer is trained on multilingual DTs of the source languages. This is illustrated as the DNN-2 system in Fig. 3(b). There could be a third softmax layer trained using self-training transcripts (ST) generated by decoding unsupervised data in the target language. This is the DNN-3 system in Fig. 3(c). During test time, only the PT softmax layer is retained for decoding while discarding the other softmax layers. In our experiments, all the three layers had the same set of multilingual senones. The senones in the PT softmax layer could as well be adjusted only to the target language as the PT labels are monolingual.

Our motivation for using multiple softmax layers stems from encouraging results obtained in previous studies for multilingual training [14],[15], [10] and for multi-task learning [13]. In this work, our conjecture is that simultaneous training of PTs along with DTs offers multiple advantages. We intend to do more experiments to verify these advantages. a) First, the *spurious* or incorrect error gradients back propagated by the noisy PT labels fed to the PT softmax layer are partially corrected by the *true* error gradients back propagated by the high quality DT labels fed to the DT softmax layer. Therefore, due to strong supervision of highly reliable DT labels, the net result is an improved non-linear transformation learned by the SHLs and hence better feature separation. This advantage is clearly lost with the single softmax DNN-1 system trained using PTs since the training steps are inherently sequential in nature - first train using multilingual DTs and then fine tune using monolingual PTs. The noise introduced by PTs in the SHLs thus cannot be corrected. b) Since the output nodes of the DNN have one-to-one correspondence with a multilingual senone decision tree, the outputs nodes of each softmax layer represent multilingual senones and hence act as universal softmax layers. By exclusively training the PTs in the first softmax layer, we train only those softmax weights which are connected to nodes representing senones in the target language. The weights for the other senones remain untrained. This is expected to reduce the entropy of the output activation vectors. In addition, if the quality of the PTs improves, it will further lead to improved softmax weights. c) Unlike [14] where each language was assigned its own softmax layer, we assign all source languages with DTs to only one softmax layer since the primary role of DTs is to fix SHLs. This reduces the complexity of the network structure.

3. Experiments and Results

In this section, we explore the effect of the three DNNs discussed in Section 2.2 in terms of PER.

3.1. Data

Multilingual audio files were obtained from the Special Broadcasting Service (SBS) network which publishes multilingual radio podcasts in Australia. These data include over 1000 hours of speech in 68 languages. The following languages were used in our experiments - Swahili (swh), Hungarian (hun), Cantonese (yue), Mandarin (cmn), Arabic (arb), Urdu (urd). Out of these, the first three were used as target languages. The utterances were short in length (5s) which makes it easy for crowd workers to annotate the utterances since they did not understand the utterance language. Each utterance was transcribed by 10 distinct Turkers and merged using [1] to create the PTs. More than 2500 Turkers participated in these tasks, with roughly 30% of them claiming to know only English. The remaining Turkers claimed knowing other languages such as Spanish, French, German, Japanese, and Chinese.

Since English was the most common language among crowd workers, they were asked to annotate the sounds using English letters. The sequence of letters were 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 to non-sense syllables. The same set of utterances were labeled by native transcribers in the utterance language which constitute the DTs. This was required during ASR evaluations.

PTs and DTs, worth about 1 hour of audio, were collected from crowd workers and native transcribers respectively. 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. As an example, if swh is the target language to be recognized, then the training set consists of 40 minutes of PTs in swh and 200 minutes of DTs in hun, yue, cmn, arb, and urd combined.

The orthographic transcriptions for the PTs and DTs were converted to IPA based phone transcriptions. The canonical pronunciation was derived from a lexicon. If a lexicon was not available, a language specific G2P model was used. To form a set of multilingual phone symbols, dipthongs/tripthongs were split into two/three indivudal phone symbols unless they were the same as English dipthongs. Diacritics such as tones and stress markers tend to make the phone symbols unique to a particular language. Therefore, to enable phone merging across languages, such language specific diacritics were removed from the canonical phone transcriptions.

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 sufficiently shuffled so as to avoid biasing to a subset of speakers or to a specific gender.

3.2. Monolingual HMM and DNN

In the first baseline, monolingual HMM and DNN 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-HMM monolingual acoustic models 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 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) [16] 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 [17]. The LDA+MLLT+SAT model is the final HMM model that will be simply referred to as HMM in all experiments. The forced aligned senones obtained from the HMM were treated as the ground truth labels for DNN training.

Table 1: SBS Multilingual Corpus.

Language	Utterances		Phones
	Train	Test	
Swahili (swh)	463	123	53
Hungarian (hun)	459	117	70
Cantonese (yue)	544	148	37
Mandarin (cmn)	467	113	57
Arabic (arb)	468	112	51
Urdu (urd)	385	94	45
All	-	-	82

Table 2: PERs of monolingual HMM and DNN models. Dev set in parentheses.

Lang	PER (%)	
	HMM	DNN
swh	35.63 (47.00)	34.18 (39.49)
hun	38.72 (40.33)	35.62 (37.32)
cmn	31.80 (26.14)	28.26 (25.16)

Table 3: PERs of multilingual HMM and DNN models. Dev set in parentheses.

Lang	PER (%)		
	HMM	DNN	# Senones
swh	65.73 (67.58)	61.17 (63.12)	1003
hun	67.55 (68.50)	63.25 (63.65)	1012
cmn	71.09 (69.10)	64.68 (63.84)	994

Table 4: PERs of self-trained DNN models trained using STs. Dev set in parentheses.

Lang	PER %
swh	60.14 (62.07)
hun	61.05 (62.26)
cmn	63.67 (61.94)

For DNN training, we start with greedy layer-wise Restricted Boltzmann Machines (RBMs) unsupervised pretraining since this leads to better initialization [18]. Then the DNNs were fine-tuned using supervised cross-entropy training. All experiments were conducted using the Kaldi toolkit [19]. The monolingual PERs over a total of about 7K-8K phones are given in Table 2. This give us an estimate about the approximate lower bound PERs thereby indicating this is possibly the best we can achieve.

3.3. Multilingual HMM and DNN

Since the paper assumes zero DTs in the target language during training, in the second baseline, multilingual DTs were used to train HMMs and DNNs where the multilingual DTs *exclude* the DTs in the target language. The training procedure was the same as the one outlined in Section 3.2. The DNNs were trained using 6 hidden layers with 1024 nodes per layer. The total number of output nodes in the softmax layer representing multilingual senones was around 1000. The PERs are given in Table 3. Expectedly, due to lack of DTs in the target language, the PERs

Table 5: PERs of HMM, DNN-1, DNN-2, DNN-3 models trained using PTs. First element in parentheses is the PER of the dev set. Second element is the absolute improvement in PER of the test set over MAP HMM.

Lang	PER (%)			
	MAP HMM	DNN-1	DNN-2	DNN-3
swh	44.77 (50.97,0.0)	45.14 (47.83,-0.37)	**43.03 (45.87,1.74)**	43.50 (45.95, 1.27)
hun	56.85 (57.69,0.0)	56.13 (57.21,0.72)	**55.53 (56.08,1.32)**	55.69 (56.85, 1.16)
cmn	59.23 (58.05,0.0)	54.95 (54.35,4.28)	53.70 (53.94,5.53)	**53.05 (53.59, 6.18)**

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.

3.4. Self-training DNN

In this experiment, we explore a self-training algorithm [9] in which a multilingual ASR system decodes the audio in the target language and then uses the confidence selected decoded labels to retrain itself in the target language [7]. The objective of this experiment is to evaluate the efficacy of the STs (self-training transcripts) vs PTs. Since we are interested in generating STs from an ASR, we ignore the PTs from crowd workers and decode the 40 minutes of audio in the training set using the multilingual DNN from Section 3.3. The results are given in Table 4. Compared to the multilingual DNN in Table 3, the improvement due to self-training is in the range 1.01%-2.20%. We determined frame confidence thresholds as 0.5 or 0.6 from the development set.

3.5. Training one softmax DNNs using PT: DNN-1

In this experiment, we use PT labels from crowd workers to train the DNN-1 system in Fig. 3(a). In the first step, the multilingual HMM models in Section 3.3 are adapted to the PTs using MAP adaptation. Details of this step are given in [20]. The PER results for the MAP adapted HMM are given in the first column of Table 5. The absolute improvement in PER over multilingual DNN models in Table 3 is significantly higher than self-training, in the range 5.45%-16.4%. The conventional way to adapt a DNN using DTs is to retain the SHLs of the multilingual DNN, replace the existing softmax layer with a single randomly initialized layer and fine tune this new layer using 1-hot senone alignments from an HMM [11]. Here, for the case of PTs, the MAP adapted HMM generates soft alignments of the PTs which are used for fine tuning the new softmax layer. The results for DNN-1 are given in the second column of Table 5. Since the DNNs are now tuned to the target language PTs, we compare their performance with MAP adapted HMMs. The absolute improvement is in the range -0.37-4.28. Clearly, DNN-1 performed worse than MAP HMM for Swahili and the improvement is marginal for Hungarian. Thus, DNN-1 exhibits chance performance. Hence, this approach does not work very well for PTs largely due to the presence of incorrect labels in PTs.

3.6. Training two softmax DNNs using PT and DT: DNN-2

In this experiment, instead of using a single softmax layer, we use two separate softmax layers illustrated as the DNN-2 system in Fig. 3(b). The first softmax layer is trained with target language PTs only whereas the second softmax layer is trained with multilingual DTs. While training DNN-2, we find introducing an additional copy of the multilingual DTs may sometimes lead to better PERs. This was observed in the case of Hungarian. For the other two languages (Swahili and Mandarin), additional copies were not required. We determined the number of copies from the development set. The results are given in the third column of Table 5. This time the improvement in PERs over MAP HMM is consistent and significantly higher (1.32%-5.53%) than the improvement in DNN-1.

3.7. Training three softmax DNNs using PT, DT, and ST: DNN-3

In this experiment, we introduce a third softmax layer (see Fig. 3(c)) for ST labels generated from decoding additional unsupervised data in the target language (4000 utterances $\sim$ 5.5 hours) using the DNN-2 system. We use the DNN-2 to decode unsupervised data instead of the multilingual DNN since the former is better adapted to recognize the target language. Hence, the languages of PTs and STs are matched. Frames which had confidences below a threshold of 0.9 were discarded since frames above this threshold are expected to have reliable labels. To balance the effect of disproportionate amounts of data between the DTs and STs, we created multiple (2-4) copies of the frames labeled with DTs where the number of copies were determined from the development set. The PER results are presented in Table 5. The results are similar to DNN-2 with improvements in the range 1.16%-6.18% . It appears that DNN-3 is not significantly better than DNN-2 but still outperforms DNN-1. Perhaps decoding more unsupervised audio to generate more STs or adding the STs to the PTs and then retraining using the DNN-2 architecture might be useful. This is currently under investigation.

Finally, comparing the PERs in Table 4 (self-train), Table 5 (DNN adapted) with the lower and upper bound PERs listed in Table 2 (monlingual) and Table 3 (multilingual) respectively, three findings are evident. First, from Table 5, DNN-2 or DNN-3 outperform DNN-1 and MAP-HMM systems. Thus, for PTs, we recommend DNN-2/DNN-3 as reliable baselines for adapting DNN to PTs instead of the conventional adaptation in DNN-1. Second, DNN-2/DNN-3 are able to close between 28% and 67% (relative) of the gap between Table 3 and Table 2. Thus, we can say that PTs are between one and two thirds as useful as DTs. Third, PTs from crowd workers are more useful than STs generated from an ASR system (Table 5 vs Table 4).

4. Conclusions

We investigated multiple DNN training strategies to adapt DNNs to probabilistic transcripts collected from crowd workers not familiar with the target language. We demonstrated that adaptation to probabilistic transcripts using the conventional DNN-1 system is not reliable. As a result, we proposed adaptation using DNN-2 or DNN-3 systems which consistently outperform DNN-1 and HMM systems. The absolute PER improvement for 3 languages were in the range 1.3%-6.2%.

5. Acknowledgements

The work reported here was started at the JSALT 2015 workshop in the University of Washington, Seattle and was partly supported by the Johns Hopkins University via grants from Google, Microsoft, Amazon, Mitsubishi Electric, and MERL. The authors thank Paul Hager, Massachusetts Institute of Technology and Karel Veselý, Brno University of Technology for discussions.

6. References

[1] P. Jyothi and M. Hasegawa-Johnson, "Transcribing continuous speech using mismatched crowdsourcing," in *Interspeech*, 2015.

[2] 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.

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

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

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

[6] 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.

[7] K. Knill, M. J. F. Gales, A. Ragni, and S. Rath, "Language independent and unsupervised acoustic models for speech recognition and keyword spotting," in *Interspeech*, 2014.

[8] M. Hasegawa-Johnson, E. Lalor, K. Lee, P. Jyothi, D. McCloy, M. Mirbagheri, A. Das, G. D. Liberto, B. Ekin, C. Liu, V. Manohar, H. Tang, P. Hager, T. Kekona, and R. Sloan, "Probabilistic transcription: WS15 Final Report," unpublished presentation, Jelinek Speech and Language Technology Workshop, 8/14/2015.

[9] K. Vesely, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *IEEE ASRU Workshop.*, 2013, pp. 267–272.

[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] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, "Feature learning in deep neural networks - studies on speech recognition tasks," in *Int. Conf. Learn. Rep.*, 2013.

[13] M. L. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *ICASSP*, 2013, pp. 6965–6969.

[14] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Interspeech*, 2008, p. 27112714.

[15] K. Veselý, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," 2012.

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

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

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

[19] 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.

[20] 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.