LANGUAGE-INVARIANT BOTTLENECK FEATURES FROM ADVERSARIAL END-TO-END ACOUSTIC MODELS FOR LOW RESOURCE SPEECH RECOGNITION

Jiangyan Yi[1], *Jianhua Tao*[1,2,3], *Ye Bai*[1,3]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences, Beijing, China
{jiangyan.yi, jhtao, ye.bai}@nlpr.ia.ac.cn

ABSTRACT

This paper proposes to learn language-invariant bottleneck features from an adversarial end-to-end acoustic model for low resource languages. The multilingual end-to-end model is trained with a connectionist temporal classification loss function. The model has shared and private layers. The shared layers are the hidden layers utilized to learn universal features for all the languages. The private layers are the language-dependent layers used to capture language-specific features. Attention based adversarial end-to-end language identification is used to capture enough language information. Furthermore, orthogonality constraints are used to make private and shared features dissimilar. Experiments are conducted on IARPA Babel datasets. The results show that the target model trained with the proposed language-invariant bottleneck features outperforms the target model trained with the conventional multilingual bottleneck features by up to 9.7% relative word error rate reduction.

Index Terms— Language-invariant, adversarial, end-to-end, low resource, speech recognition

1. INTRODUCTION

A lot of efforts have been made to improve the performance of low resource speech recognition tasks. The bottleneck features are helpful to train acoustic models for target languages [1, 2, 3, 4].

Previously, deep neural network (DNN) based bottleneck models are used to generate multilingual bottleneck features [5, 6, 7]. Recently, Hartmann et al. [8] use bi-directional long-short term memory networks (BLSTM) and very deep convolutional neural networks to extract monolingual bottleneck features. Previous studies [9, 10, 11] have shown that the acoustic model trained using bottleneck features outperforms the model trained only with the target language, especially when the training data is limited. Nevertheless, the bottleneck features may contain some unnecessary language-specific information. Yi et al.[12] propose to transfer shared parameters via language adversarial transfer learning for the target language. Yi et al. also [13] propose to use adversarial multilingual training to extract universal bottleneck features for low resource languages. The results show that the proposed method is effective. However, this method still has some limitations. First, the language adversarial model in [13] is trained with a cross entropy loss function, but it is unclear whether the model trained with a connectionist temporal classification (CTC) [14] loss function is effective or not. Second, input features of several frames do not contain much language information. Finally, the shared and private features may have some similarities.

In order to address the above problems, this paper proposes to learn language-invariant bottleneck features from an adversarial end-to-end model. Many studies [15, 16] have shown that CTC based end-to-end acoustic models have achieved promising results. Therefore, the BLSTM model with the CTC loss function (BLSTM-CTC) is utilized to train the adversarial bottleneck model. In addition, inspired by the success of end-to-end language identification tasks [17], this paper proposes an adversarial end-to-end language identification to capture enough language information. Furthermore, inspired by the recent domain adaptation work [18], this paper employs the difference loss to encourage the shared and private extractors to encode different aspects of the inputs. The difference loss is implemented by orthogonality constraints [18]. Thus, the end-to-end bottleneck model can learn language-independent features.

The main contributions of this paper are as follows. (1) Adversarial multilingual training is employed to train CTC based end-to-end acoustic model. (2) Adversarial end-to-end language identification is proposed to capture utterance-level language information. (3) Orthogonality constraints are used to make private and shared representations dissimilar. Experiments are conducted on IARPA Babel datasets. The results show that the proposed adversarial end-to-end bottleneck acoustic model outperforms the baseline multilingual bottleneck model by up to 9.7% relative word error rate (WER) reduction.

The rest of this paper is organized as follows. Section

2 introduces the adversarial end-to-end bottleneck acoustic model. Section 3 presents experiments and results. This paper is concluded in Section 4.

2. ADVERSARIAL END-TO-END BOTTLENECK ACOUSTIC MODEL

The adversarial end-to-end bottleneck acoustic model is based on BLSTM-CTC which has an additional end-to-end language discriminator with gradient reversal layer (GRL) [19, 20]. The architecture of the model is depicted in Fig.1.

The bottleneck model has private and shared hidden layers. The shared layers are the hidden layers utilized to learn universal features for all the languages. The private layers are the language-dependent layers used to capture language-specific features. The private layers consist of two BLSTM layers. The shared layers are composed of three BLSTM layers, with the middle layer being a bottleneck (BN) layer.

The language discriminator has a fully connected (FC) hidden layer and an attention layer. The attention mechanism [17] is used to convert an utterance's features into a fixed-size real-valued vector. The GRL has no parameters, which is introduced to ensure the feature distributions over all the languages are as indistinguishable as possible for the language discriminator. Furthermore, orthogonality constraints are used to make private and shared representations dissimilar. So the shared layers can learn more language-invariant features.

2.1. Connectionist temporal classification

Connectionist temporal classification (CTC) loss function is used to select the most probable label sequences for a given input sequence [14]. Let x denote an input sequence, and z be an output sequence over the alphabet of labels. In general, each training sample in S is defined as a pair of sequences (x, z). The aim of maximum likelihood training is to minimise the following objective function:

$$\mathcal{L}_{\text{ctc}} = -\ln \prod_{(x,z)\in S} p(z|x) = -\sum_{(x,z)\in S} \ln p(z|x) \qquad (1)$$

where $(x, z) \in S$ denotes a training sample.

2.2. Multilingual training

For the m-th language, given a dataset with N_m training samples $\{x_i^{(m)}, z_i^{(m)}\}_{i=1}^{N_m}$, where $\{x_i^{(m)}, z_i^{(m)}\}$ is the i-th training sample (utterance-level), $x_i^{(m)} \in R^{k\times d}$ is a feature matrix, e.g. filterbank coefficients, k is the frame number of an utterance, d is the dimension of the features, $z_i^{(m)}$ is the corresponding labels (phones), e.g. "*y eh s y uh r r ay t*". The multilingual training is to minimize the CTC loss over all the languages:

$$\mathcal{L}_{\text{mulctc}} = -\sum_{m=1}^{M}\sum_{i=1}^{N_m} \ln p(z_i^{(m)}|x_i^{(m)}) \qquad (2)$$

where M is the total number of source languages.

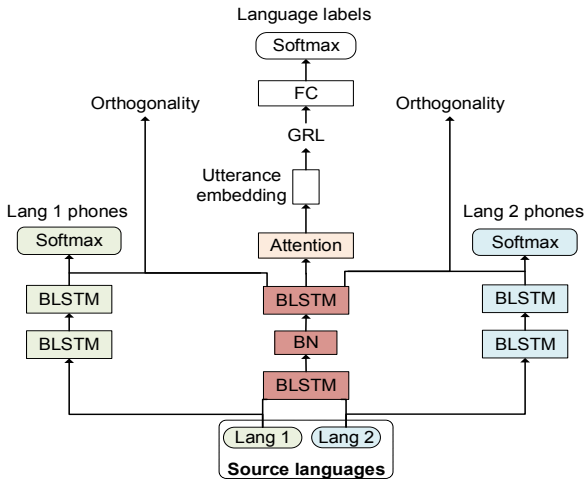

Fig. 1. The architecture of the proposed adversarial bottleneck end-to-end acoustic model.

2.3. Language adversarial training

In adversarial training procedure, a language discriminator is used to recognize the language label. Since the GRL [20] is below the language classifier, the gradients minimizing language classification errors are passed back with an opposite sign to the shared hidden layers. Thus, it ensures the feature distributions over all the languages are as indistinguishable as possible for the language discriminator. Given an additional language label for each training sample (utterance-level) $\{x_i^{(m)}, z_i^{(m)}, m\}$, where $m \in \{1, ..., M\}$ denotes the language label for each utterance. The loss function of the adversarial language discriminator is formulated as:

$$\mathcal{L}_{\text{adv}} = -\sum_{m=1}^{M}\sum_{i=1}^{N_m} \ln p(m|x_i^{(m)}) \qquad (3)$$

2.4. Orthogonality constraints

Motivated by recent work [18], the difference loss is employed to encourage the shared and private extractors to encode different aspects of the inputs. The difference loss is implemented by orthogonality constraints. Let $\mathbf{A}$ be matrices whose rows are the shared representations. Let $\mathbf{B_m}$ denotes matrices whose rows are the private representations for the

Table 1. Overall experimental data distributions.There are four source languages and three target languages.

Language	Language (Id)	Language Family	Dataset	Training (hours)	Dev (hours)	#Phones	Lexicon Size
Source	Assamese (102)	Indo-European	FLP	61	10	50	23904
	Bengali (103)	Indo-European	FLP	62	10	53	26508
	Kurmanji (205)	Indo-European	FLP	41	10	37	14411
	Lithuanian (304)	Indo-European	FLP	42	10	89	32713
Target	Pashto (104)	Indo-European	FLP	78	10	44	18745
			LLP	10	10	44	6186
	Turkish (105)	Turkic	FLP	77	10	42	41320
			LLP	10	10	42	10110
	Vietnamese (107)	Austroasiatic	FLP	88	10	68	6422
			LLP	11	10	68	3205

m-th language. The difference loss encourages orthogonality between the shared and private representations.

$$\mathcal{L}_{\text{diff}} = \sum_{m=1}^{M} \left\| \mathbf{A}^{\top} \mathbf{B}_{\mathbf{m}} \right\|_F^2 \tag{4}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm.

2.5. Improved adversarial multilingual training

The improved adversarial multilingual training is to jointly optimize the above-mentioned three loss functions. So the final loss function of the adversarial end-to-end bottleneck model is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{mulctc}} + \lambda \mathcal{L}_{\text{adv}} + \gamma \mathcal{L}_{\text{diff}} \tag{5}$$

where $\lambda \in R$ and $\gamma \in R$ are hyper-parameters.

3. EXPERIMENTS

A series of experiments are conducted on IARPA Babel datasets to evaluate our proposed method.

3.1. Datasets

Our experiments are conducted on the datasets of IARPA Babel program. The IARPA Babel datasets consist of conversational telephone speech for 28 languages collected across a variety of environments. More details can be found in our previous work [13]. Table 1 describes experimental data statistics.

We select 4 languages as the source languages: Assamese, Bengali, Kurmanji and Lithuanian. All the source languages are the full language pack (FLP), which are only used to train the source models. We also select 3 languages as the target languages: Pashto, Turkish and Vietnamese. The FLP and the limited language pack (LLP) of the target language are both used to train the target models, respectively. Each language has a *training* set and *dev* set. All the results of the target models are reported in terms of WER on 10-hours *dev* set, respectively.

3.2. Experimental setup

Our experiments are conducted using the Kaldi speech recognition toolkit [21] and TensorFlow [22]. The decoding of the target ASR systems is performed using Kaldi toolkit. The BLSTM-CTC BN models are implemented using TensorFlow. The features are extracted with a 25-ms sliding window with a 10-ms shift. Each frame is represented by 3-dimensional pitch features and 40-dimensional log mel-filter bank features plus their delta and delta-delta.

The source models are trained only using four source languages. The BLSTM models use a single frame as the input, with no frame stacking. The private layers consist of 2 BLSTM layers. The shared layers consist of 3 BLSTM layers, with the middle layer being a BN layer. Motived by the work in [23], each BLSTM layer consists of peephole connections and a recurrent projection layer. Each BLSTM layer has two directions: the forward direction and the backward direction. Each direction is a regular LSTM layer. The LSTM layer has 512 memory cells and the recurrent projection layer would project the output to 300 dimensions. Inspired by the work in [8], the LSTM based BN layer has 512 memory cells and the dimention of the recurrent projection layer is 40. The BLSTM layers are initialized to the range (-0.02, 0.02) with a uniform distribution. We use the back-propagation through time learning algorithm to compute parameter gradients. The activations of memory cells is clipped to range [-50, 50]. The activation function of the FC hidden layer of the language discriminator is rectified linear units (ReLU) [24]. The FC layer has 2048 nodes. The dropout rate is fixed at 0.1. Note that the λ is used only for updating the shared layers of the source model. However, for updating the language classification component, we use a fixed $\lambda = 1$, to ensure that the latter trains as fast as the phones classifiers [19]. γ is gradually increased from 0 to 1 as epoch increases so that the model is stably trained.

The target models are DNN based monolingual models. The Gaussian mixture model hidden Markov model is used to generate frame-level state alignments for DNN models. Input features for the DNN models use a sliding context window of

Table 2. WERs (%) results of the target models on *dev* data for LLP and FLP models.

Source Model Setting	Target Languages (LLP)			Target Languages (FLP)		
	Pashto	Turkish	Vietnamese	Pashto	Turkish	Vietnamese
Only target language	59.1	57.9	59.7	50.7	47.3	51.4
Multilingual BLSTM-CTC (Baseline)	53.8	54.2	58.0	46.7	44.9	50.5
+ language identification	53.1	53.6	57.5	46.2	44.5	50.2
+ gradient reversal layer	50.2	51.6	55.8	44.3	43.3	49.0
+ orthogonality constraint (Ours)	**48.6**	**50.1**	**54.3**	**43.4**	**42.5**	**48.0**

11 consecutive speech frames as inputs. For LLP systems, the DNN models have 5 hidden layers with 2048 nodes in each layers. For FLP systems, the DNN models have 6 hidden layers with 2048 nodes in each layer. The output labels of the model is language specific senones. The number of senones is about 3000 for each language. The DNN models are trained using stochastic gradient descent with a momentum term to minimize the cross entropy loss. The initial learning rate and momentum are set to 0.003 and 0.9, respectively. The learning rate is exponentially decayed during training. The dropout rate is fixed at 0.2.

The 3-gram language models are trained using the transcriptions of the training data for each target language. The vocabulary of the language model is the officially released vocabulary from IARPA Babel datasets. At the test stage, decoding is performed using fully composed 3-gram weighted finite state transducers.

3.3. Results

In the first group of experiments, the target model is trained only using the target data. All the models are trained on the LLP and FLP datasets, respectively. The results on the LLP and FLP datasets are listed in Table 2.

In the second group of experiments, the target model is trained using tandem features. The tandem features consist of BN features from the source model and the input features of each target languages. We train four source models. At first, the baseline source model is the conventional multilingual BLSTM-CTC end-to-end model. Then, the second source model is the BLSTM-CTC model having an additional language identification without GRL. The third source model is the BLSTM-CTC model having an additional language identification with GRL. Finally, the fourth source model is the proposed model, which utilizes orthogonality constraints on the third source model.

The results of the target models are reported in Table 2. The results show that the target model trained with BN features from the BLSTM-CTC model outperforms the target model trained only using the target data. When adding a language identification without GRL on the BLSTM-CTC model, the target model obtains improvements. In addition, the target model obtains further improvements when adding a language identification with GRL on the BLSTM-CTC model. Furthermore, the target model achieves the best performance when the adversarial BLSTM-CTC model using orthogonality constraints.

The results show that the target models trained with the proposed language-invariant bottleneck features obtain 9.7%, 7.6%, 6.4%, 7.1%, 5.3%, 5.0% relative WER reduction on Pashto, Turkish and Vietnamese LLP and FLP condition over the target models trained with the conventional multilingual bottleneck features, respectively.

The above experimental results show that the proposed method is effective. Adversarial multilingual training is effective for the CTC based end-to-end bottleneck model. The GRL and orthogonality constraints ensure that the shared layers learn language-invariant features. The target model benefits from the language-invariant features.

4. CONCLUSIONS

This paper proposes to learn language-invariant bottleneck features from an adversarial end-to-end acoustic model for low resource languages. The end-to-end model is trained with a CTC loss function. Attention based adversarial end-to-end language identification is proposed to capture more language information. Orthogonality constraints are used to make private and shared features dissimilar. Experiments are conducted on IARPA Babel datasets. The results show that the target model trained with the proposed language-invariant bottleneck features outperforms the target model trained with the conventional multilingual bottleneck features by up to 9.7% relative WER reduction. Future work includes learning language-independent features using more source languages and exploring the similarity between source and target languages.

5. ACKNOWLEDGMENTS

This work is supported by the National Key Research & Development Plan of China (No.2018YFB1005003) and the National Natural Science Foundation of China (NSFC) (No.61425017, No.61831022, No.61773379, No.61771472), and Inria-CAS Joint Research Project (No.173211KYSB20170061).

6. REFERENCES

[1] J. Cui, B. Kingsbury, B. Ramabhadran, A. Sethy, K. Audhkhasi, X. Cui, E. Kislal, L. Mangu, M. Nussbaum-Thom, and M. Picheny, "Multilingual representations for low resource speech recognition and keyword search," in *ASRU*, 2015, pp. 259–266.

[2] T. Alumae, S. Tsakalidis, and R. Schwartz, "Improved multilingual training of stacked neural network acoustic models for low resource languages," in *INTERSPEECH*, 2016, pp. 3883–3887.

[3] J. Trmal, M. Wiesner, V. Peddinti, X. Zhang, P. Ghahremani, Y. Wang, V. Manohar, H. Xu, D. Povey, and S. Khudanpur, "The kaldi openkws system: Improving low resource keyword search," in *INTERSPEECH*, 2017, pp. 3597–3601.

[4] F. Keith, W. Hartmann, M. Siu, J. Ma, and O. Kimball, "Optimizing multilingual knowledge transfer for time-delay neural networks with low-rank factorization," in *ICASSP*, 2018, pp. 4294–4298.

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

[6] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual mrasta features for low-resource keyword search and speech recognition systems," in *ICASSP*, 2014, pp. 7854–7858.

[7] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *ICASSP*, 2013, pp. 8619–8623.

[8] W. Hartmann, R. Hsiao, and S. Tsakalidis, "Alternative networks for monolingual bottleneck features," in *ICASSP*, 2017, pp. 5290–5294.

[9] H. Xu, H. Su, C. Ni, X. Xiao, H. Huang, E.S. Chng, and H. Li, "Semi-supervised and cross-lingual knowledge transfer learnings for dnn hybrid acoustic models under low-resource conditions," in *INTERSPEECH*, 2016, pp. 1315–1319.

[10] M. Karafiat, M.K. Baskar, P. Matjka, K. Vesely, F. Grzl, L. Burget, and J. Aernocky, "2016 but babel system: Multilingual blstm acoustic model with i-vector based adaptation," in *INTERSPEECH*, 2017, pp. 719–723.

[11] T. Alumae, D. Karakos, W. Hartmann, R. Hsiao, Zh. Le, N. Long, S. Tsakalidis, and R. Schwartz, "The 2016 bbn georgian telephone speech keyword spotting system," in *ICASSP*, 2017, pp. 5755–5759.

[12] J. Yi, J. Tao, Z. Wen, and Y. Bai, "Language-adversarial transfer learning for low-resource speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 3, pp. 621–630, 2019.

[13] J. Yi, J. Tao, Z. Wen, and Y. Bai, "Adversarial multilingual training for low-resource speech recognition," in *ICASSP*, 2018, pp. 4899–4903.

[14] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," pp. 6645–6649, 2013.

[15] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *ASRU*, 2016, pp. 167–174.

[16] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*, 2017, pp. 4835–4839.

[17] W. Geng, W. Wang, Y. Zhao, X. Cai, and B. Xu, "End-to-end language identification using attention-based recurrent neural networks," in *INTERSPEECH*, 2016, pp. 2944–2948.

[18] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *NIPS*, 2016.

[19] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[20] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015, pp. 1180–1189.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlaek, Y. Qian, and P. Schwarz, "The kaldi speech recognition toolkit," in *ASRU*, 2011.

[22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "Tensorflow: a system for large-scale machine learning," 2016.

[23] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *INTERSPEECH*, 2014, pp. 338–342.

[24] M. Andrew, H. Lempitsky, and Ng. Lempitsky, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013.