Language-Adversarial Transfer Learning for Low-Resource Speech Recognition

Jiangyan Yi, *Member, IEEE*, Jianhua Tao, *Senior Member, IEEE*, Zhengqi Wen, *Member, IEEE*, and Ye Bai, *Student Member, IEEE*

***Abstract*—The acoustic model trained using the knowledge from the shared hidden layer (SHL) model outperforms the model trained only by using the target language, especially under low resource conditions. However, the shared features may contain some unnecessary language dependent information. It will degrade the performance of the target model. Therefore, this paper proposes language-adversarial transfer learning to alleviate this problem. Adversarial learning is used to ensure that the shared layers of the SHL-model can learn more language invariant features. Experiments are conducted on IARPA Babel datasets. The results show that the target model trained using the knowledge transferred from the adversarial SHL-model achieves up to 10.1% relative word error rate reduction when compared with the target model trained using the knowledge transferred from the SHL-model.**

***Index Terms*—Adversarial training, transfer learning, cross-lingual, low-resource, speech recognition.**

I. Introduction

DEEP neural networks (DNN) based acoustic models have obtained significant improvement for automatic speech recognition (ASR) systems [1]–[4]. However it is still challenging to rapidly build an ASR system for a novel language with significantly less labeled training data [5]–[7]. This was also the goal of IARPA Babel program. Without any question, data collection and annotation are very time-consuming and expensive. Therefore, how to effectively use an available larger set of languages to improve the performance of the novel language is very important.

It is easy for human beings to transfer knowledge from other languages when learning a new language [8]. Human beings not only share the same vocal tract architecture, but also use the universal phonetic systems of different languages. Similarly, acoustic models are able to share language invariant low-level components across various languages [9]–[11]. An acoustic model trained using other source languages is referred to as a source model. An acoustic model trained using a novel target language is called a target model. Multilingual training is an effective technique to train the source model [14]–[17]. This approach benefits from multi-task learning [18]. The source model is trained jointly on several languages [19], [20]. In addition, language identification based multilingual training is proposed to extract multilingual bottleneck features for low resource speech recognition [21], [22]. The knowledge from the source model can be transferred to the target model via transfer learning. The goal of transfer learning [12], [13] is to improve the performance of the target model via using knowledge from the source model. The transfer learning methods can be roughly classified into two categories: transferring bottleneck features [17], [23]–[26] and transferring model parameters [10], [11], [27]. This paper focuses on the latter.

The basic idea of transferring model parameters is that the source model is trained on the source languages, and the trained parameters are used to initialize the target model for the novel language. Previously, shared feature representations for low-resource languages have been studied by Thomas [28]. Scanzio *et al.* [29] present a front-end consisting of an artificial neural network architecture trained with multilingual data. The proposed network is called a shared hidden layer model (SHL-Model). Huang *et al.* propose to use DNN based SHL-Model [30] to transfer model parameters for unseen languages. All the hidden layers of the SHL-Model are shared across multiple languages. The softmax layers of the SHL-Model are language dependent. Recently, Xu *et al.* [31] combine this method and semi-supervised learning to transfer cross-lingual knowledge to a target model. More recently in [6], Karafiat *et al.* use bi-directional long-short term memory (BLSTM) based source model to transfer shared parameters. The above-mentioned SHL-Models only use softmax layers to learn language specific features. However, a lot of variants of SHL-Models are proposed to use hidden layers and softmax layers to learn language dependent information [23], [32]. These models have language specific hidden layers prior to softmax layers.

The results show that the target model trained using the knowledge transferred from the SHL-Model as shown in Fig. 1 performs better than the model trained only using the target training data, especially when the amount of labeled data of the target language is limited. However, the shared hidden layers may learn some unnecessary language specific information. It will degrade the performance of the target model.

Manuscript received December 19, 2017; revised June 19, 2018 and September 25, 2018; accepted December 16, 2018. Date of publication December 24, 2018; date of current version January 25, 2019. This work was supported in part by the National Key Research and Development Plan of China under Grant 2017YFC0820602, in part by the National Natural Science Foundation of China (NSFC) under Grants 61425017, 61773379, 61603390, and 61771472, and in part by Inria-CAS Joint Research Project under Grant 173211KYSB20170061. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tara N. Sainath. *(Corresponding author: Jianhua Tao.)*

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: jiangyan.yi@nlpr.ia.ac.cn; jhtao@nlpr.ia.ac.cn; zqwen@nlpr.ia.ac.cn; baiye2016@ia.ac.cn).

Digital Object Identifier 10.1109/TASLP.2018.2889606

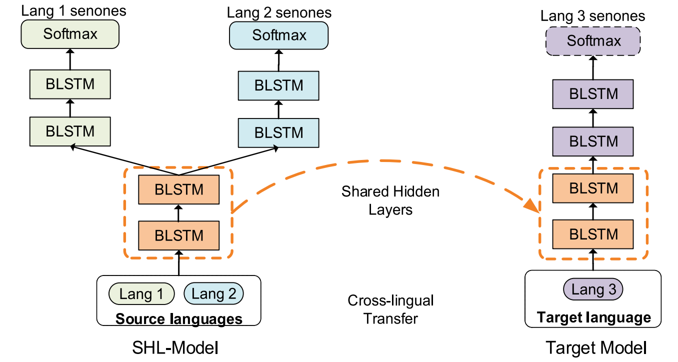


Fig. 1. The architecture of the conventional cross-lingual knowledge transfer learning method. The left model is the shared hidden layer model (SHL-Model), which is referred to as the source model. The right model is the target model. The shared parameters of the SHL-Model are transferred to the target model. Each language has its own hidden layers and softmax layer. The labels of the softmax layer are language specific senones (tied triphone states).

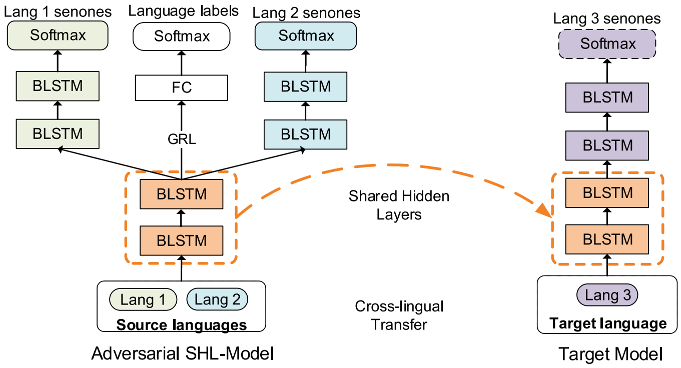


Fig. 2. The architecture of the proposed language-adversarial transfer learning method. The left model is the adversarial SHL-Model, which is referred to as the source model. The right model is the target model. The adversarial SHL-Model denotes that the SHL-Model having an additional language discriminator. FC denotes the fully connected layer. The gradient reversal layer (GRL) is introduced to ensure the feature distributions over all the languages are as indistinguishable as possible for the language discriminator. The output labels of the language discriminator are language labels.

Therefore, this paper proposes language-adversarial transfer learning to alleviate this problem. Inspired by the success of adversarial learning on domain adaptation [33], adversarial learning [34] is used to ensure that the shared layers of the source model can learn more language invariant features as shown in Fig. 2.

Recently, adversarial learning of various neural networks has attracted attention in many tasks. Ganin *et al.* [33] and Tzeng *et al.* [35] use adversarial strategy for domain adaptation in image classification tasks. More recently, Chen *et al.* [36] utilize adversarial learning for Chinese word segmentation on various heterogeneous annotation data. Zhang *et al.* [37] use adversarial strategy to obtain bilingual lexicon without cross-lingual knowledge. Shinohara [38] utilizes adversarial training to perform environment adaptation for robust speech recognition. Saon *et al.* [3] use adversarial learning for speaker adaptation in speech recognition systems. The results show that all these methods achieve state-of-the-art performance. However, this paper uses adversarial learning to train multilingual acoustic models which are treated as source models. The shared parameters of the source model are used to initialize the target model. There has been no work, to the best of our knowledge, that uses adversarial strategy and transfer learning to improve the performance of low-resource speech recognition systems.

The main contribution of this paper is that language adversarial training is used to force the shared layers of the SHL-Model to learn language invariant features. Experiments are conducted on IARPA Babel datasets. The results show that the target model trained using the knowledge transferred from the adversarial SHL-Model obtains improvement by up to 10.1% relative word error rate (WER) reduction over the target model trained using the knowledge transferred from the SHL-Model.

The rest of this paper is organized as follows. Section II briefly reviews conventional cross-lingual knowledge transfer learning method. The proposed language-adversarial transfer learning method is described in Section III. Experiments are presented in Section IV. Section V discusses the results. This paper is concluded in Section VI.

II. Review of Conventional Cross-Lingual Knowledge Transfer Learning Method

This section briefly introduces the conventional shared hidden layer model (SHL-Model) and the transfer learning method.

A. Shared Hidden Layer Model

The SHL-Model is widely used for multilingual tasks [29]–[31]. A lot of variants of SHL-Models are proposed to train multilingual models [10], [23], [27], [32]. The SHL-Model is composed of shared hidden layers and language dependent layers. The shared hidden layers and the language dependent layers are jointly optimized using a multilingual training set. The shared layers can be treated as a universal feature transformation that works well for other novel languages. One kind of the SHL-Models only uses the softmax layer to learn language specific features [30]. The other kind of SHL-Models is proposed to use hidden layers to learn more language dependent information [23], [32]. These models have language specific hidden layers prior to softmax layers. The latter SHL-Model is used to train multilingual models in this paper. The architecture of the SHL-Model is depicted in the left of Fig. 1. The SHL-Model consists of 2 shared BLSTM hidden layers and 2 language specific BLSTM hidden layers prior to softmax layers. The labels of the softmax layer are language specific senones (tied triphone states).

B. Multilingual Training

Multilingual training is an instance of multi-task learning [18]. The source model is trained simultaneously on the training data of multiple languages. Each language has its own softmax layer to estimate the posterior probabilities of language specific senones.

For the m-th language, given a dataset with N_m training samples $\{x_i^{(m)}, y_i^{(m)}\}_{i=1}^{N_m}$, where $\{x_i^{(m)}, y_i^{(m)}\}$ is the i-th training sample (frame-level), $x_i^{(m)} \in R^d$ is a feature vector, e.g., filterbank coefficients, d is the dimension of the feature vector, $y_i^{(m)} \in \{1, \ldots, C_y^{(m)}\}$ is the corresponding label (senone) for the feature vector $x_i^{(m)}$, $C_y^{(m)}$ is the total number of senones. The multilingual model is trained to minimize the cross-entropy for all the languages. So the loss function of the SHL-Model is defined as:

$$L_{Mul}(\theta^m, \theta^s) = -\sum_{m=1}^{M} \sum_{i=1}^{N_m} logP(y_i^{(m)}|x_i^{(m)}; \theta^m, \theta^s) \quad (1)$$

where m denotes the index of the m-th language, θ^m denotes the parameters of the language specific layers for the m-th language, θ^s denotes the parameters of the shared layers for all the languages, M is the total number of the source languages.

$P(y_i^{(m)}|x_i^{(m)}; \theta^m, \theta^s)$ is computed with a parametric classifier, such as a BLSTM based model with a set of trainable weights and biases. Stochastic gradient descent (SGD) is commonly used to optimize the parameters. Specifically, its gradient w.r.t the parameters are calculated using back-propagation through time (BPTT) and the parameters are updated as:

$$\theta^m \leftarrow \theta^m - \alpha \frac{\partial L_{Mul}}{\partial \theta^m} \quad (2)$$

$$\theta^s \leftarrow \theta^s - \alpha \frac{\partial L_{Mul}}{\partial \theta^s} \quad (3)$$

where $\alpha \in R$ is the learning rate. The update procedure is repeated util convergence.

C. Transferring Model Parameters

The shared layers of the SHL-Model are transferable to an unseen target language [30]. An acoustic model trained for the unseen target language is called a target model as shown in the right of Fig. 1. The target model consists of shared layers of the SHL-Model and target language specific layers.

III. Proposed Language-Adversarial Transfer Learning Method

The SHL-Model divides the feature space into shared and private spaces. However, the shared spaces may contain some unnecessary language dependent features. A good representation for cross-lingual knowledge transfer is one for which an algorithm can not learn to identify the language origin of the input observation [30], [33]. Thus, we jointly optimize the shared layers of the SHL-Model model via adversarial training. An adversarial loss is used to prevent the shared space from containing language specific features. This training strategy is called language-adversarial training. It is to find a representation of the samples where all the languages are as indistinguishable as possible. Therefore, the adversarial SHL-Model is proposed to realize this idea.

A. Adversarial Shared Hidden Layer Model

The adversarial SHL-Model is the SHL-Model which has an additional adversarial language discriminator with the gradient reversal layer (GRL) [33], [39]. The adversarial SHL-Model is depicted in the left of Fig. 2.

The language discriminator is used to recognize the language label of each frame using the shared features. The outputs of the shared layers are the inputs of the language discriminator through GRL. The language discriminator is implemented as a fully connected (FC) neural network with a single hidden layer. The activation function of the hidden layer is rectified linear units (ReLU) [40].

The GRL is introduced to ensure that the feature distributions over all the languages are as indistinguishable as possible for the language discriminator. So the adversarial SHL-Model is to learn a representation that can generalize well from one language to another. They ensure that the internal representation of the shared layers contains no discriminative information about the origin of the input. Thus the shared layers can learn more language invariant features. The language invariant features will be helpful for the target language.

B. Adversarial Training

In adversarial training procedure, a language discriminator is used to recognize the language label. Since the GRL 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 $\{x_i^{(m)}, y_i^{(m)}, m\}$, where $m \in \{1, \ldots, M\}$ denotes the language label for each frame, and M is the total number of language labels. The loss function of the language discriminator is formulated as:

$$L_{Adv}(\theta^a, \theta^s) = -\sum_{m=1}^{M} \sum_{i=1}^{N_m} logP(m|x_i^{(m)}; \theta^a, \theta^s) \quad (4)$$

where θ^a denotes the parameters of the FC and softmax layer of the language discriminator, θ^s denotes the parameters of the shared layers.

Although the language classifier is optimized to minimize the language classification error, the gradient of the language classifier is negative so that the bottom shared layers are trained to be language independent. Therefore, the parameters of the language classifier are updated as:

$$\theta^a \leftarrow \theta^a - \alpha \frac{\partial L_{Adv}}{\partial \theta^a} \quad (5)$$

$$\theta^s \leftarrow \theta^s + \alpha \frac{\partial L_{Adv}}{\partial \theta^s} \quad (6)$$

where $\alpha \in R$ is the learning rate.

C. Language-Adversarial Training

The language-adversarial training is to jointly optimize the two loss functions $L_{Mul}(\theta^m, \theta^s)$ and $L_{Adv}(\theta^a, \theta^s)$. Unlike the standard multilingual training where the shared representation is trained to maximize the classification accuracies of the primary and other languages, the parameters of the shared layers are optimized in order to minimize the loss of the senone classifiers and maximize the loss of the language discriminator. However, the latter works adversarially to the language discriminator by GRL. Thus it encourages language invariant features to emerge in the course of the optimization. So the shared features become senone discriminative and language invariant. The improved language invariance leads to the improved performance of the target language. So the loss function of the adversarial SHL-Model is defined as:

$$L(\theta^m, \theta^a, \theta^s) = L_{Mul}(\theta^m, \theta^s) + \lambda L_{Adv}(\theta^a, \theta^s) \tag{7}$$

where $\lambda \in R$ is the loss weight.

The GRL has no parameters associated with it. At the feed-forward stage, the GRL acts as an identity transformation. During the back-propagation, however, the GRL takes the gradient from the subsequent level and changes its sign, i.e., multiplying by -1, before passing it to the preceding layer. In other words, the GRL reverses the gradient (multiplies $-\lambda$). So the parameters are updated as:

$$\theta^m \leftarrow \theta^m - \alpha \frac{\partial L_{Mul}}{\partial \theta^m} \tag{8}$$

$$\theta^a \leftarrow \theta^a - \alpha\lambda \frac{\partial L_{Adv}}{\partial \theta^a} \tag{9}$$

$$\theta^s \leftarrow \theta^s - \alpha \left(\frac{\partial L_{Mul}}{\partial \theta^s} - \lambda \frac{\partial L_{Adv}}{\partial \theta^s} \right) \tag{10}$$

where λ is gradually increased from 0 to 1 as epoch increases so that the model is stably trained [33].

D. Cross-Lingual Knowledge Transfer Learning

Cross-lingual knowledge transfer learning is a special case of transfer learning. The knowledge is referred to as shared model parameters in this paper. With the help of adversarial learning, the shared layers can learn language invariant features easily. The language invariant parameters can be viewed as off-the-shelf knowledge used for the unseen new languages.

There are several methods proposed to transfer shared layers to the target model. One kind of the methods is widely used in previous work [10], [27], [30]. This method is to initialize all the hidden layers of the target model using the shared layers. The softmax layer of the target model is randomly initialized. The other kind of methods is to initialize part of the hidden layers of the target model using the shared layers. Another part of the hidden layers and softmax layer are target language dependent. The share hidden layers can be sequence or parallel with target language specific hidden layers [41], [47]. In this paper, the target model consists of share hidden layers and target language specific hidden layers as shown in the right of Fig. 2. The output of the shared layers is the input of the target language specific hidden layers. The target model is fine-tuned using the standard BPTT algorithm.

IV. Experiments

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

A. 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. The speech is collected in real-life scenarios and recorded under different conditions, such as mobile phone conversation made on the street. Most of the languages contain a small amount of data collected using a distant microphone. The total amount of transcribed audio data varies depending on the language and condition.

Only 15 languages from the Babel datasets are available for us. Therefore, we select 12 languages as the source languages. 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, Vietnamese. The FLP and the limited language pack (LLP) of the target language are both used to train the target models, respectively. Table I describes experimental data statistics.

Each language has a *training* set and *dev* set. The training sets of in-languages are available for the target language. The parameters of all the models are updated on the *training* set. The *dev* set is used to adjust hyper-parameters and select models. All the results of the target models are reported in terms of WER on 10-hours *dev* set, respectively.

B. Experimental Setup

Our experiments are conducted using the Kaldi speech recognition toolkit [32] and the open source deep learning framework called TensorFlow [42]. The Gaussian mixture model hidden Markov models (GMM-HMM) are trained using the Kaldi toolkit. The decoding of the ASR systems is also performed using Kaldi toolkit. The BLSTM based models are trained using TensorFlow.

The features are extracted with a 25-ms sliding window with a 10-ms shift. Input features for the GMM-HMM based models consist of 3-dimensional pitch features and 13-dimensional MFCC and their delta and delta-delta. We follow the officially released Kaldi recipe to build GMM-HMM based models for each language. The GMM-HMM based models are used to generate frame-level state alignments for BLSTM based models. The tied triphone states are called senones. The last column of Table I reports the number of senones for each language.

Language classification usually requires long-term context compared to the ASR task. Various ASR efforts in the last couple of years have shown improved performance with i-vector features in addition to the acoustic features in the ASR modeling [3], [6]. The i-vector approach is also successfully applied to language recognition [43]. The approach provides an elegant way of reducing high-dimensional sequential input data to a low-dimensional fixed-length feature vector while retaining most of the relevant information. Given that the i-vector features carry language information, we use i-vector features to capture long-term context for language identification tasks. There are 15 languages used to train i-vector extractors. We use 19-dimensional MFCC coefficients with energy and their delta and double delta coefficients which results in 60-dimensional

TABLE I
OVERALL EXPERIMENTAL DATA DISTRIBUTIONS. THERE ARE 12 SOURCE LANGUAGES AND 3 TARGET LANGUAGES

Language Set	Language (Id)	Language Family	Dataset	Training (hours)	Dev (hours)	#Phones	Lexicon Size	#Senones
Source Languages	Assamese (102)	Indo-European	FLP	61	10	50	23904	4362
	Bengali (103)	Indo-European	FLP	62	10	53	26508	4560
	Kurmanji (205)	Indo-European	FLP	41	10	37	14411	4306
	Lithuanian (304)	Indo-European	FLP	42	10	89	32713	4489
	Tamil (204)	Dravidian	FLP	69	10	34	58470	4810
	Telugu (303)	Dravidian	FLP	41	10	50	6306	4281
	Haitian (201)	Creole	FLP	32	10	67	14017	4157
	Tok Pisin (207)	Creole	FLP	39	10	37	2103	4201
	Zulu (206)	Niger-Congo	FLP	62	10	47	60608	4758
	Kazakh (302)	Turkic	FLP	39	10	61	6062	4258
	Georgian (404)	Kartvelian	FLP	50	10	33	35244	4657
	Lao (203)	Tai-Kadai	FLP	66	10	43	6340	4587
Target Languages	Pashto (104)	Indo-European	FLP	78	10	44	18745	4787
			LLP	10	10	44	6186	3225
	Turkish (105)	Turkic	FLP	77	10	42	41320	4761
			LLP	10	10	42	10110	3176
	Vietnamese (107)	Austroasiatic	FLP	88	10	68	6422	4969
			LLP	11	10	68	3205	3126

feature vectors. More details on i-vector extraction can be found in [43]. The results are reported with 100-dimensional i-vectors in this paper.

All the BLSTM models use a single frame as the input, with no frame stacking. The BLSTM acoustic models are based on the work in [44], where 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 320 memory cells and the recurrent projection layer would project the output to 160 dimensions.

The BLSTM layers are initialized to the range $(-0.02, 0.02)$ with a uniform distribution. We use the BPTT learning algorithm to compute parameter gradients. Each update is based on 20 time-steps of recurrent forward-propagations and back-propagations. Apart from clipping the activations of memory cells to range $[-50, 50]$, we do not limit the activations of other units, the weights or the estimated gradients. The training terminates, if only a little improvement between two epochs has been observed.

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

C. Target Models Trained Only Using One Target Language

In this section, the target model is trained only using the training data of the target language. The cross-lingual knowledge transferred from the source model is not used to train the target model.

The target model is BLSTM based monolingual model. The BLSTM model consists of 4 hidden layers, which is called BLSM-4L model. Each BLSTM layer consists of peephole connections and a recurrent projection layer. Each direction has 320 memory cells and the recurrent projection layer would project the output to 160 dimensions. The output labels are language specific senones. The number of the senones for each language is listed in Table I. The BLSTM models are trained using SGD with a momentum term to minimize the cross-entropy criterion. 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 method is applied to regularize the target model. The dropout rate is fixed at 0.5. All the models are trained on the LLP and FLP datasets, respectively.

In the first group of experiments, we only use 3-dimensional pitch and 40-dimensional log mel-filter bank (Fbank) features plus their delta and delta-delta parameters as input features to train the BLSTM models. The results of the three target monolingual models on the *dev* data set are listed in Table II.

In the second group of experiments, we use 100-dimensional i-vectors features in addition to the above-mentioned acoustic features to train the three target models. The results of the target models on the *dev* data set are listed in Table IV.

D. Target Models Trained With Knowledge From SHL-Model

In this section, we conduct a series of experiments to evaluate the performance of the target models trained with cross-lingual knowledge from the SHL-Model. We select 4 languages from the Babel datasets as the source languages, which are composed of Assamese, Bengali, Kurmanji and Lithuanian. The source languages belong to the same language family (Indo-European). All the source languages are the FLP datasets, which are only used to train the source models.

The architecture of the SHL-Model is shown in the left of Fig. 1. It has language specific hidden layers. It consists of 2 shared BLSTM hidden layers and 2 language specific BLSTM hidden layers prior to softmax layers. Each BLSTM layer consists of peephole connections and a recurrent projection layer. Each direction has 320 memory cells and the recurrent projection layer would project the output to 160 dimensions. The output labels are language specific senones. The number of the senones for each source languages is listed in Table I.

The SHL-Model is trained using SGD with a momentum term to minimize the cross-entropy criterion. The initial learning rate

TABLE II
WERs (%) of the Target Models Trained Using Shared Parameters From Various Source Models Trained Using 4 Source Languages (Assamese, Bengali, Kurmanji and Lithuanian)

Source Model	Source Model Setting	Target Languages (LLP)			Target Languages (FLP)		
		Pashto	Turkish	Vietnamese	Pashto	Turkish	Vietnamese
None	only target language	56.8	54.2	57.4	47.8	45.7	48.6
SHL-Model	with language specific hidden layers	52.9	51.9	56.4	45.2	44.1	48.2
LID-SHL-Model	+ language identification (LID)	52.3	51.5	56.1	44.9	43.2	48.0
Adversarial SHL-Model	+ gradient reversal layer (GRL)	49.1	49.5	55.2	43.5	42.6	47.5

and momentum are set to 0.002 and 0.9, respectively. The learning rate is exponentially decayed during training. The dropout rate is fixed at 0.1. We use the BPTT learning algorithm to compute parameter gradients. The training terminates, if only a little improvement between two epochs has been observed.

The shared hidden layers of SHL-Model are transferred to the target models. The target model consists of 2 transferred layers and 2 language specific BLSTM hidden layers. The configuration of each BLSTM layer is identical to the BLSTM layer of monolingual BLSTM-4L model. The output units of the softmax layer are listed in Table I. There are two fine-tuning methods for the target model: *Private* and *Overall*.

Private: At first, the BLSTM layers are initialized to the range $(-0.02, 0.02)$ with a uniform distribution. The softmax layers are randomly initialized. The parameters of the transferred layers are fixed. Then, only the parameters of the private layers are fine-tuned using the training data of the target language. The private layers consist of the language specific BLSTM layers and one softmax layer.

Overall: At first, the BLSTM layers are initialized to the range $(-0.02, 0.02)$ with a uniform distribution. The softmax layers are randomly initialized.Then, all the layers are fine-tuned using the training data of the target language.

The target model is fine-tuned using the standard BPTT algorithm on target training data. The dropout rate is fixed at 0.5. When the target model is trained using the LLP dataset, the initial learning rate is set to 0.0005. The initial learning rate is set to 0.001 when the target model is trained using the FLP dataset. The learning rate is exponentially decayed during training.

Experimental results show that the performance of the target model trained using the *Private* fine-tuning method outperforms the target model trained using the *Overall* fine-tuning method on both LLP and FLP datasets. Therefore, we only report the results using *Private* fine-tuning method. The results of the target models are reported in Table II. The results show that all the target models trained with the knowledge transferred from the SHL-Model outperform the monolingual BLSTM-4L models.

E. Target Models Trained With Knowledge From Adversarial SHL-Model

In this section, a series of experiments are performed to evaluate the performance of the target models trained with cross-lingual knowledge from the adversarial SHL-Model. We also select 4 languages from the Babel datasets as the source languages, which are composed of Assamese, Bengali, Kurmanji and Lithuanian. The source languages belong to the same language family (Indo-European). All the source languages are the FLP datasets, which are only used to train the source models.

The architecture of the adversarial SHL-Models is shown in the left of Fig. 2. The network configuration of the adversarial SHL-Model is similar to the SHL-Model. The only difference is that the adversarial model has an additional language discriminator with GRL.

We also train another source model to compare with the adversarial SHL-Model. This source model is the SHL-Model having an additional language identification without GRL, which is called the LID-SHL-Model. The LID-SHL-Model is trained using the conventional SGD without adversarial loss.

The language discriminator has one FC layer and a softmax layer. The activation function of the FC layer is ReLU. The FC layer has 2048 nodes. The softmax layer has 4 language labels. The GRL has no parameters.

The loss weight λ is initiated at 0 and is gradually changed to 1 using the following formula [33]:

$$\lambda = \frac{2}{1 + exp(-\gamma \cdot p)} - 1 \tag{11}$$

where p is the training progress linearly changing from 0 to 1, γ is set to 10 in all experiments.

This strategy allows the language classifier to be less sensitive to noisy signal at the early stages of the training procedure. 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 senone classifiers [33].

The results of the source languages on *dev* data sets are reported in Table III. The results show that the WERs of the source languages on the LID-SHL-Model are lower than the source languages on the SHL-Model. However, the source languages on the adversarial SHL-Model obtains the best performance.

For the LID-SHL-Model and the adversarial SHL-Model, the configurations of the target models are shown in the right of Fig. 2, which are identical to the target model for the SHL-Model. The target models are trained using the shared parameters transferred from the LID-SHL-Model and the adversarial SHL-Model, respectively. The results of the target models are reported in Table II.

In Table II, the results show that the target models trained using shared parameters from the adversarial SHL-Model outperform the target models trained using shared parameters from the SHL-Model. The results also show that the target models trained using shared parameters from the adversarial

TABLE III
WERs (%) OF SOURCE LANGUAGES ON SOURCE MODELS TRAINED USING 4 SOURCE LANGUAGES (ASSAMESE, BENGALI, KURMANJI AND LITHUANIAN)

Source Model	Source Model Setting	Source Languages (FLP)			
		Assamese	Bengali	Kurmanji	Lithuanian
SHL-Model	with language specific hidden layers	48.4	51.5	64.5	64.9
LID-SHL-Model	+ language identification (LID)	48.1	51.3	64.2	64.6
Adversarial SHL-Model	+ gradient reversal layer (GRL)	47.3	50.3	63.2	63.7

TABLE IV
WERs (%) OF THE TARGET MODELS TRAINED USING SHARED PARAMETERS FROM VARIOUS SOURCE MODELS TRAINED USING 4 SOURCE LANGUAGES (ASSAMESE, BENGALI, KURMANJI AND LITHUANIAN). NOTE: INPUT FEATURES HAS I-VECTORS

Source Model	Source Model Setting	Target Languages (LLP)			Target Languages (FLP)		
		Pashto	Turkish	Vietnamese	Pashto	Turkish	Vietnamese
None	only target language + i-vector	55.9	53.5	56.7	46.9	44.9	47.8
SHL-Model	with language specific hidden layers + i-vector	50.3	50.1	55.5	44.1	42.9	47.6
LID-SHL-Model	+ language identification (LID)	49.7	49.6	54.9	43.7	42.2	47.2
Adversarial SHL-Model	+ gradient reversal layer (GRL)	45.2	47.1	53.7	42.1	41.1	46.5

TABLE V
WERs (%) OF SOURCE LANGUAGES ON SOURCE MODELS TRAINED USING 4 SOURCE LANGUAGES (ASSAMESE, BENGALI, KURMANJI AND LITHUANIAN). NOTE: INPUT FEATURES HAS I-VECTORS

Source Model	Source Model Setting	Source Languages (FLP)			
		Assamese	Bengali	Kurmanji	Lithuanian
SHL-Model	with language specific hidden layers + i-vector	47.3	50.4	63.4	63.7
LID-SHL-Model	+ language identification (LID)	47.1	50.2	63.1	63.5
Adversarial SHL-Model	+ gradient reversal layer (GRL)	46.2	49.1	62.2	62.6

SHL-Model outperform the target models trained using shared parameters from the LID-SHL-Model without adversarial loss.

The equal error rates (EERs) of the language classifier in source models are listed in the third column of Table VI. The results show that the performance of the adversarial language classifier is worse than the language classifier significantly.

F. Target Models Trained With Knowledge From Source Models Using I-Vector Features

In this group of experiments, we use i-vector features with the above acoustic features to train source models and target models. We also select 4 languages from the Babel datasets as the source languages, which are composed of Assamese, Bengali, Kurmanji and Lithuanian.

Language classification usually requires long-term context compared to the ASR task. The i-vector features carry language information. So we use i-vector features to capture this context.

There are 15 languages used to train i-vector extractors. The dimension of i-vectors is 100. Therefore, we use 100-dimensional i-vectors in addition to 3-dimensional pitch and 40-dimensional Fbank features plus their delta and delta-delta parameters as input features to train source models and target models. The configurations of the source models and target models are similar to the above-mentioned source models and target models, respectively. We only change the input features.

The EERs of the language classifier in source models with i-vector features are listed in the fourth column of Table VI. The results show that the performance of the adversarial language classifier is worse than the language classifier significantly.

TABLE VI
EER (%) OF LANGUAGE IDENTIFICATION (LID) IN SOURCE MODELS TRAINED WITHOUT OR WITH I-VECTOR FEATURES. NOTE: SOURCE MODELS ARE TRAINED USING 4 SOURCE LANGUAGES (ASSAMESE, BENGALI, KURMANJI AND LITHUANIAN)

Source Model	Source Model Setting	no i-vector	i-vector
LID-SHL-Model	LID	24.81	20.54
Adversarial SHL-Model	+ GRL	81.05	85.81

The results of the source languages on source models are reported in Table V. The results show that the WERs of the source languages on LID-SHL-Model are lower than the source languages on SHL-Model. However, the source languages on adversarial SHL-Model outperform the source languages on LID-SHL-Model. From Table V and III, we can find that all source models trained using i-vector features obtain more performance gains compared to the source models trained without i-vector features.

The results of the target models are listed in Table IV. The results in Table IV show that the target models trained using shared parameters from the adversarial SHL-Model outperform the target models trained using shared parameters from the SHL-Model and the LID-SHL-Model without adversarial loss. From Table IV and II, we can find that all target models trained using i-vector features obtain more performance gains compared to the target models trained without i-vector features.

Pashto target model trained using knowledge from the SHL-Model achieves up to 10.0% relative WER reduction over monolingual target model with i-vector features. Pashto target model trained using knowledge from the adversarial SHL-Model also achieves up to 19.1% relative WER reduction over monolingual target model with i-vector features. However, Vietnamese target model trained using knowledge from the SHL-Model

TABLE VII
WERs (%) OF THE TARGET MODELS TRAINED USING SHARED PARAMETERS FROM VARIOUS SOURCE MODELS TRAINED USING DIFFERENT NUMBER OF SOURCE LANGUAGES

#Source Languages	Source Model Setting	Target Languages (LLP)			Target Languages (FLP)		
		Pashto	Turkish	Vietnamese	Pashto	Turkish	Vietnamese
0	only target language + i-vector	55.9	53.5	56.7	46.9	44.9	47.8
4	with language specific hidden layers + i-vector	50.3	50.1	55.5	44.1	43.2	47.6
	+ language identification (LID)	49.7	49.6	54.9	43.7	42.9	47.2
	+ gradient reversal layer (GRL)	45.2	47.1	53.7	42.1	42.1	46.5
8	with language specific hidden layers + i-vector	49.2	48.8	55.1	43.5	42.7	47.3
	+ language identification (LID)	48.7	48.3	54.6	43.2	42.4	47.0
	+ gradient reversal layer (GRL)	44.7	46.4	53.2	41.6	41.7	46.2
12	with language specific hidden layers + i-vector	48.7	48.1	54.8	43.1	42.1	46.9
	+ language identification (LID)	48.2	47.5	54.3	42.8	41.8	46.6
	+ gradient reversal layer (GRL)	44.4	45.8	52.8	41.3	41.2	45.7

achieves up to 2.1% relative WER reduction over monolingual target model with i-vector features. Vietnamese target model trained using knowledge from the adversarial SHL-Model also achieves up to 5.3% relative WER reduction over monolingual target model with i-vector features. The possible reason is that the 4 source languages belong to Indo-European language family. Meanwhile, Pashto is also Indo-European language. So the cross-lingual knowledge transferred from the source model trained on the 4 source languages can be more helpful for Pashto. However, Vietnamese is Austroasiatic languages which is different from Indo-European languages. So the Vietnamese language gets less benefit from the source model trained on the 4 source languages.

G. The Effect of Different Number of Source Languages

In this section, our main concern is to evaluate the performance of adversarial training when the source models are trained on more languages. The number of the source languages in multilingual training was one of the important factors in Babel Project. Therefore, we try to use more source languages to train source models.

Although previous studies show that the Babel datasets consist of 28 languages, we can only obtain 15 languages from the linguistic data consortium (LDC). So we use 12 languages as source languages and 3 languages as target languages. The source languages are divided into 3 sets. The first set of source languages has 4 languages: Assamese, Bengali, Kurmanji and Lithuanian. The second set of source languages has 8 languages: Assamese, Bengali, Kurmanji, Lithuanian, Tamil, Telugu, Haitian and Tok Pisin. The third set of source languages has 12 languages, which are all the source languages in Table I.

The above-mentioned experiments show that all target models trained using i-vector features obtain more improvements compared to the target models without i-vector features. So the input features of the source and target models are 100-dimensional i-vectors in addition to 3-dimensional pitch and 40-dimensional Fbank features plus their delta and delta-delta parameters in this section.

The configurations of the source models and target models are similar to the above-mentioned source and target models, respectively. The only difference is that the source models are trained using more source languages. The results of the target models are reported in Table VII.

The results show that the target model trained using the shared layers from the SHL-Model obtains WER reduction when the number of source languages increases. The results also show that the target model trained using the shared layers from the adversarial SHL-Model achieves further performance improvement when the number of source languages increases.

When the source models are trained using 4 source languages, the target model trained using the knowledge transferred from the adversarial SHL-Model achieves up to 10.1% relative WER reduction compared to the target model trained using the knowledge transferred from the SHL-Model.

When the source models are trained using 8 source languages, the target model trained using the knowledge transferred from the adversarial SHL-Model achieves up to 9.1% relative WER reduction compared to the target model trained using the knowledge transferred from the SHL-Model.

When the source models are trained using 12 source languages, the target model trained using the knowledge transferred from the adversarial SHL-Model achieves up to 8.8% relative WER reduction compared to the target model trained using the knowledge transferred from the SHL-Model.

Previous work [24] on Pashto FLP condition using the Babel data reported WER as low as 45.7%. But many competitive teams [46], [47] in NIST OpenKWS 2013 reported WER of 47.1% or higher. In addition, lots of competitive systems [46], [47] in NIST OpenKWS 2013 reported WER of 48.1% or higher on Turkish FLP condition. Past work [45], [46] on Vietnamese FLP condition using the Babel data reported WER as low as 45% on combined systems. But many promising systems in NIST OpenKWS 2013 reported WER of 50% or higher, among them the Babel team SWORDFISH (led by ICSI) reported 55.9% on a single system.

In our experiments, the best model achieves up to 41.3%, 41.2%, 45.7% WER on Pashto, Turkish and Vietnamese FLP condition, respectively. Our best model also achieves up to 44.4%, 45.8%, 52.8% WER on Pashto, Turkish and Vietnamese LLP condition, respectively.

V. DISCUSSIONS

The above experimental results show that the proposed language-adversarial transfer learning is effective. Some interesting observations are made as follows.

All the target models benefit from both better feature coverage and better initialization via cross-lingual knowledge transfer learning. Cross-lingual knowledge transfer learning is a special case of transfer learning. The parameters of the shared layers transferred from the source model are used to initialize the target model. This is helpful for at least two reasons. One reason is that the target model will have parameters for feature types observed in the source languages as well as the target language. Thus it has better feature coverage. The other reason is that the training objective is non-convex. So this initialization can be helpful in avoiding bad local optima.

The target model trained utilizing the shared knowledge transferred from the adversarial SHL-Model outperforms the target models trained using the shared parameters transferred from the SHL-Model. Moreover, the target model trained utilizing the shared knowledge transferred from the adversarial SHL-Model also outperforms the target models trained using the shared parameters transferred from the LID-SHL-Model without adversarial loss. This is because the shared hidden layers of the SHL-Model and LID-SHL-Model learn some unnecessary language specific features. The adversarial training makes the shared layers to prevent from learning the language dependent features. So the shared layers of the the adversarial SHL-Model can learn more language invariant features. The language invariant features are helpful for improving the performance of the target model.

The target model trained using the shared parameters from the SHL-Model obtains WER reduction when the number of source languages increases. Moreover, the target model trained using the shared parameters from the adversarial SHL-Model achieves further performance improvement when the number of source languages increases.

In summary, all the target models benefit from both better feature coverage and better initialization via transfer learning. Furthermore, the adversarial learning forces the shared hidden layers of the shared-private model to learn more language invariant features. The target models trained using the shared parameters from the adversarial SHL-Model obtain performance gains when the number of source languages increases. Finally, the target model benefits from the language invariant features by language-adversarial transfer learning.

VI. Conclusion

This paper proposes language-adversarial transfer learning to improve the performance of low-resource speech recognition tasks. Adversarial learning is used to ensure that the shared layers can learn language invariant features. Experiments are conducted on IARPA Babel datasets. The results show that the target model trained using the knowledge from the adversarial SHL-Model obtains performance improvement by up to 10.1% relative WER reduction over the target model trained using the knowledge transferred from the SHL-Model. The results also show that the target model trained using the shared parameters from the adversarial SHL-Model achieves WER reduction when the number of source languages increases. The current study randomly chooses the source languages. Further work will study how to select source languages effectively. Moreover, it is well known that adversarial learning is difficult to get its best performance. More dedicated algorithm will be studied.

References

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

[2] W. Xiong *et al.*, "The Microsoft 2016 conversational speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5255–5259.

[3] G. Saon *et al.*, "English conversational telephone speech recognition by humans and machines," in *Proc. INTERSPEECH*, 2017, pp. 132–136.

[4] B. Li *et al.*, "Acoustic modeling for Google home," in *Proc. INTERSPEECH*, 2017, pp. 399–403.

[5] J. Ma, F. Keith, T. Ng, M. H. Siu, and O. Kimball, "Improving deliverable speech-to-text systems with multilingual knowledge transfer," in *Proc. INTERSPEECH*, 2017, pp. 127–131.

[6] M. Karafiat *et al.*, "2016 BUT Babel system: Multilingual BLSTM acoustic model with i-vector based adaptation," in *Proc. INTERSPEECH*, 2017, pp. 719–723.

[7] T. Sercu *et al.*, "Network architectures for multilingual speech representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5295–5299.

[8] S. M. Gass and L. Selinker, *Language Transfer in Language Learning*. Amsterdam, the Netherlands: John Benjamins, 1986.

[9] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *Proc. Spoken Lang. Technol.*, 2012, pp. 336–341.

[10] G. Heigold *et al.*, "Multilingual acoustic models using distributed deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8619–8623.

[11] P. Swietojanski, A. Ghoshal, and S. Renals, "Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR," in *Proc. Spoken Lang. Technol. Workshop*, 2013, pp. 246–251.

[12] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[13] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, vol. 27, pp. 3320–3328.

[14] Y. Miao and F. Metze, "Improving low-resource CD-DNN-HMM using dropout and multilingual DNN training," in *Proc. INTERSPEECH*, 2013, pp. 2237–2241.

[15] J. Cui *et al.*, "Multilingual representations for low resource speech recognition and keyword search," in *Proc. Autom. Speech Recognit. Understanding*, 2015, pp. 259–266.

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

[17] J. Trmal *et al.*, "The Kaldi OpenKWS system: Improving low resource keyword search," in *Proc. INTERSPEECH*, 2017, pp. 3597–3601.

[18] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[19] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7319–7323.

[20] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual MRASTA features for low-resource keyword search and speech recognition systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 7854–7858.

[21] Y. Zhang, E. Chuangsuwanich, and J. Glass, "Language 1-D-based training of multilingual stacked bottleneck features," in *Proc. INTERSPEECH*, 2014, pp. 1–5.

[22] S. Thomas, K. Audhkhasi, J. Cui, B. Kingsbury, and B. Ramabhadran, "Multilingual data selection for low resource speech recognition," in *Proc. INTERSPEECH*, 2016, pp. 3853–3857.

[23] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. Lecun, "Very deep multilingual convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4955–4959.

[24] W. Hartmann, R. Hsiao, and S. Tsakalidis, "Alternative networks for monolingual bottleneck features," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5290–5294.

[25] N. F. Chen *et al.*, "Low-resource spoken keyword search strategies in georgian inspired by distinctive feature theory," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Summit Conf.*, 2017, pp. 1322–1327.

[26] V. H. Do, N. F. Chen, B. P. Lim, and M. A. Hasegawa-Johnson, "Multitask learning for phone recognition of under resourced languages using mismatched transcription," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 3, pp. 501–514, Mar. 2018.

[27] H. Xu, V. H. Do, X. Xiao, and E. S. Chng, "A comparative study of BNF and DNN multilingual training on cross-lingual low-resource speech recognition," in *Proc. INTERSPEECH*, 2015, pp. 2132–2136.

[28] S. Thomas, "Data-driven neural network based feature front-ends for automatic speech recognition," in Ph.D. dissertation, Center Lang. Speech Process., Johns Hopkins Univ., Baltimore, MD, USA, 2012.

[29] S. Scanzio, P. Laface, L. Fissore, R. Gemello, and F. Mana, "On the use of a multilingual neural network front-end," in *Proc. INTERSPEECH*, 2008, pp. 2711–2714.

[30] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7304–7308.

[31] H. Xu *et al.*, "Semi-supervised and cross-lingual knowledge transfer learnings for DNN hybrid acoustic models under low-resource conditions," in *Proc. INTERSPEECH*, 2016, pp. 1315–1319.

[32] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *Proc. Autom. Speech Recognit. Understanding*, 2011, pp. 1–4.

[33] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[34] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, vol. 3, pp. 2672–2680.

[35] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 2962–2971.

[36] X. Chen, Z. Shi, X. Qiu, and X. Huang, "Adversarial multi-criteria learning for chinese word segmentation," in *Proc. Assoc. Comput. Linguistics*, 2017, pp. 1193–1203.

[37] M. Zhang, Y. Liu, H. Luan, and M. Sun, "Adversarial training for unsupervised bilingual lexicon induction," in *Proc. Assoc. Comput. Linguistics*, 2017, pp. 1959–1970.

[38] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition," in *Proc. INTERSPEECH*, 2016, pp. 2369–2372.

[39] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1180–1189.

[40] M. Andrew, H. Lempitsky, and Ng. Lempitsky, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–6.

[41] P. Golik, Z. Tske, R. Schlter, and H. Ney, "Multilingual features based keyword search for very low-resource languages," in *Proc. INTERSPEECH*, 2015, pp. 1260–1264.

[42] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Operating Syst. Des. Implementation*, 2016, pp. 265–283.

[43] D. Martłnez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in i-vectors space," in *Proc. INTERSPEECH*, 2011, pp. 861–864.

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

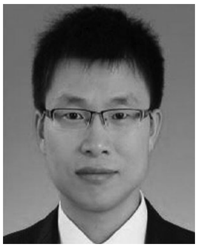
[45] N. F. Chen *et al.*, "Strategies for vietnamese keyword search," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 4121–4125.

[46] F. Metze *et al.*, "Models of tone for tonal and non-tonal languages," in *Proc. Autom. Speech Recognit. Understanding*, 2013, pp. 261–266.

[47] K. M. Knill, M. J. F. Gales, S. P. Rath, P. C. Woodland, C. Zhang, and S. X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *Proc. Autom. Speech Recognit. Understanding*, 2013, pp. 138–143.

Jiangyan Yi received the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2018, and the M.A. degree from the Graduate School of Chinese Academy of Social Sciences, Beijing, China, in 2010. She was a Senior R&D Engineer with Alibaba Group during 2011 to 2014. She is currently an Assistant Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her current research interests include speech processing, speech recognition, distributed computing, deep learning, and transfer learning.

Jianhua Tao received the Ph.D. degree from Tsinghua University, Beijing, China, in 2001, and the M.S. degree from Nanjing University, Nanjing, China, in 1996. He is currently a Professor with NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He has authored or coauthored more than eighty papers on major journals and proceedings including the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. His current research interests include speech recognition, speech synthesis and coding methods, human–computer interaction, multimedia information processing, and pattern recognition. He is the Chair or Program Committee Member for several major conferences, including ICPR, ACII, ICMI, ISCSLP, NCMMSC, etc. He is also the Steering Committee Member for the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, an Associate Editor for *Journal on Multimodal User Interface* and *International Journal on Synthetic Emotions*, and the Deputy Editor-in-Chief for Chinese *Journal of Phonetics*. He was the recipient of several awards from the important conferences, such as Eurospeech, NCMMSC, etc.



Zhengqi Wen received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2008, and the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 2013. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include speech processing, speech recognition, and speech synthesis.

Ye Bai received the B.S. degree from China Agricultural University, Beijing, China, in 2016. He is currently working toward the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing, China. His current research interests include keywords spotting, speech recognition, language modeling, and decoding.