BUILDING DOMAIN INVARIANT SPOKEN LANGUAGE IDENTIFICATION SYSTEM USING ADVERSARIAL MULTI-TASK LEARNING

Muralikrishna H, Shantanu Kapoor, Dileep Aroor Dinesh

MANAS Lab, Indian Institute of Technology Mandi, India

ABSTRACT

State-of-the-art spoken language identification (LID) systems are vulnerable to domain-mismatch, especially when the training dataset has very limited channel and speaker diversity. In this paper, we consider a special case, where, the training database contains extremely clean speech samples recorded from a small number of speakers. This limited channel and speaker diversity in the training makes the LID system highly vulnerable to unseen target domain. To address this issue, we propose to use two data-augmentation schemes followed by an adversarial multi-task learning (AMTL) based training strategy. Specifically, we use a new channel perturbation and state-of-the-art speed perturbation methods of data-augmentation to increase the intra-class variations in the dataset. Followed by this, we use AMTL on the augmented dataset for encouraging the network to learn channel- and speaker-invariant representation of the input. We consider the IIT-Madras-Indic-TTS dataset which contains clean read speech samples for training and IIT-Mandi Indian languages dataset that contains spontaneous speech from YouTube videos for testing. We also release this IIT-Mandi dataset for the benefit of research community.

Index Terms— spoken language identification, data-augmentation, domain-mismatch, unseen domain

1. INTRODUCTION

State-of-the-art spoken language identification (LID) systems are vulnerable to domain-mismatch occurring due to the mismatch in channel, speaking style, background noise conditions, etc., between the training and testing samples. The impact of domain-mismatch can be reduced by using a training dataset containing sufficient training samples from all possible environments to which the system will be exposed to. However, this requirement is often not satisfied in low-resource languages. In this paper, we consider one such case, where, the training dataset contains only high-quality read speech samples that were actually recorded for building a text-to-speech synthesis (TTS) system. Also, in each language, the dataset contains samples from only two native speakers (one male and one female). Due to very limited speaker and channel (recording device) diversity in the training dataset, the LID system built using traditional approaches can not guarantee the best performance in an unseen real-world target domain. In this work, we investigate different methods to address this issue.

Data-augmentation is a common strategy used to increase the diversity and number of samples in the training database thereby encouraging the system to generalize in a better way. It is typically done by augmenting transformed version of the original training samples obtained using some label-preserving transformations [1, 2, 3, 4, 5, 6] or by augmenting unlabeled samples from other domains using semi-supervised approaches [7, 8]. While some of these label-preserving transformations are applied directly on the raw speech samples (such as speed perturbation [1] or artificially adding noise [2, 3]), other transformations are applied in the feature space (such as stochastic feature mapping (SFM) [4] and vocal tract length perturbation (VTLP) [5, 6]). In a typical semi-supervised approach [7, 8], the unlabeled samples from other domain are first passed through an existing system for obtaining their class labels and then only the samples classified with high confidence (above a threshold) are selected for data-augmentation. Note that, effectiveness of the data-augmentation in improving the robustness of a system depends on how well the augmented data represents the target domain.

Apart from data-augmentation, adversarial multi-task learning (AMTL) scheme is the another way of improving the robustness of deep neural network (DNN) based systems, which encourages the DNN to learn domain-invariant features. AMTL has been used successfully to enforce domain-invariance in many applications like speech recognition [9], speaker verification/recognition [10, 11, 12, 13], speech emotion recognition [14], etc. In AMTL, the overall network is typically considered to have three parts: a front-end feature extractor, a primary classifier for the main task and an auxiliary domain classifier for identifying the domain of the training samples [15]. By using training samples belonging to multiple domains (or datasets) with corresponding domain labels, AMTL forces the network to learn features that are discriminative for the main learning task but indistinguishable across domains (domain-invariant) by using adversarial training between the feature extractor and the domain classifier [15]. Note that, the amount of domain-invariance learnt by the DNN in this approach depends on the amount of

inter-domain differences present in the training samples.

Motivated by the strengths of data-augmentation and AMTL in improving the generalization of the systems, we propose a training strategy which combines both of them to improve the robustness of a LID system. Specifically, we first use two data-augmentation methods: a novel channel perturbation method to increase the channel diversity and a state-of-the-art speed perturbation method [1] to increase intra-class variations in the training dataset. We then use AMTL scheme on the augmented dataset to encourage the system to learn a speaker- and channel-independent representation of the input. We show that, the proposed combination of data-augmentation and AMTL significantly improves the robustness of LID system to an unseen target domain compared to robustness achieved when either of them used individually. To the best of our knowledge, this is the first reported effort to build a robust LID system using a training dataset which contains extremely clean speech samples from very limited number of speakers and with very less channel diversity.

The highlights of this paper are as follows. 1) Usage of two data-augmentation methods (including a novel channel perturbation strategy) to increase the diversity in the training dataset. 2) An adversarial multi-task learning (AMTL) based training to improve the robustness of a DNN based LID system. 3) Extensive experimentation using the proposed approach and comparison with state-of-the-art approaches on publicly available datasets.

Rest of this paper is organized as follows. In section 2, a description of datasets used in the study is given. In section 3, we explain the proposed method for improving the robustness of LID system. Section 4 describes experiments and results followed by conclusions in section 5.

2. DATASETS USED IN THE STUDY

We use a set of 9 Indian languages in this study. To train our LID systems, we use the samples from IIT-Madras-Indic-TTS dataset which was actually created for building text-to-speech synthesis system in Indian languages[1]. These samples were originally recorded with a sampling rate of 48 kHz using high quality recording devices in a controlled environment without any background noise and echo [16]. In each language, the recording was done by 2 professional native speakers (one male and one female) by reading the given text with minimum pitch variation. These speech samples do not contain any code-mixing (no out-of-vocabulary words). In order to simulate language identification in low-resourced conditions, we have used only around 7 hours of speech (equally distributed among male and female speakers) in each of the languages. Out of 7 hours, we used 6 hours of speech for training and remaining 1 hour of speech for validation. This validation data represents the **seen** test data as the background conditions in this dataset matches with that of training dataset. Note that, the amount of speaker, channel and pitch variations in this training dataset are very limited. More details about this dataset can be seen in [16].

We use the samples from IIT-Mandi Indian languages dataset for testing the performance of LID systems. This dataset contains the audio samples extracted from YouTube videos on education and personal interviews. There are 6 speakers (3 male and 3 female) in each of the languages. Due to spontaneous nature of the speech, there is significant variation in pitch and tone in these utterances. Some of the samples contain code-mixed speech due to the usage of few common English words during the normal speech. Also, as different type of microphones are used in different recordings, there is significant channel diversity in these samples. This dataset represents the **unseen** target domain for the LID system as no samples from this test dataset are used for training or validation. We release this dataset for the benefit of research community[2].

In our experiments, we have down-sampled all the speech files to 8 kHz. We have divided larger speech files into smaller chunks such that all speech samples used in this work have a duration between 8 to 15 seconds. The list of languages along with total duration in hours and number of speakers for each language in training (including validation) and testing datasets are given in Table 1.

Table 1: Indian languages used in the study along with their duration in hours and number of speakers.

Language	Indic-TTS dataset (Train + validation)		IIT-Mandi dataset (Test dataset)	
	Hours	Speakers	Hours	Speakers
Assamese	7.21	2	0.5	6
Bengali	7.02	2	0.5	6
Gujarathi	7.10	2	0.5	6
Hindi	7.02	2	0.5	6
Kannada	7.20	2	0.5	6
Malayam	7.12	2	0.5	6
Manipuri	7.04	2	0.5	6
Odia	7.10	2	0.5	6
Telugu	7.14	2	0.5	6

3. PROPOSED METHOD

The block diagram of proposed method for obtaining a domain-mismatch robust LID system is given in Fig. 1. Raw speech samples from the original training dataset are first processed by channel perturbation and speed perturbation blocks to provide an augmented dataset. This augmented dataset is then used for training the DNN-based LID system using AMTL. The proposed LID system contains a feature extractor block to extract an utterance-level embedding (represented as

[1]https://www.iitm.ac.in/donlab/tts/index.php

[2]https://speechiitmandi.github.io/lid/

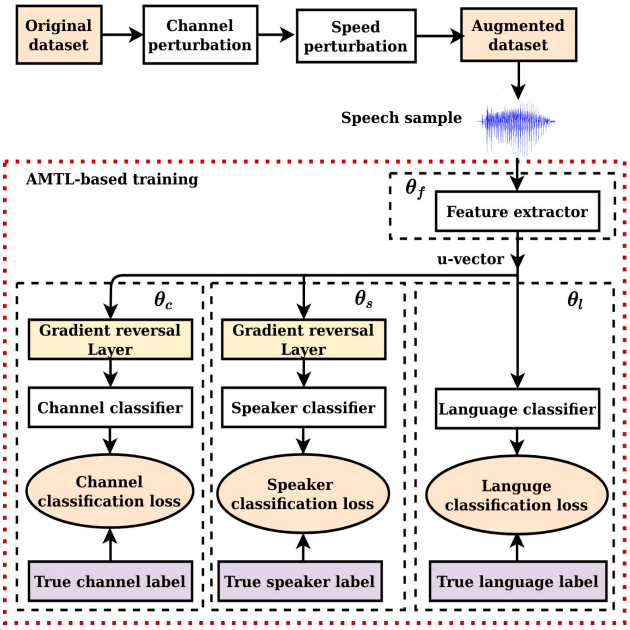

Fig. 1: Block diagram of the proposed method for obtaining domain-mismatch robust LID system.

u-vector in Fig.1) of the input speech. Let $\boldsymbol{\theta_f}$ represent the parameters (weights and biases) of the feature extractor block. This u-vector is then simultaneously processed by a primary language classifier block with parameters $\boldsymbol{\theta_l}$, a speaker classifier block with parameters $\boldsymbol{\theta_s}$ and a channel classifier block with parameters $\boldsymbol{\theta_c}$. Using proposed AMTL-based training, the parameters of the LID system $\boldsymbol{\theta_{net}} = \{\boldsymbol{\theta_f}, \boldsymbol{\theta_l}, \boldsymbol{\theta_s}, \boldsymbol{\theta_c}\}$ are tuned such that the obtained u-vector carries only LID-specific contents by ignoring speaker and channel specific contents in the input. This is achieved using the adversarial training between speaker classifier and feature extractor as well as channel classifier and feature extractor.

3.1. Data-augmentation using channel perturbation

Since all speech samples in the original training database were recorded using high quality recording devices (microphones), it contains very limited channel diversity. In order to increase the channel diversity, we propose a simple channel perturbation strategy. This method is motivated by the fact that different microphones have different frequency responses. Due to this, same speech utterance recorded using different microphones can be heard quite differently. In the proposed approach, we pass the speech sample through a set of two band-pass filters (BPF) with different pass-band frequencies to imitate the recordings obtained by two hypothetical microphones as shown in Fig. 2. This results in 3 fold increase in the size of dataset (original and 2 perturbed replicas).

The bandwidths of these filters have to be chosen such that the resultant samples after perturbation are heard quite

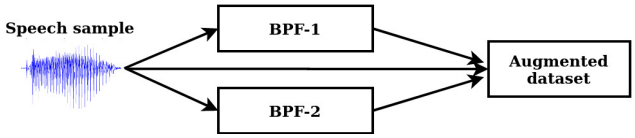

Fig. 2: Proposed channel perturbation method of data-augmentation.

differently than the original sample, but clearly intelligible. The Fig. 3 shows the effect of channel perturbation when the first BPF (BPF-1) has a pass-band from 100 to 2500 Hz and second BPF (BPF-2) has a pass-band from 500 to 3500 Hz. The top most spectrogram (represented as A) in Fig. 3 belongs to the original signal, middle spectrogram (B) belongs to the output of BPF-1 and bottom most spectrogram (C) belongs to output of BPF-2. Note that, the BPF-2 has suppressed all frequency components below 500 Hz (including the pitch frequency) in the sample which makes the resultant sample to be heard significantly different than the original sample.

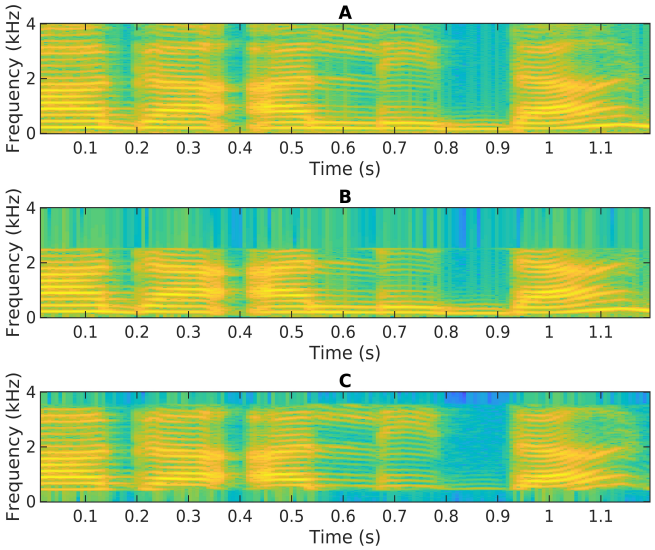

Fig. 3: Spectrograms of the original and channel perturbed signals.

3.2. Data-augmentation using speed perturbation

Audio speed perturbation is a simple label preserving transformation that produces warped time signal [1]. Given a speech signal $x(t)$, speed perturbation produces a warped time signal $x(\alpha t)$, where, α is the factor used to vary speed of the signal. As suggested in [1], we used speed factor values as 0.9 and 1.1 to produce two replicas of original samples which are then augmented to the original training dataset. Due to the variation in speed, this operation produces more variety within each language class.

Due to the combined action of speed perturbation and channel perturbation, there is a 9 fold increase in the size of the training dataset. This augmented dataset is used for training the DNN-based LID system.

3.3. Front-end feature extractor (θ_f)

The role of the feature extractor block is to provide an utterance-level embedding (u-vector) of the input speech. The architecture of this block is shown in Fig. 4. It contains a bottleneck feature (BNF) extractor at the front followed by two bidirectional long short-term memory (BLSTM) layers and a mean pooling layer to provide the u-vector. We use a pre-trained BNF extractor to convert the speech utterance in to a sequence of 80-dimensional BNFs [17]. This BNF extractor was originally trained for identifying 3096 phone states from 17 languages. Each of the extracted BNFs covers a total context of 31 frames (325 ms) of input speech [17]. The BLSTM layers then analyze the sequence of BNFs by dividing them into small fixed-size chunks to provide LID-seq-senones [18]. These LID-seq-senones are nothing but the activations obtained at the output of second BLSTM layer for each of the fixed-size chunks [18]. The final utterance-level representation (u-vector) of the speech is computed as the mean of these LID-seq-senones. Though the number of LID-seq-senones obtained for a given sample depends on its length, the u-vector is a fixed-length utterance-level representation of the input speech.

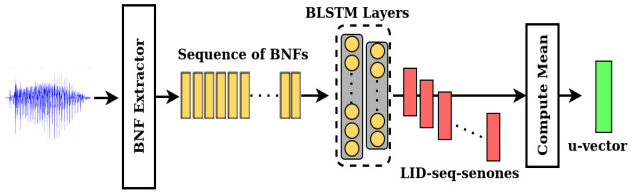

Fig. 4: Front-end feature extractor for obtaining u-vector.

Note that, without a sophisticated method, the obtained u-vector contains information about the recording session (like speaker and channel related information) embedded in it along with the required LID-specific contents. To improve the robustness of the LID-network towards unseen target domain, these nuisance factors in the u-vector have to be suppressed. We propose to use AMTL-based training to achieve this.

3.4. AMTL for domain-mismatch robust LID system

As shown in Fig. 1, the proposed system contains a feature extractor, a primary language classifier and auxiliary speaker and channel classifiers blocks. The output layers of all these three classifiers have *softmax* activation which enables us to interpret the output as class posteriors for the given input.

Specifically, given an input speech sample $\boldsymbol{x}$, the output of the language classifier is $P(y_l|\boldsymbol{x}, \boldsymbol{\theta_f}, \boldsymbol{\theta_l})$; where $y_l \in \{l_1, l_2, .., l_{N_l}\}$ denotes the language class. Here, N_l is the total number of languages. For the given training dataset with K samples, we can compute the total cross entropy loss for the language classifier as:

$$L_l(\boldsymbol{\theta_f}, \boldsymbol{\theta_l}) = -\sum_{i=1}^{K} log(P(y_l^i|\boldsymbol{x}^i, \boldsymbol{\theta_f}, \boldsymbol{\theta_l})) \quad (1)$$

Similarly, given input sample $\boldsymbol{x}$, the output of the speaker classifier is $P(y_s|\boldsymbol{x}, \boldsymbol{\theta_f}, \boldsymbol{\theta_s})$; where $y_s \in \{sp_1, sp_2, ..., sp_{N_s}\}$ denotes the speaker and N_s denotes total number of speakers. For the given training dataset with K samples, we can compute the cross entropy loss for the speaker classifier as:

$$L_s(\boldsymbol{\theta_f}, \boldsymbol{\theta_s}) = -\sum_{i=1}^{K} log(P(y_s^i|\boldsymbol{x}^i, \boldsymbol{\theta_f}, \boldsymbol{\theta_s})) \quad (2)$$

Similarly, for the given input sample $\boldsymbol{x}$, the output of the channel classifier is $P(y_c|\boldsymbol{x}, \boldsymbol{\theta_f}, \boldsymbol{\theta_c})$; where $y_c \in \{ch_1, ch_2, .., ch_{N_c}\}$ denotes the identity of the channel and N_c denotes total number of channels. For the given training dataset with K samples, we can compute the cross entropy loss for the channel classifier as:

$$L_c(\boldsymbol{\theta_f}, \boldsymbol{\theta_c}) = -\sum_{i=1}^{K} log(P(y_c^i|\boldsymbol{x}^i, \boldsymbol{\theta_f}, \boldsymbol{\theta_c})) \quad (3)$$

Using these individual loss functions, we define the total loss function for the network as:

$$L_T(\boldsymbol{\theta_f}, \boldsymbol{\theta_l}, \boldsymbol{\theta_s}, \boldsymbol{\theta_c}) = L_l(\boldsymbol{\theta_f}, \boldsymbol{\theta_l}) - \lambda_1 L_s(\boldsymbol{\theta_f}, \boldsymbol{\theta_s}) - \lambda_2 L_c(\boldsymbol{\theta_f}, \boldsymbol{\theta_c}) \quad (4)$$

Here, λ_1 and λ_2 are the trade off parameters to decide the impact of speaker and channel classification losses in the AMTL training. To make the u-vector independent of speaker and channel, the parameters of the network have to be jointly optimized as follows. The $\boldsymbol{\theta_f}$ has to be optimized to maximize both speaker and channel classification losses. $\boldsymbol{\theta_c}$ and $\boldsymbol{\theta_s}$ have to be adjusted to minimize channel and speaker classification losses respectively. Along with these, $\boldsymbol{\theta_f}$ and $\boldsymbol{\theta_l}$ have to be optimized to minimize the language classification loss.

These parameters are optimized using the stochastic gradient descend method as follows:

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{f}} \leftarrow \boldsymbol{\theta_f} - \mu(\frac{\partial L_l^i}{\partial \boldsymbol{\theta_f}} - \lambda_1 \frac{\partial L_s^i}{\partial \boldsymbol{\theta_s}} - \lambda_2 \frac{\partial L_c^i}{\partial \boldsymbol{\theta_c}}) \quad (5)$$

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{l}} \leftarrow \boldsymbol{\theta_l} - \mu \frac{\partial L_l^i}{\partial \boldsymbol{\theta_l}} \quad (6)$$

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{s}} \leftarrow \boldsymbol{\theta_s} - \mu \frac{\partial L_s^i}{\partial \boldsymbol{\theta_s}} \quad (7)$$

$$\widehat{\boldsymbol{\theta}}_{\boldsymbol{c}} \leftarrow \boldsymbol{\theta_c} - \mu \frac{\partial L_c^i}{\partial \boldsymbol{\theta_c}} \quad (8)$$

Here, L_l^i, L_s^i and L_c^i respectively indicate language, speaker and channel classification losses evaluated at i^{th} training example and $\widehat{\boldsymbol{\theta}}_{\boldsymbol{f}}$, $\widehat{\boldsymbol{\theta}}_{\boldsymbol{l}}$, $\widehat{\boldsymbol{\theta}}_{\boldsymbol{s}}$ and $\widehat{\boldsymbol{\theta}}_{\boldsymbol{c}}$ are the updated parameters. μ represents the learning rate. For implementing these relations, we use a gradient reversal layer (GRL) [15] between the feature extractor and speaker classifier as well as between feature extractor and channel classifier. The GRL acts as an identity transform during the forward pass. During the backpropagation, GRL reverses the sign of the gradient [15]. This enables

the adversarial training between the feature extractor and domain classifier (speaker or channel classifier).

Due to the AMTL-based training, the LID system learns to produce u-vector that is independent of speaker and channel related contents. This improves the robustness of the LID system towards mismatch in speakers and channels during the testing.

4. EXPERIMENTS AND RESULTS

In this section, we study the effectiveness of the proposed training strategy with data-augmentation and AMTL. All LID systems in this paper are evaluated using accuracy (in %) and the C_{avg} (in %) which is given in the NIST Language Recognition Evaluation 2015 [19]. Lower values of C_{avg} indicates better performance. Pytorch tool [20] is used for the implementation of all networks in this work.

In all experiments, the BLSTM layers in the feature extractor process the sequence of BNFs by dividing it into fixed-size chunks of 35 BNFs. Each chunk covers approximately 665 ms of speech. The weights of the pre-trained BNF extractor used in the feature extractor block are kept unchanged during the training.

4.1. Baseline systems

In order to compare the effectiveness of the proposed data-augmentation and AMTL training, we first build a baseline LID system that does not use either of them. This is a DNN-based LID system that contains only feature extractor and language classifier blocks in Fig. 1 which is trained using the samples from original dataset. The number of nodes in the first and second BLSTM layers in the feature extractor are set as 128 and 64 respectively. The language classifier block first processes the u-vector using a dense layer with 128 nodes having *tanh* activation followed by the output layer with 9 nodes to represent the languages. The result obtained for this baseline system (LIDnet) on validation (Indic-TTS) and test (IIT-Mandi) datasets are shown in the first row of Table 2.

Apart from this DNN-based LID system, we have also built an i-vector based LID system [21] using 128-dimensional LID-seq-senones (obtained using pre-trained LIDnet) as input. Note that, while DNN-based LID system uses the u-vector obtained as a mean of LID-seq-senones to represent the input, this system uses the factor analysis of the LID-seq-senones to get the i-vector. We used a GMM-UBM with 2048 components with diagonal-covariance matrix. The total i-vector extractor is trained in 5 iterations. The dimensionality of i-vectors is set to 400. We used only the original training dataset for training this system. These i-vectors are then classified using probabilistic linear discriminant analysis (PLDA) approach. Implementation was done using MSR identity toolbox [22]. Results for this sytem are given in 2^{nd} row of Table 2.

Table 2: Performances of baseline systems

LID system	Indic-TTS		IIT-Mandi	
	Acc	Cavg	Acc	Cavg
LIDnet	98.20	1.04	23.22	43.54
i-vec+PLDA	97.50	1.33	24.00	43.02

From the results, it is evident that both baseline LID system have over-fitted to the background conditions of the training dataset. Both systems perform very well on the validation data (Indic-TTS) due to the perfect matching in background conditions of the validation samples with that of training samples. However, the performance on the unseen target domain (IIT-Mandi dataset) is very poor due to the significant mismatch in the background conditions. Next, we study the effect of data-augmentation in reducing the impact of domain-mismatch.

4.2. Effect of data-augmentation on the performance

We first perform data-augmentation using proposed channel perturbation method which results in 3 fold increase in the size of training dataset. Since the network has to learn more complex relation in this case, the number of nodes in the 1^{st} and 2^{nd} BLSTM layers in the feature extractor are increased to 192 and 96 keeping other parameters of the network (LIDnet) unchanged. The results obtained for the LID system trained on the augmented dataset obtained using channel perturbation are shown in 1^{st} row of Table 3 (represented as LIDnet+ch). We also experimented by applying only speed perturbation on the original training dataset. Like channel perturbation, this also leads to 3 fold increase in the size of dataset. The results obtained for speed perturbation case (LIDnet+sp) are shown in 2^{nd} row of Table 3. It can be seen from the results that, both channel perturbation and speed perturbation methods have provided improvement in performance in unseen target domain compared to baseline systems.

Table 3: Performances of proposed approach

LID system	Indic-TTS		IIT-Mandi	
	Acc	Cavg	Acc	Cavg
LIDnet+ch	98.30	1.02	30.37	39.17
LIDnet+sp	97.60	1.70	28.68	41.38
LIDnet+ch+sp	98.66	0.93	33.55	36.95
LIDnet+AMTL	94.33	3.46	32.20	38.02
LIDnet+ch+AMTL	93.50	3.78	38.44	34.62
LIDnet+ch+sp+AMTL	92.35	4.42	40.20	31.12

Next, we experimented by combining both augmentation methods in which, we applied channel perturbation first followed by speed perturbation. This leads to a total 9 fold increase in the size of the dataset. In this case, the number of nodes in the 1^{st} and 2^{nd} BLSTM layers of the network are increased to 320 and 128 respectively. Results obtained for

this system (LIDnet+ch+sp) are given in 3^{rd} row of Table. 3.

From the results it is seen that, a good improvement in the performance is obtained on unseen target domain (IIT-Mandi datset) due to the combination these two data-augmentation methods compared to the improvement obtained by individual data-augmentation methods. This indicates that these two augmentation techniques are complementary to each other. Reason for this is the following. The speed perturbation scales the time axis by a factor α due to which all frequency components in the signal gets scaled by a factor α^{-1}. All frequency components in the original signal are effected by this operation. However, channel perturbation simply suppresses the frequency components in the stop-band of BPF, leaving other frequency components unaffected. Hence, these two operations have different effect on the output of Mel-scale filter banks. Since the BNFs used in this work are derived from the output of Mel-scale filter banks [17], these two perturbation methods affect the BNFs differently.

Note that, the performance of LID systems on Indic-TTS validation samples remain almost the same in all 3 cases of data-augmentation. Reason is that, the validation dataset is actually a (non-overlapping) part of original training dataset. Hence, the system has already seen the background conditions in the validation samples during the training.

4.3. Effect of AMTL on the performance

Firstly, we trained a LID system on the original training dataset (without data-augmentation) using AMTL approach. In this, the system contains only language and auxiliary speaker classifiers excluding channel classifier (due to lack of channel diversity in the original dataset). The speaker classifier contains a dense layer with 128 nodes followed by the output layer with 18 nodes to represent the speakers. Using adversarial training between feature extractor and speaker classifier, this LID system is trained to capture a speaker-independent representation of the input. Results obtained for this LID system (LIDnet+AMTL) are shown in 4^{th} row of Table 3. This system has performed moderately well on the IIT-Mandi dataset.

Next, we experimented by using AMTL training on the augmented dataset obtained after channel perturbation (which leads to 3 fold increase in the size of dataset). In this case, the system contains both speaker and channel classifiers along with the primary language classifier as shown in Fig. 1. Like language and speaker classifiers, the channel classifier contains a dense layer with 128 nodes followed by the output layer. There are 3 nodes in the output layer to represent 3 channels (original and 2 perturbed). Results obtained for this system (LIDnet+ch+AMTL) are shown in 5^{th} row of Table 3. There is significant performance improvement on the IIT-Mandi dataset as the AMTL training encourages the network (LIDnet+ch+AMTL) to learn both speaker- and channel-independent u-vector.

Finally, we experimented by using AMTL on the augmented dataset obtained after both speaker and channel perturbation (9 fold increase in the size of dataset). Like in the previous case, the LID system contains auxiliary speaker and channel identifier blocks along with the language classifier. The results obtained for this system (LIDnet+ch+sp+AMTL) are given in the 6^{th} row of Table 3.

This system (LIDnet+ch+sp+AMTL) has performed better than all other systems on IIT-Mandi dataset. The AMTL-based training has encouraged the network to learn a speaker- and channel-independent representation of the input. Due to this, the system became less vulnerable to the mismatch in speaker and channel conditions in the unseen IIT-Mandi dataset. Also, the channel and speed perturbation based data-augmentation has helped the system to generalize in a better way.

Note that, all three systems trained using AMTL approach have performed poorly on Indic-TTS validation samples compared to systems without AMTL. Reason is that, AMTL-based training prevents the system in utilizing the background-specific contents in deciding the class label. Since the speakers in Indic-TTS training and validation datasets are the same, systems without AMTL training can additionally utilize the speaker characteristics along with the LID-specific contents in the input to decide the language class. However, this is undesirable as we want the systems to learn only LID-specific contents in the input.

In spite of data-augmentation and AMTL-based training, the performance of our LID system is far below than the ideal performance (100% accuracy). This indicates the hard challenge to be faced in building a robust LID system in low-resource conditions.

5. CONCLUSIONS

In this paper, we proposed a strategy to improve the robustness of a LID system towards an unseen target domain in low-resource conditions. The proposed approach combines the strengths of data-augmentation and AMTL-based training. Specifically, we proposed a new channel perturbation method of data-augmentation for increasing the diversity in the training dataset. Results obtained shows that the combination of proposed channel perturbation and state-of-the-art speed perturbation methods significantly improves the generalization of the LID system. In addition to this, the AMTL-based training of the LID system using augmented dataset further improves the robustness of the system towards unseen target domain.

In the future work, we will explore more methods of data-augmentation and training strategies to further improve the robustness of the LID system.

6. REFERENCES

[1] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[2] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[3] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al., "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[4] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury, "Data augmentation for deep neural network acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1469–1477, 2015.

[5] Navdeep Jaitly and Geoffrey E Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013, vol. 117.

[6] Naoyuki Kanda, Ryu Takeda, and Yasunari Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 309–314.

[7] Anton Ragni, Kate M Knill, Shakti P Rath, and Mark JF Gales, "Data augmentation for low resource languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[8] Lan Wang, Mark JF Gales, and Philip C Woodland, "Unsupervised training for mandarin broadcast news and conversation transcription," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*. IEEE, 2007, vol. 4, pp. IV–353.

[9] Yusuke Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition.," in *Interspeech*. San Francisco, CA, USA, 2016, pp. 2369–2372.

[10] Zhong Meng, Yong Zhao, Jinyu Li, and Yifan Gong, "Adversarial speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6216–6220.

[11] Gajan Suthokumar, Vidhyasaharan Sethu, Kaavya Sriskandaraja, and Eliathamby Ambikairajah, "Adversarial multi-task learning for speaker normalization in replay detection," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6609–6613.

[12] Hongji Wang, Heinrich Dinkel, Shuai Wang, Yanmin Qian, and Kai Yu, "Cross-domain replay spoofing attack detection using domain adversarial training.," in *Interspeech*, 2019, pp. 2938–2942.

[13] Shuai Wang, Johan Rohdin, Lukás Burget, Oldrich Plchot, Yanmin Qian, Kai Yu, and Jan Cernockỳ, "On the usage of phonetic information for text-independent speaker embedding extraction.," in *INTERSPEECH*, 2019, pp. 1148–1152.

[14] M. Abdelwahab and C. Busso, "Domain adversarial for acoustic emotion recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2423–2435, 2018.

[15] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*, 2015, pp. 1180–1189.

[16] Arun Baby, Anju Leela Thomas, NL Nishanthi, TTS Consortium, et al., "Resources for indian languages," in *Proceedings of Text, Speech and Dialogue*, 2016.

[17] Anna Silnova, Pavel Matejka, Ondrej Glembek, Oldrich Plchot, Ondrej Novotny, Frantisek Grezl, Petr Schwarz, Lukas Burget, and Jan Cernocky, "BUT/phonexia bottleneck feature extractor," in *proceedings of Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 283–287.

[18] H. Muralikrishna, S. Pulkit, J. Anuksha, and A. D. Dileep, "Spoken language identification using bidirectional lstm based lid sequential senones," in *proceeding of 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019)*, pp. 320–326.

[19] "The 2015 NIST Language Recognition Evaluation plan (lre15)," *https://www.nist.gov/itl/iad/mig/2015-language-recognition-evaluation*, 2015.

[20] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," in *proceedings of NIPS-W*, 2017.

[21] Najim Dehak, Pedro A Torres Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via i-vectors and dimensionality reduction," in *proceedings*

of Twelfth annual conference of the international speech communication association, 2011.

[22] Seyed Omid Sadjadi, Malcolm Slaney, and Larry Heck, "Msr identity toolbox v1. 0: A matlab toolbox for speaker-recognition research," .