Expert Systems With Applications 110 (2018) 290–297

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Curriculum learning based approach for noise robust language identification using DNN with attention

Ravi Kumar Vuddagiri*, Hari Krishna Vydana, Anil Kumar Vuppala

Speech Processing Laboratory, International Institute of Information Technology, Hyderabad, India

ARTICLE INFO

Article history:
Received 6 December 2017
Revised 11 May 2018
Accepted 2 June 2018
Available online 7 June 2018

Keywords:
Automatic language identification
Background environments
I-vector
Deep neural network (DNN)
DNN with attention (DNN-WA)
Multi signal-to-noise (SNR) models
Curriculum learning

ABSTRACT

Automatic language identification (LID) in practical environments is gaining a lot of scientific attention due to rapid developments in multilingual speech processing applications. When an LID is operated in noisy environments a degradation in the performance can be observed and it can be majorly attributed to mismatch between the training and operating environments. This work is aimed towards developing an LID system that can robustly operate in clean and noisy environments. Traditionally, to reduce the mismatch between training and operating environments, noise is synthetically induced to the training corpus and these models are termed as multi-SNR models. In this work, various curriculum learning strategies are explored to train multi-SNR models, such that the trained models have better generalization in performance over varying background environments. I-vector, Deep neural networks (DNN) and DNN With Attention (DNN-WA) architectures are used in this work for developing LID systems. Experimental verification of the proposed approach is carried out using IIIT-H Indian database and AP17-OLR database. The performance of LID system is tested at different signal-to-noise ratio (SNR) levels using white and vehicular noises from NOISEX dataset. In comparison to multi-SNR models, the LID systems trained with curriculum learning have performed better in terms of equal error rate (EER) and generalization in EER across varying background environments. The degradation in the performance of LID systems due to environmental noise has been effectively reduced by training multi-SNR models using curriculum learning.

1. Introduction

Language identification (LID) refers to the task of tagging the spoken utterance with its language identity. Growing interest in multilingual dialog systems has developed a lot of scientific attention towards the development of LID systems. Human-computer interaction through speech would be more effective when the interaction can take place in multiple languages. LID systems can act as a front-end switch for a multilingual dialog system. Using LID system as a front-end switch in a dialog system, the language-specific phonotactic constraints could be used to operate the dialog system more robustly. Due to similarities in the origin and the overlapping phone-sets, developing an LID system for Indian languages is quite challenging. There have been studies focusing on the development of LID systems for Indian languages, but developing an LID system that can operate robustly in varying background environments (i.e. the presence of noise at varying signal-to-noise (SNR) levels) is largely understudied problem. The degradation in the performance of an LID system in varying background environments is mostly due to a mismatch in training and operating environments. This work aims at developing LID systems that can be operated in varying background environments.

A detailed review of developing various language identification systems has been presented in Ambikairajah, Li, Wang, Yin, and Sethu (2011), Zissman and Berkling (2001), Zissman (1996), Li, Ma, and Lee (2007), Muthusamy, Barnard, and Cole (1994) and Ma, Guan, Li, and Lee (2002). Majority of earlier studies for developing LID systems are inspired by the speaker recognition systems. Based on these models, LID systems are developed for Indian languages using gaussian mixture model (GMM), gaussian mixture model- universal background model (GMM-UBM) systems in Ravi Kumar, Hari Krishna, and Anil Kumar (2015) and Maity, Anil Kumar, Rao, and Nandi (2012). Performance of LID systems in mobile environments is studied in Ravi Kumar, Hari Krishna, Bhupathiraju, Suryakanth, and Anil kumar (2015). Features derived from prosody of speech signal are explored for building LID systems in Rouas, Farinas, Pellegrino, and André-Obrecht (2003), Mary and Yegnanarayana (2008), Reddy, Maity, and Rao (2013) and Rao, Maity, and Reddy (2013). Though these systems have per-

* Corresponding author.
E-mail addresses: ravikumar.v@research.iiit.ac.in (R.K. Vuddagiri), hari.vydana@research.iiit.ac.in (H.K. Vydana), anil.vuppala@iiit.ac.in (A.K. Vuppala).

https://doi.org/10.1016/j.eswa.2018.06.004
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

formed reasonably better, they do not capture the information from temporal context which can be more discriminative for developing an LID system (Brij Mohan Lal, Hari Krishna, Anil Kumar, & Shrivastava, 2017). As i-vectors can better model the temporal context, LID systems developed using i-vectors as features have performed better (Dehak, Torres-Carrasquillo, Reynolds, & Dehak, 2011).

Developments in deep learning technologies have lead to the development of neural network architectures that can process the whole utterance to identify the language. These systems can better capture the long-term temporal patterns and have shown a greater promise in developing LID systems. Convolutional neural networks and recurrent neural networks, which can better capture long-term temporal cues have been explored for developing LID systems in Gonzalez-Dominguez, Lopez-Moreno, Sak, Gonzalez-Rodriguez, and Moreno (2014) and Lei, Ferrer, Lawson, McLaren, and Scheffer (2014). An explicit LID system has been developed in Brij Mohan Lal et al. (2017), the acoustic sequence is converted to a sequence of tokens using an acoustic model and the sequence of these tokens are modeled using recurrent neural network language modeling (RNNLM), SRI Language Modeling Toolkit (SRILM) based language models for discriminating the languages.

In Mounika, Sivanand, Lakshmi, Suryakanth, and Anil Kumar (2016), a feed-forward neural network with self-attention has been explored for developing LID systems. The network can operate in feed-forward fashion as well as capture the temporal context.In this study, DNN and DNN-with attention networks described in Mounika et al. (2016) have been explored for developing robust LID systems. In DNN based LID system, the decision of language is taken at every frame and the averaged decision is used as the language ID of the utterance, while DNN-WA network can give the utterance level decision of the language.

Traditionally, two major approaches have been explored to develop LID systems in varying background conditions. They are:

- Using a front-end enhancement method
- Inducing noise into the training corpus.

The former approach uses a clean dataset in building an LID system and expects testing utterance to have varying background noise. A front-end enhancement method is employed to enhance the noisy speech, and the enhanced version of the noisy speech is used to obtain the identity of a language using an LID system. Better performance of these approaches highly relies on the efficiency of enhancement methods employed. The later approach hypothesizes that the degradation in the performance of LID system operating in varying background environments is due to the mismatch in training and operating environments. To reduce this mismatch, noise is induced into the training corpus. Though these approaches have performed better, their performance is superior when there is a closer match between the training and operating environments. Moreover, the real-life environments are complex, mixed, time-varying and are hard to simulate. It is more likely to have the mismatch between the training and operating environments. With this motivation, an LID system is developed that can operate reasonably well even in the presence of a mismatch between training and operating environments.

In this work, to reduce the mismatch between training and operating environments multi-SNR models have been trained to develop LID systems. In training multi-SNR models, training data is augmented with multiple versions of noisy data synthetically generated by adding noise to the training corpus at different SNR levels, and the augmented dataset is used for developing LID systems. Training method plays a crucial role in the convergence and generalization and the performance of neural networks. In this study, curriculum learning based learning schedules have been explored in training multi-SNR models for developing robust LID system.Curriculum learning refers to the task of training a neural network with examples in some specific meaningful order rather than randomly sampling the training dataset. Better curriculum learning (CL) strategy helps the model to converge better and eases the optimization of the neural network (Bengio, Louradour, Collobert, & Weston, 2009; Braun, Neil, & Liu, 2016).The use of CL based training strategies have been explored for a variety of applications such as probabilistic linear discriminant analysis for noise robust speaker recognition (Ranjan & Hansen, 2017; Ranjan, Misra, & Hansen, 2017).

To assess the proposed methodâs performance, experimentation was done on two different and challenging LID datasets: 1. A dataset built from International institute of information technology-Hyderabad (IIIT-H) Indian language database for LID and 2. The Oriental Language Recognition AP17-OLR database. Thus, first, the proposed approach is tested in a practical environment; and second, the same is verified in a familiar and standard evaluation framework for the LID community. In both cases, it is focused on test utterances having 5s duration.

The remaining paper is organized as follows: Databases used in this work are described in Section 2. Section 3 describes the performances of various baseline LID systems and their performances in varying background environments. Section 4 presents the performance of LID systems using spectral subtraction and minimum mean square error enhancements as a front-end preprocessing method. Section 5 uses various curriculum learning strategies for developing robust LID systems. Conclusion and future scope are presented in Section 6.

2. Language identification databases

IIIT-H Indian language database and Oriental Language Recognition AP17-OLR database are used during the study. A brief description of both the databases are presented below.

The details of IIIT-H Indian language speech corpus database (Mounika et al., 2016) consists of 13 official Indian languages.They are Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Manipuri, Marathi, Odiya, Punjabi, Telugu, Tamil and Urdu. In this database, each language contains of minimum of 25 male and 25 female speakers from various age groups. The data volume collected for 12.5 h of data from each language, 9 h is used for training, 1 h is used for vaidation and 2.5 h of data is used for testing. Each utterance in the database is of 5 s duration, sampled at 16 kHz and a sample size of 16 bits.

Most of the Indian languages are phonetic in nature having an overlapping set of phonemes. At the same time, there are not many differences between the phone sets of individual Indian languages. Most of the Indian languages are originated from Sanskrit i.e., Devanagari script rooting in Brahmi script family (Raj et al., 2007). Most of the south Indian languages like Telugu, Kannada, and Malayalam follow Nandinagari script. North Indian languages like Hindi, Gujarati, and Marathi follow Devanagari script. This makes the task of Indian language identification even more challenging.

The Oriental Language Recognition(OLR) challenge AP17-OLR (Tang, Wang, Chen, & Chen, 2017; Wang, Li, Tang, & Chen, 2016) is jointly organized by Speech Ocean and the NSFC M2ASR project. This dataset includes various languages spoken in east, northeast and southeast Asia and consists of 10 languages - Cantonese in China Mainland and Hongkong, Uyghur, Kazakh and Mandarin in China, Korean in Korea, Japanese in Japan, Vietnamese in Vietnam, Russian in Russia and Indonesian in Indonesia. The data volume for each language is about 10 h of speech signals recorded in reading style. The signals are collected in different modes, with a sampling rate of 16 kHz with a size of 16 bits. For each language, 1800 utterances are selected as development set and the rest are used as the training set. The results are presented on development data only as test data key is not yet released. During the study, noise samples are generated using NOISEX database.

3. Baseline LID system

In this work, i-vector, DNN, and DNN-WA-based LID system have been developed, and these LID systems are briefly described in the following subsections. In this work, spectral features mel frequency cepstral coefficients (MFCC) with 39 dimension are used as features for developing LID system. In this study, the performance of LID system is presented in terms of equal error rate (EER). The EER is used as a performance metric considering only scores of each individual language and average for the same is computed.

3.1. LID system using i-vector

The i-vector or total variability space approach has become state-of-the-art in speaker verification (Dehak, Kenny, Dehak, Dumouchel, & Ouellet, 2011) and, it has been shown that the same can be successfully applied to language identification (Martınez, Plchot, Burget, Glembek, & Matejka, 2011). The i-vector based approach maps the sequence of frames from a given utterance into a low-dimensional feature space with fixed dimension. It is also referred to as the total variability space, based on a factor analysis technique. The approach provides an elegant way to reduce high-dimensional sequential input data to a low-dimensional fixed-length feature vector while retaining most relevant information.

The parameters of Gaussian mixture model (GMM) are estimated from features using expectation maximization (EM) algorithm (Torres-Carrasquillo et al., 2002). In maximum, a posteriori (MAP) adaptation (Castaldo, Colibro, Dalmasso, Laface, & Vair, 2007), is employed to adapt the features to train/test the parameters of the universal background model (UBM). The basic idea of i-vector space is adapting UBM for a large database in training, collected from many languages, to represent the language-independent distribution of the features in the utterance level. A 2048 mixture is trained for UBM, with diagonal covariance matrices, based on the assumption that the dimensions of feature vectors are independent.

The mean vectors of all the component densities of GMM are concatenated to form a 79,872-dimensional (2048 × 39) GMM super-vector. The GMM super-vector contains language- and channel-dependent specific information. In order to reduce the dimensionality of the super vector, it is typically projected to a low dimensional subspace, termed as i-vector subspace as follows:

$$L = L_U + Tw \tag{1}$$

The identity vector or i-vector (Dehak, Torres-Carrasquillo, et al., 2011) in above equation, where L is the GMM super-vector of an utterance, L_U is the language and channel-independent of the mean super-vector, T is a low-rank i-vector matrix (T-matrix) and w is a weight vector with a standard normal prior. Sufficient matrices of T-matrix is used to train the adapted GMM models, obtained from a large amount of background data, by employing the EM algorithm as described in Martınez et al. (2011). L_U is mean of super-vector UBM (39 × 2048). A total variability subspace of 100-dimensional is trained, using all available data from the same databases considered for UBM training. The resultant T -matrix (100 × 39 × 2048) is used to obtain the low dimensional representation of language-specific information in the form of i-vectors (Dehak, Kenny, et al., 2011). And w is a weight 100 × 13 × 2746 (100 is i-vector dimensional reduction, 13 is number of languages of IIIT-H (it is 10 for AP17-OLR) and 2746 is number of training utterances in each language. The performance of i-vector is presented in row 2 of Table 1.

3.2. LID system using DNN

In literature, LID systems have been developed using convolutional neural networks (Ganapathy et al., 2014) and shallow architecture (Gonzalez-Dominguez, Lopez-Moreno, Moreno, & Gonzalez-Rodriguez, 2015). In this work, deep neural networks are used for developing an LID system. Deep neural networks have performed superior compared to i-vector in Lopez-Moreno, Gonzalez-Dominguez, and Plchot (2014) using a corpus of 200 h, both DNN and i-vector based LID systems have performed comparatively equal with a corpus of 100 h.

In this work, DNN based LID system has been developed similar to the system presented in Mounika et al. (2016) and the results are tabulated in Table 1. The network comprises of four hidden layers and each layer comprising of 700, 500, 200 and 100 units with rectified linear unit (ReLU) activation functions and output is a softmax layer.

The network is trained with stochastic gradient descent. The learning rate, momentum factor, and mini-batch size used during the training are 0.001, 0.98 and 200, respectively. The network is trained with categorical entropy objective function. The frame level evidence obtained by the network are averaged over the utterance to obtain the language identity of the utterance. The performance of LID developed using DNN is presented in Table 1. Multiple experiments have been performed and the performance obtained with best-performing hyperparameters have been presented in row 3 of Table 1.

3.3. LID system using DNN with attention

In DNN-based LID system, the decision on language is taken at every frame, while language identification is usually assigned to a whole utterance. The language discriminative information from the long-term temporal patterns of the speech cannot be modeled using a DNN. However, this discriminative information (Gonzalez-Dominguez et al., 2014) can be modeled using long-short-term memory networks (LSTMs), but due to sequential nature of these networks, they are computationally slow and are not parallelizable. DNN-WA network operates in a feed-forward fashion as well as captures the temporal context. In this work, DNN-WA model described in Mounika et al. (2016) has been implemented. The network comprises of 4 hidden layers with each layer comprising of 700, 500, 200 and 100 units with ReLU activation functions and the attention layer comprises of A multilayer perceptron (MLP) with a single hidden layer. The entire network is trained end-to-end with categorical entropy function. The network is trained with stochastic gradient descent, learning rate, momentum factor used during the training are 0.001 and 0.98. The network is trained with variable batch size and for every utterance, the batch size is same as the length of an acoustic sequence. Both DNN and DNN-WA models use spectral features of 39-dimensions are used as inputs and the number of languages classes in respective databases are used as output dimensions i.e., 10, 13 in AP17-OLR, IIIT-H datasets respectively.

The DNN-WA architecture is shown in Fig. 1, the output function for hidden layer is given by

$$H = f(x_t, h_t) \tag{2}$$

where x_t is sequence of input feature vectors $\{x_1, x_2, \ldots., x_T\}$ and the sequence of hidden state vectors is $\{h_1, h_2, \ldots.h_T\}$. The output of hidden layer h_t, is computed by forward pass through regular DNN and a self attention is computed on these hidden features.

The attention mechanism $a(h_t)$ shown in Fig 1. is computed using a single layer perceptron and then a softmax operation is preformed to normalize the values between zero and one.

$$\beta = tanh(W_{wa}h_t + b_{wa}) \tag{3}$$

Table 1
EER of baseline i-vector,DNN and DNN-WA LID systems developed using IIIT-H database .

Language	Ass	Ben	Guj	Hin	Kan	Mal	Man	Mar	Odia	Pun	Tam	Tel	Urd	Average
i-vector	5.12	13.01	13.26	19.94	14.64	7.51	4.4	10.16	6.49	7.16	19.9	4.99	5.73	10.18
DNN	3.53	7.03	9.25	22	15.8	18.8	6	13.6	5.46	4.68	20	3.18	7.47	10.52
DNN-WA	5.64	6.63	6.8	18.8	7.58	9.63	6.52	7.20	3.59	6	12.6	4.72	6.80	7.88

Table 2
EER of baseline i-vector,DNN and DNN-WA LID systems developed using OLR database.

Language	Kazak	Tibet	Uyghu	ct-cn	id-id	jp-ja	ko-kr	ru-ru	vi-vn	zh-cn	Average
i-vector	6.51	4	4.52	7.65	21.4	8.15	12.15	3.56	8.56	9.22	8.57
DNN	18.8	2.72	6.66	17.14	40.8	10.6	27.38	10.08	4.29	18.57	15.72
DNN-WA	7.85	5	5.19	9.8	29.4	11.16	17	8.77	8	11.35	11.35

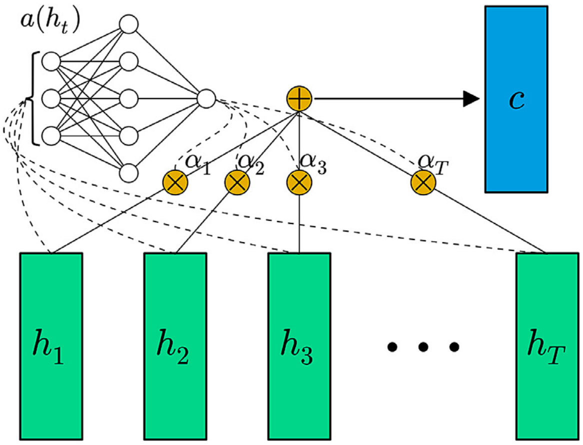

Fig. 1. Deep neural network with attention model (Raffel & Ellis, 2015).

Table 3
EER of baseline system in varying background environments.

EER of LID using IIIT-H database						
	i-vector	DNN	DNN-WA	i-vector	DNN	DNN-WA
Clean	10.18	10.52	7.88	–	–	–
		White noise			Vehicle noise	
20 dB	25.45	26.95	24.16	17.13	17.47	12.94
15 dB	33.97	34.37	32.51	18.83	19.32	14.6
10 dB	41.01	41.06	39.25	21.15	21.44	16.92
5 dB	44.48	45.69	43.65	23.5	24.49	20.62
		EER of LID using OLR database				
	i-vector	DNN	DNN-WA	i-vector	DNN	DNN-WA
Clean	8.57	15.72	11.35	–	–	–
		White noise			Vehicle noise	
20 dB	38.24	39.46	33.39	31.45	32.26	24.58
15 dB	41.12	42.47	36.33	34.98	35.46	27.97
10 dB	43.56	44.87	39.79	38.11	39.51	31.51
5 dB	46.11	47.67	43.12	42.78	43.47	35.09

$$\alpha = softmax(\beta) \tag{4}$$

In the above equations, α is referred to as attention vector, and W_{wa}, b_{wa} are the parameters of the attention hidden weights, the entire network is optimized along with other parameters of using backpropagation algorithm.

The attention based model computes a “context vector ” C_t

$$C_t = \sum_{j=1}^{T} \alpha H \tag{5}$$

where C_t content weighted mean, the state sequence of H, T total number of time steps in the input sequence.

The output is computed by transforming the context vector C_t using output layer weight V followed by softmax operation

$$y_o = softmax(VC_t + b_o) \tag{6}$$

where b_o is the output layer bias. Note that for the entire input utterance x_t, only a single decision vector y_o is predicted. The performance of LID system developed with DNN-WA network is presented in row 4 of Table 1.

The performance of baseline LID system is shown in Table 1. In column 1 of Table 1 are various LID system developed using i-vector, DNN and DNN-WA respectively. Column 2–14 are EERs attained for each language. Column 15 is the average EER for all the languages. Similarly the performance of baseline LID system is observed on OLR dataset in Table 2.

Motivated by the studies (Lopez-Moreno et al., 2014) and based on the discriminative nature of DNNs, which could complement the i-vector generative approach, DNNs are adapted to work at the acoustic frame level to perform LID. Particularly, in this work, we build, explore and experiment with DNN configurations and compare the obtained results with several state-of-the-art i-vector based systems trained from exactly the same acoustic features.

The performance of i-vector and DNN system are comparable. The results obtained are in accordance with the results shown in Lopez-Moreno et al. (2014), however, the performance of DNN-WA is significantly higher. The better performance of DNN-WA based LID is due to better utilization of temporal contextual information.

3.4. Performance of LID in varying background environments

To study the performance of LID system in varying background environments, LID systems in this study are developed using clean data and tested with noisy speech synthetically generated at different SNR levels. In this study, white noise and vehicle noise samples from the NOISEX dataset are added to the test set at different SNR levels from 5 dB to 20 dB at steps of 5 dB.

The performance of the LID system when operated at different SNR levels is represented in Table 3. Column 1 is the SNR level of the test utterance. Columns 2–4 is the performances of LID systems varying background environments in presence of white noise for i-vector, DNN and DNN-WA respectively. Similarly, the columns 5–7 is the performance of LID system for vehicle noise. From Table 3, it is evident that a degradation in the performance of LID systems can be observed. The degradation may be majorly attributed to a mismatch between training and operating environments i.e., the LID systems are developed using clean speech and tested with noisy utterances. As white noise affects all the frequency bands, the degradation in the performance in the presence of white noise is high compared to the vehicle noise.

Table 4
EER of LID systems when operated in noisy environments using a front-end enhancement methods.

EER of LID using IIIT-H database						
	Enhanced with SS			Enhanced with MMSE		
	i-vector	DNN	DNN-WA	i-vector	DNN	DNN-WA
			White noise			
20 dB	20.45	21.43	23.12	18.11	18.37	19.53
15 dB	24.15	25.14	28.68	20.98	21.3	25.03
10 dB	29.12	30.92	35.59	25.95	26.04	30.64
5 dB	35.15	36.18	41.38	31.14	32.65	37.27
			Vehicle noise			
20 dB	13.95	14.5	15.03	13.11	14.9	12.64
15 dB	15.12	15.61	13.33	14.34	15.6	13.33
10 dB	16.10	16.39	14.35	15.95	16.39	14.35
5 dB	17.05	17.41	16.36	16.65	17.41	15.16
			EER of LID using OLR database			
			White noise			
	i-vector	DNN	DNN-WA	i-vector	DNN	DNN-WA
20 dB	30.56	31.19	27.67	29.11	28.54	25.36
15 dB	33.76	34.04	30.19	32.47	33.46	28.17
10 dB	36.78	37.30	33.29	34.85	35.41	31.64
5 dB	39.71	40.86	36.87	36.45	37.64	34.62
			Vehicle noise			
20 dB	27.14	28.99	22.34	25.45	26.04	20.39
15 dB	30.56	31.46	24.97	26.57	27.54	22.81
10 dB	34.45	35.65	27.16	29.15	30.38	24.85
5 dB	36.45	37.48	31.71	33.15	34.15	28.73

4. Performance of LID systems with spectral subtraction and minimum mean square error as front-end enhancement scheme

Due to the mismatch between training and operating environments, a degradation in performance can be observed. To study the performance of LID systems in noisy environments, two different enhancement methods are employed viz., spectral subtraction (SS) (Boll, 1979) and minimum mean square error (MMSE) (Martin, 2005). The LID systems are developed using the clean speech data. The noisy utterances are initially enhanced using an enhancement method and the enhanced version of the noisy utterance is used for obtaining the language identity from an LID system. The performances of these LID systems with front-end enhancement methods are presented in Table 4.

The LID systems presented in Table 4 are trained on clean speech corpus and tested with the enhanced version of the noisy utterances. The performance of the LID systems using a front-end enhancement method at different SNR levels is presented in Table 4. Column 1 is the SNR level of the test utterance. Columns 2–4 and 5–7 are the performances of LID systems with spectral subtraction and MMSE as front-end enhancement scheme. From Table 4, it can be observed that the performance of LID systems is better with front-end enhancement.

From the Table 4, it can be observed that the performance of LID is influenced by the efficiency of enhancement method. At low SNRs, the efficiency of enhancement approach is low and the performance of the LID is poor. Though the use of front-end enhancement an improvement in the performance is observed, it is much lesser than the performance of LID system attained when operated in clean environments.

5. Multi-SNR models with different curriculum learning strategies

In this work, to reduce the mismatch between training and operating environments, noise is induced into the training corpus and the augmented dataset is used for developing LID systems. Multiple LID systems are developed using datasets with induced noise at different SNR levels and their performance is presented in Tables 5 and 6. The LID systems presented in this study are trained at a specific SNR and tested with data at different SNR levels.

The performance of LID system is studied in matched and mismatched scenarios are shown in Tables 5 and 6. In the matched scenario, LID system is trained and tested using speech samples at same SNR level, while in a mismatched scenario, the LID systems are trained and operated at different SNR levels. Row 1 is the model employed for developing LID system. Column 1 is the SNR level of the training data, while row 2 is the SNR level of the testing speech sample. Rows 4–7 and 9–12 are the performances of LID for white and vehicle noise respectively. In Tables 5 and 6, the performances of LID systems obtained in the matched scenario are bolded.

From Tables 5 and 6, it is evident that the performance of multi-SNR LID systems is superior to LID systems with a front-end enhancement method. It can be noted that inducing the noise in training corpus reduces the mismatch between training and operating environments and has lead to better performance. Also, a superior performance can be observed when there is a closer match between the training and operating environments i.e., best performance is seen when trained and tested at same SNR level. It can be observed that a decline in the performance can be noted when there is a larger mismatch in SNR levels of training and testing utterances i.e., the model trained with 20 dB SNR when tested with a test utterance of SNR of 5 dB, the performance of LID system drastically degraded. But the real-life environments are complex and mixed. It is more likely that there is always a mismatch between training and operating environments. It is ideally desired to have an LID, which is less sensitive to mismatch in the training and operating environments. Similar trend in the performances can also be noted using OLR corpus for developing LID systems.

In training multi-SNR, models various curriculum learning strategies are explored. Curriculum learning strategies can be task specific, and the network can better generalize with an apt-curriculum learning strategy. In this study, three different curriculum learning strategies (CLS) are explored to develop robust LID systems. They are CL-full SNR, CL-high SNR, and CL-low SNR. In these strategies, the augmented dataset is formed by arranging the corpus in a sequence of SNR levels as shown in Table 7.

The SNR level of the corpus used in training the network is maintained in a sequence as shown in Table 7. At every stage, training is continued till an increase in validation accuracy gets saturated and the best weights are stored and used in the next stage. After every stage, the learning rate is halved. The results obtained by various curriculum learning strategies are presented in Tables 8 and 9.

The performance of various LID systems developed following different curriculum learning strategies is presented in Tables 8 and 9. Column 1 is the LID system developed with learning strategies described in Table 7. Column 2–6, 7–11 and 12–16 are the performances of LID systems developed using i-vector, DNN and DNN-WA models. Row 4–7 and 9–12 are the performances of LID systems developed by inducing white and vehicle noises in obtaining the augmented dataset. In Table 8, multi-SNR models are trained by randomly sampling the augmented training dataset. The performance of these models is comparable to the performance of LID with matched training (i.e., model trained at the same SNR as operating environment), in certain cases, the performance is slightly higher than matched training as the augmented dataset is 5–6 times larger than the original dataset. From Table 8, it is evident that the LID systems trained using the learning strategies full-SNR and High-SNR have performed poorly compared to the multi-SNR training and they did not generalize well across different SNR levels.

Table 5
EER of LID systems when operated in matched and mismatched SNR levels using IIIT-H database.

EER of LID using IIIT-H database															
Model	i-vector					DNN					DNN-WA				
Train SNR	Clean	20	15	10	5	Clean	20	15	10	5	Clean	20	15	10	5
								White noise							
20	30.15	**12.15**	19.85	30.56	39.36	31.16	**12.84**	20.85	31.91	39.98	12.59	**9.05**	13.5	24.58	34.5
15	34.16	20	**14.26**	28.56	30.25	35.15	20.41	**14.57**	27.78	31.04	17.58	11.67	**9.25**	12.5	33.4
10	37.45	30.84	23.45	**16.47**	24.68	38.5	31.61	24.29	**16.92**	25.48	23.57	17.55	12.2	**9.5**	22.5
5	39.46	36.47	34.16	28.16	**17.23**	40.6	37.89	35.49	29.15	**18.52**	29.96	22.32	18.42	12.6	**9.75**
								Vehicle noise							
20	16.14	**9.84**	14.68	22	25.18	16.57	**10.53**	15.05	22.83	26.73	9.72	**8.16**	10.79	20.04	22.53
15	18.99	11.89	**10.99**	14.56	21.82	19.92	12.37	**11.99**	14.93	22.27	13.77	10.42	**8.01**	11.35	18.69
10	23	18.76	13	**11.23**	20.11	23.99	19.27	13.31	**11.71**	20.32	14.26	12.58	11.36	**8.63**	14.56
5	27.42	16.59	14.11	12.94	**12.16**	27.67	17.02	14.88	13.05	**12.75**	16.16	14.27	12.96	11.12	**8.19**

Table 6
EER of LID systems when operated in matched and mismatched SNR levels using OLR database.

EER of LID using OLR-dataset															
Model	i-vector					DNN					DNN-WA				
Train SNR	Clean	20	15	10	5	Clean	20	15	10	5	Clean	20	15	10	5
								White noise							
20	41.11	**19.56**	24.23	33.66	42.67	41.42	**20.20**	25.57	33.74	43.33	19.11	**12.92**	16.32	23.75	32.01
15	44.23	25.72	**21.93**	29.13	37.56	44.85	26.16	**22.15**	29.48	38.48	24.87	12.76	**12.10**	16.18	26.80
10	45.22	33.46	29.19	**23.43**	34.16	45.83	34.89	29.85	**24.85**	34.75	32.26	19.21	14.78	**12.79**	19.51
5	43.56	38.16	36.47	33.75	**28.76**	44.79	39.43	38.02	34.39	**29.05**	32.69	23.54	20.20	16.80	**15.14**
								Vehicle noise							
20	32.0	**15.91**	18.14	19.62	26.11	32.14	**16.13**	18.94	20.09	26.67	19.89	**12.40**	13.49	17.70	24.09
15	34.16	16.94	**14.73**	16.33	22.75	34.91	17.38	**15.95**	16.83	23.44	25.19	15.8	**13.33**	11.95	14.91
10	38.11	18.95	17.11	**14.35**	18.01	37.24	19.26	17.57	**15.57**	18.18	26.96	17.62	14.45	**11.83**	12.16
5	39.11	25.45	23.67	20.67	**18.37**	39.72	24.46	22.53	19.96	**17.77**	28.57	19.90	16.52	13.14	**11.48**

Table 7
Sequence of steps involved in various curriculum learning strategies.

CLTS	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
CL-Full SNR	clean	20 dB	15 dB	10 dB	5 dB
CL-High SNR	20 dB	15 dB	10 dB	5 dB	-
CL-Low SNR	5 dB	10 dB	15 dB	20 dB	clean

In this learning strategies, the training progresses from low SNR levels to high SNR levels as described in Table 7. The models perform better while testing at low SNR levels, but could not generalize to high SNR levels. From Table 8, it can be noted that the performance of LID trained with CL-low SNR has performed better than the multi-SNR models. The LID systems developed using CL-low learning strategy has generalized over the test utterances at different SNR levels. The degradation in the performance of LID systems due to a mismatch between training and operating environments is reduced using CL-low learning strategy. From Tables 8 and 9, CL-low SNR has performed better that other learning strategies and the model have shown better generalization across SNR levels. A similar trend can be observed using i-vector, DNN and DNN-WA, even in the presence of noise DNN-WA model can have access to less corrupted regions of speech for detecting the language yielding significantly better performance compared to i-vector and DNN based LID system. The proposed method has shown better performances across the two datasets used.

Though the proposed method has shown better generalization in varying background environments, large computational resources are required for creating multiple copies of data injected with different noises. The training procedure of this approach is iterative by changing noise levels in the dataset. In-spite of multi-SNR training with curriculum learning, the operating environments with entirely different noise characteristics can create degradation in performance.

6. Conclusion and future scope

This work attempts to model automatic language identification system using i-vector, DNN and DNN-WA with different training methods to improve the noise robustness. In this work, white and

Table 8
EER of LID systems developed using various curriculum learning strategies using IIIT-H database.

Model	i-vector					DNN					DNN-WA				
Learning schedule	Clean	20	15	10	5	Clean	20	15	10	5	Clean	20	15	10	5
								White noise							
Multi SNR	13.22	17.94	19.65	23.76	28.11	13.9	18.5	20.15	24	28.42	12.52	17.54	19.45	22.5	25.83
CL-Full-SNR	43.11	36.77	33.34	27.11	22.64	42.33	37.83	34.9	27.93	23.48	30.26	37.94	23.6	20.05	18.18
CL-High-SNR	42.98	39.45	36.12	29.11	**22.34**	41.14	38.33	35.85	28.75	**22.35**	29.25	25.02	22.83	19.92	**16.5**
CL-Low-SNR	**10.74**	**11.45**	**13.2**	**15.11**	23.45	**11.51**	**12.67**	**14.4**	**16.65**	22.78	**8.25**	**11.87**	**13.3**	**15.02**	16.91
								Vehicle noise							
Multi SNR	10.73	16.28	17	20.16	20.49	11.11	16.57	17.8	19.49	21.3	10.5	11.62	13.25	17	19.5
CL-Full-SNR	30.63	27.83	24.76	22.19	18.78	31.19	28.06	25.52	22.56	19.54	24.7	22.9	20.23	18.38	16.3
CL-High-SNR	29.56	27.32	23.82	21.35	17.94	30.2	27.68	24.53	21.79	18.54	23.05	21.81	19.95	17	15.9
CL-Low-SNR	**9.84**	**11.24**	**12.73**	**15.21**	**17.22**	**10.18**	**11.79**	**13.2**	**15.37**	**17.73**	**7.82**	**9.8**	**11.3**	**12.8**	**14.9**

Table 9
EER of LID systems developed using various curriculum learning strategies using OLR database.

Model	i-vector					DNN					DNN-WA				
Learning schedule	Clean	20	15	10	5	Clean	20	15	10	5	Clean	20	15	10	5
							White noise								
Multi SNR	17.85	22.11	27.56	30.46	35	18.23	22.45	28.56	31.22	35.89	14.25	19.87	24.52	27.54	34.25
CL-Full-SNR	42.63	39.31	35.49	31.76	28.35	43.03	39.92	36.05	32.70	29.98	38.14	34.33	30.27	27.08	24.02
CL-High-SNR	41.59	38.18	34.96	30.49	27.67	42.60	38.46	35.22	31.1	28.54	37.14	34	29.98	26.84	23.64
CL-Low-SRN	**16.11**	**18.62**	**21.93**	**23.46**	**27.86**	**16.24**	**19.52**	**22.26**	**24.49**	**28.24**	**12.18**	**16.36**	**18.88**	**20.65**	**23.11**
							Vehicle noise								
Multi SNR	16.05	18.73	22.16	25.56	31.89	16.23	19.25	22.97	26.33	32.24	12.56	16.24	18.54	22.41	27.56
CL-Full-SNR	40.13	35.42	31.67	29.46	25.94	39.33	34.25	30.67	28.60	24.72	33.15	29.12	25.02	22.81	19.82
CL-High-SNR	39.75	34.25	30.76	28.79	24.18	38.54	33.75	29.33	27.13	23.31	32.91	28.97	24.06	21.95	18.32
CL-Low-SNR	**13.45**	**16.21**	**17.86**	**19.56**	**22.63**	**14.75**	**16.81**	**18.02**	**20.58**	**23.11**	**11.33**	**12.11**	**14.63**	**16.44**	**18**

vehicle noise are used to study the performance of LID in varying background environments. A degradation in the performance of LID is observed when there is a mismatch between the training and operating environments. To reduce the mismatch between training and operating environments, multi-SNR training is explored by augmenting the original training data with multiple versions of training data with induced noise at different SNR levels. In training the multi-SNR models, various curriculum learning strategies are explored for developing robust LID systems. Curriculum learning is a learning strategy in which neural network is trained with examples in a specific order. In this work, three strategies CL-full-SNR, CL-high-SNR and CL-low-SNR where the network is trained by examples in a sequence of increasing and decreasing SNR levels are used. The networks trained using CL-low-SNR strategy (i.e., by presenting examples in an increasing order of SNR levels) has performed better compared to the other strategies, has shown better generalization over different SNR levels.

Developing LID systems that can operate in mobile environments has to be explored. LID systems that can operate on short utterances have to be explored. Robust approaches that can model both short-term, as well as long-term temporal patterns, has to be investigated further for building LID systems. A feature-model pair which can efficiently utilize longterm temporal information for developing LID systems can perform better in noisy environments. As the environmental noise do not equally effect all the regions of an utterance, the regions in the utterance which are less effected by noise could be more informative of language, detecting and using them could result in a robust LID system.

Acknowledgements

The authors would like to thank Science & Engineering Research Board (SERB) for funding Language Identification in Practical Environments (YSS/2014/000933) project.

References

Ambikairajah, E., Li, H., Wang, L., Yin, B., & Sethu, V. (2011). Language identification: A tutorial. *IEEE Circuits and Systems Magazine, 11*(2), 82–108. doi:10.1109/MCAS.2011.941081.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proc. of the 26th annual international conference on machine learning* (pp. 41–48). ACM.

Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 27*(2), 113–120.

Braun, S., Neil, D., & Liu, S.-C. (2016). A curriculum learning method for improved noise robustness in automatic speech recognition. arXiv:1606.06864.

Brij Mohan Lal, S., Hari Krishna, V., Anil Kumar, V., & Shrivastava, M. (2017). Significance of neural phonotactic models for large-scale spoken language identification. In *Proc. neural networks (ijcnn), 2017 international joint conference on, May*. IEEE.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., & Vair, C. (2007). *Acoustic language identification using fast discriminative training*. In Proc. International Speech Communication Association.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., & Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing, 19*(4), 788–798.

Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D., & Dehak, R. (2011). Language recognition via i-vectors and dimensionality reduction. In *Proc. twelfth annual conference of the international speech communication association*.

Ganapathy, S., Han, K., Thomas, S., Omar, M., Segbroeck, M. V., & Narayanan, S. S. (2014). Robust language identification using convolutional neural network features. In *Proc. fifteenth annual conference of the international speech communication association* (pp. 1846–1850).

Gonzalez-Dominguez, J., Lopez-Moreno, I., Moreno, P. J., & Gonzalez-Rodriguez, J. (2015). Frame-by-frame language identification in short utterances using deep neural networks. *Neural Networks, 64*, 49–58.

Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., Gonzalez-Rodriguez, J., & Moreno, P. J. (2014). Automatic language identification using long short-term memory recurrent neural networks.. In *Proc. interspeech* (pp. 2155–2159).

Lei, Y., Ferrer, L., Lawson, A., McLaren, M., & Scheffer, N. (2014). Application of convolutional neural networks to language identification in noisy conditions. *Proc. Odyssey-14, Joensuu, Finland*.

Li, H., Ma, B., & Lee, C.-H. (2007). A vector space modeling approach to spoken language identification. *IEEE Transactions on Audio, Speech, and Language Processing, 15*(1), 271–284.

Lopez-Moreno, I., Gonzalez-Dominguez, J., & Plchot, O. (2014). Automatic language identification using deep neural networks. In *Proc. ieee international conference on acoustics, speech and signal processing (icassp)*.

Ma, B., Guan, C., Li, H., & Lee, C.-H. (2002). Multilingual speech recognition with language identification. In *Proc. interspeech*.

Maity, S., Anil Kumar, V., Rao, K. S., & Nandi, D. (2012). IITKGP-MLILSC speech database for language identification. In *Proc. national conference on communication* (pp. 1–5). IEEE.

Martin, R. (2005). *Statistical methods for the enhancement of noisy speech*. Springer.

Martınez, D., Plchot, O., Burget, L., Glembek, O., & Matejka, P. (2011). Language recognition in i-vectors space. *In Proc. INTERSPEECH, Firenze, Italy*, 861–864.

Mary, L., & Yegnanarayana, B. (2008). Extraction and representation of prosodic features for language and speaker recognition. *Speech Communication, 50*(10), 782–796.

Mounika, K. V., Sivanand, A., Lakshmi, H., Suryakanth, V. G., & Anil Kumar, V. (2016). An investigation of deep neural network architectures for language recognition in Indian languages. In *In proc. interspeech* (pp. 2930–2933).

Muthusamy, Y. K., Barnard, E., & Cole, R. A. (1994). Reviewing automatic language identification. *IEEE Signal Processing Magazine, 11*(4), 33–41.

Raffel, C., & Ellis, D. P. W. (2015). Feed-forward networks with attention can solve some long-term memory problems. CoRR, abs/1512.08756.

Raj, A. A., Sarkar, T., Pammi, S. C., Yuvaraj, S., Bansal, M., & Prahallad, K. (2007). Text processing for text-to-speech systems in Indian languages. In *Ssw* (pp. 188–193).

Ranjan, S., & Hansen, J. H. (2017). Curriculum learning based approaches for noise robust speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Ranjan, S., Misra, A., & Hansen, J. H. (2017). Curriculum learning based probabilistic linear discriminant analysis for noise robust speaker recognition. In proc. interspeech 2017, (pp. 3717–3721).

Rao, K. S., Maity, S., & Reddy, V. R. (2013). Pitch synchronous and glottal closure based speech analysis for language recognition. *International Journal of Speech Technology, 16*(4), 413–430.

Ravi Kumar, V., Hari Krishna, V., Bhupathiraju, J. V., Suryakanth, V. G., & Anil kumar, V. (2015). Improved language identification in presence of speech coding. In *Proc. international conference on mining intelligence and knowledge exploration, india* (pp. 312–322). Springer.

Ravi Kumar, V., Hari Krishna, V., & Anil Kumar, V. (2015). Significance of GMM-UBM based modelling for Indian language identification. *Procedia Computer Science, 54*, 231–236.

Reddy, V. R., Maity, S., & Rao, K. S. (2013). Identification of Indian languages using multi-level spectral and prosodic features. *International Journal of Speech Technology, 16*(4), 489–511.

Rouas, J.-L., Farinas, J., Pellegrino, F., & André-Obrecht, R. (2003). Modeling prosody for language identification on read and spontaneous speech. In *Acoustics, speech,* *and signal processing, 2003. proceedings.(icassp'03). 2003 ieee international conference on: 6* (pp. I–40). IEEE.

Tang, Z., Wang, D., Chen, Y., & Chen, Q. (2017). Ap17-olr challenge: Data, plan, and baseline. arXiv:1706.09742.

Torres-Carrasquillo, P. A., Singer, E., Kohler, M. A., Greene, R. J., Reynolds, D. A., & Deller Jr, J. R. (2002). Approaches to language identification using gaussian mixture models and shifted delta cepstral features. In *Proc. interspeech*.

Wang, D., Li, L., Tang, D., & Chen, Q. (2016). Ap16-ol7: A multilingual database for oriental languages and a language recognition baseline. In *Signal and information processing association annual summit and conference (apsipa), 2016 asia-pacific* (pp. 1–5). IEEE.

Zissman, M. A. (1996). Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on Speech and Audio Processing, 4*(1), 31.

Zissman, M. A., & Berkling, K. M. (2001). Automatic language identification. *Speech Communication, 35*(1), 115–124.