NOISE-ROBUST SPOKEN LANGUAGE IDENTIFICATION USING LANGUAGE RELEVANCE FACTOR BASED EMBEDDING

Muralikrishna H, Shikha Gupta, Dileep Aroor Dinesh, Padmanabhan Rajan

School of Computing and Electrical Engineering, Indian Institute of Technology Mandi, India

ABSTRACT

State-of-the-art systems for spoken language identification (LID) use i-vector or embedding extracted using a deep neural network (DNN) to represent the utterance. These fixed-length representations are obtained without explicitly considering the relevance of individual frame-level feature vectors in deciding the class label. In this paper, we propose a new method to represent the utterance that considers the relevance of the individual frame-level features. The proposed representation can also preserve the locally available LID-specific information in the input features to some extent. To better utilize the local-level information in the new representation, we propose a novel segment-level matching kernel based support vector machine (SVM) classifier. The proposed representation of the utterance based on the relevance of frame-level features improves the robustness of the LID system to different background noise conditions in the speech. The experiments conducted on speech with different background conditions show that the proposed approach performs better than state-of-the-art approaches in noisy speech and performs similarly to the state-of-the-art systems in clean speech condition.

Index Terms— spoken language identification, language relevance factor, self-attention, support vector machine

1. INTRODUCTION

The success of a language identification (LID) system mainly depends on the type of features used and the method used to model the frame-level features to extract the utterance-level representation. State-of-the-art LID systems use frame-level features like mel-frequency cepstral coefficients (MFCCs) or bottleneck features (BNFs) to represent the speech utterance [1, 2, 3, 4]. But, these methods lead to a variable-length representation of speech, posing a challenge in building the back-end classifier. Typically, i-vector analysis is performed on these frame-level feature vectors to obtain an utterance-level representation [2, 3, 5, 6]. However, i-vectors are computed in an unsupervised fashion without using language labels. Hence the techniques such as linear discriminant analysis (LDA) and within-class covariance normalization (WCCN) are necessary before feeding them to a back-end classifier.

This motivated researchers to use deep neural network (DNN) to extract fixed-length embeddings of speech [7, 8, 9, 10] which allows explicit use of class information. The work in [7] and [9] uses a fixed-dimensional representation called x-vector. The DNN used for extracting x-vector has a statistics pooling layer to compute the mean and standard deviation of all frame-level features. A similar approach is reported in [10] to extract a fixed-length representation called DNN-based embeddings. However, all these methods do not explicitly consider the relevance of individual frame-level feature vectors in deciding the language label during the utterance-level embedding extraction. Though it can be assumed that LID-specific contents are equally distributed among all frame-level features in a clean speech sample, this is not the case when the speech contains real-world background noise. In the presence of background noise, the amount of LID-specific contents carried by different parts of the speech sample can vary significantly [11].

To address this issue, self-attention based DNN (SA-DNN) can be used, which computes the utterance-level embedding of the speech as a weighted average of frame-level feature vectors [11, 12, 13, 14]. The SA-DNN assigns a relevance factor (attention weight) to each frame-level feature vector indicating its importance (relevance) in deciding the class label. This enables the system to automatically ignore the parts of the speech sample that might have been significantly affected by noise.

Apart from these, some recent papers have shown that utilizing the locally available information in the input feature vectors obtained by analyzing them at a segment-level granularity can lead to better performance in applications like speech emotion recognition, speaker identification [15], [16], and LID [17]. Unlike the classifiers used in traditional approaches that uses the class-specific information available in the utterance-level representation (like i-vector and x-vector [5, 7]) to predict the class label, the classifiers used in [15, 16, 17] uses the class-specific information available at a more finer segment-level granularity obtained by dividing the utterance into smaller segments. The sequence-kernel based SVM classifiers used in these systems [15, 16, 17] are designed to preserve the temporal order of the input feature vectors to enable them to utilize the local-level information

in the input. However, like i-vector and x-vector based approaches, these sequence-kernel based approaches do not consider the relevance of individual frame-level feature vectors in deciding the class label.

Motivated by these, we propose a new representation for the speech utterance that considers the relevance of individual frame-level features in deciding the language label along with preserving the locally available LID-specific contents in them. We propose to divide the input sequence of feature vectors into a predetermined number of segments and then select a subset of frames that are significantly relevant for LID in each segment. The relevance of individual feature vectors are decided based on their language relevance factor (LRF) values estimated using an SA-DNN. In each segment, a compact representation of the selected feature vectors is computed to obtain a fixed-length segment-level embedding. The speech utterance is then represented as a sequence of these segment-level embeddings. One advantage of this approach is that, it enables us to improve the noise-robustness of the LID system by dynamically ignoring the feature vectors that might have been significantly affected by noise. One more advantage of our proposed approach is that, by maintaining the order of the segment-level embeddings, the proposed language relevance factor (LRF) based representation preserves the locally available information in the input feature vectors to some extent. To utilize the local-level information in this representation, we propose a novel segment-level matching kernel based SVM classifier.

The main contributions of this paper are: 1) a DNN to produce frame-level LID-specific features along with their LRF values, 2) a novel LRF-based representation of the speech utterance (as a sequence of segment-level embeddings) for noise-robust LID, 3) a novel segment-level matching kernel (SLMK) based SVM for classifying the sequence of segment-level embeddings into language classes, and 4) extensive experimentation with the proposed approach on clean and noisy speech and comparison with state-of-the-art approaches.

The remainder of this paper is as follows. In Section 2, we describe our approach for LID. In Section 3, a description of the database used is given. In Section 4, details of various experiments and corresponding results are given followed by conclusions in Section 5.

2. PROPOSED FRAMEWORK

The proposed approach for LID is shown in Fig. 1. The overall system consists of a bidirectional long short-term memory (BLSTM) based DNN with self-attention to extract frame-level LID-specific features with corresponding attention values. In this work, we denote these attention values as language relevance factor (LRF) values as they represent the relevance of the feature vectors in deciding the language label. We propose two approaches for LID. In the first approach, the weighted sum of these LID-specific features is computed to get a fixed-length representation of speech, which is then applied as input to a classification layer. We call this end-to-end LID system as LRF-Net in the rest of this paper. In the second approach, we propose to divide the sequence of LID-specific feature vectors into a predetermined number of segments and then select a subset of feature vectors with significant LID-specific contents in each segment. In each segment, the mean of the selected feature vectors is computed followed by ℓ_1-normalization of the mean vector to get a segment-level embedding. The given utterance is then represented as the sequence of these segment-level embeddings. This utterance-level representation is then applied to the proposed SLMK-based SVM classifier which utilizes the locally available LID-specific contents in the representation.

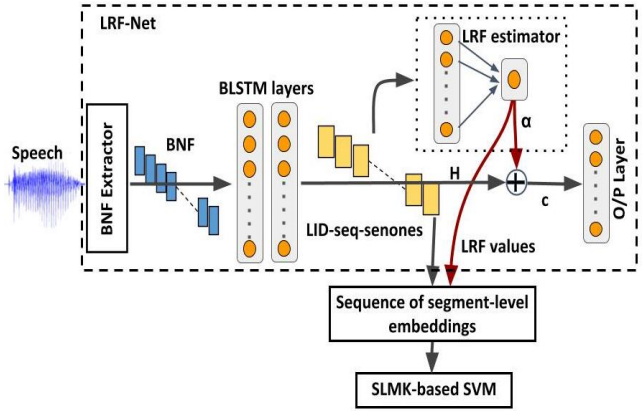

Fig. 1: Block diagram of the LRF-based framework for LID.

2.1. End-to-end LRF-Net

In our first approach, we use an end-to-end DNN called LRF-Net. The proposed LRF-Net is a BLSTM-based DNN with the self-attention mechanism as shown in Fig. 1. It contains a pretrained bottleneck feature (BNF) extractor [18] to convert the input speech into a sequence of 80-dimensional BNF vectors. This BNF extractor network was originally trained with 3096 phone states (from 17 languages) as targets and consists of a cascade of two bottleneck networks. Each of the extracted BNFs cover a total context of 31 frames (325 ms) of input speech [18].

The architecture of the proposed LRF-Net is similar to the network used in [17]. It contains 2 BLSTM layers with 256 and 64 nodes respectively in first and second layers to process the input BNF sequence. These layers process the sequence of input BNFs by dividing them into fixed-length overlapping chunks of 35 BNF vectors (covering 665 ms of speech in each chunk) to provide 128-dimensional LID-seq-senones [17]. These LID-seq-senones are nothing but the activations obtained at the output of the second BLSTM layer for each chunk of BNFs. Each LID-seq-senone is a compact representation of the LID-specific contents in the given

chunk of speech. The sequence of LID-seq-senones is then processed by an LRF (attention) estimator containing a dense layer followed by a layer with a single unit. Softmax operation is applied to the output of this single unit to compute the attention weight for each frame. Unlike in [17], where LID-seq-senones are treated independently, LRF-Net computes the weighted average of these LID-seq-senones to get a fixed-length utterance-level embedding (represented as **c** in Fig.1) of the speech as explained below.

A sequence of LID-seq-senones, $\mathbf{H} = (\mathbf{h}_1, .., \mathbf{h}_t, .., \mathbf{h}_T)$, where $\mathbf{h}_t \in \mathbb{R}^{128}$, and T is the length of a sequence, is obtained by passing the sequence of BNFs through the first 2 BLSTM layers of LRF-Net. Since this network will be trained for LID, the layers in the LRF-Net will learn to discriminate between the languages. Hence, each LID-seq-senone in **H** will contain more language-discriminative information than the input BNF. So, we call them as LID-specific features. Using **H**, an intermediate representation at the output of LRF estimator is computed as:

$$\gamma_t = \tanh(\tanh(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a)\mathbf{W}_\gamma + b_\gamma). \quad (1)$$

Here, $\mathbf{W}_a$, $\mathbf{b}_a$, $\mathbf{W}_\gamma$ and b_γ are parameters of the LRF estimator block which are to be learned along with other parameters of the LRF-Net. Using $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_t, ...\gamma_T)$, the LRF (attention) vector $\boldsymbol{\alpha} \in \mathbb{R}^T$, is then computed as:

$$\boldsymbol{\alpha} = \text{softmax}(\boldsymbol{\gamma}). \quad (2)$$

Using **H** and $\boldsymbol{\alpha}$, a fixed-length representation of the speech utterance is computed as:

$$\mathbf{c} = \mathbf{H}\boldsymbol{\alpha}. \quad (3)$$

This weighted average of LID-seq-senones (**c**, where $\mathbf{c} \in \mathbb{R}^{128}$) is then applied to the final dense output layer for classification. The network is trained for end-to-end LID using categorical cross entropy loss function. Note that, the obtained compact representation (**c**) considers the relevance of individual LID-seq-senones which might vary significantly due to the presence of background noise in a real-world speech sample. Hence, unlike i-vector and x-vector, utterance-level embedding **c** is noise-robust to some extent.

However, the obtained compact representation (**c**) does not preserve the locally available LID-specific information in the sequence of LID-seq-senones since it is computed as a weighted average of LID-seq-senones. Hence, we propose a new representation for the utterance that is computed using the LID-seq-senones with significant LRF values and preserves the locally available LID-specific information in them to some extent.

2.2. Proposed locality preserving LRF-based representation for the utterance

The Algorithm 1 explains the procedure of obtaining the proposed locality preserving representation of the utterance. Given LID-specific features **H** and LRF vector $\boldsymbol{\alpha}$ of an utterance, we divide them into a predetermined L number of segments of approximately equal length. In each segment, we select a predetermined k number of LID-seq-senones that correspond to the top-k LRF values. At each segment, the mean of the selected LID-seq-senones is computed followed by ℓ_1-normalization of the mean vector to obtain a segment-level embedding. Each segment-level embedding is a compact representation of the LID-specific contents available at that segment. By preserving the order of these segment-level embeddings, the overall temporal variations in the input sequence can be preserved. The utterance-level representation, $\widehat{\boldsymbol{\mathcal{H}}}$, is then obtained as a sequence of segment-level embeddings.

Algorithm 1 Extracting the LRF-based representation for the utterance ($\widehat{\boldsymbol{\mathcal{H}}}$) as a sequence of segment-level embeddings

Inputs:

(i) Sequence of LID-seq-senones, $\mathbf{H} = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_t, \ldots, \mathbf{h}_T)$ where, $\mathbf{h}_t \in \mathbb{R}^{128}$.

(ii) Corresponding LRF (attention) values, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_t, \ldots, \alpha_T)$ where, $\alpha_t \in \mathbb{R}$.

(iii) L: total number of segments ($j = 1, 2, ..., L$).

Procedure:

Divide **H** and $\boldsymbol{\alpha}$ into L segments.

Let $\{\mathbf{h}_i^j\}_{i=1}^{T_j}$ and $\{\alpha_i^j\}_{i=1}^{T_j}$ be the set of LID-seq-senones and corresponding set of LRF values, where, T_j indicates the number of LID-seq-senones and corresponding LRF values in j^{th} segment.

for segment number $j = 1$ **to** L **do**

Select k LID-seq-senones from $\{\mathbf{h}_i^j\}_{i=1}^{T_j}$ having maximum LRFs using top-k approach.

Compute the mean of selected LID-seq-senones and ℓ_1-normalize it to get segment-level embedding of the j^{th} segment, $\widehat{\mathbf{h}}_j$.

end for

Obtain utterance-level representation as sequence of segment-level embeddings: $\widehat{\boldsymbol{\mathcal{H}}} = (\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_2, .., \widehat{\mathbf{h}}_j, .., \widehat{\mathbf{h}}_L)$.

Outputs:

(i) Utterance-level representation of the speech sample $\widehat{\boldsymbol{\mathcal{H}}}$.

While it is possible to feed this utterance-level representation ($\widehat{\boldsymbol{\mathcal{H}}}$) to a simple classifier like Gaussian linear classifier [9, 10], we propose to use a classifier that utilizes the locally available LID-specific information in the representation. Recently, different types of sequence-kernel based SVM classifiers have shown to be effective in speech emotion recognition, speaker identification [15], [16] and LID [17]. These kernels find the similarity between two arbitrary length sequences by dividing them into increasingly finer segments and matching the corresponding segments at every level of the pyramid. These kernels use a class-independent Gaussian mixture model (CIGMM) based approach for obtaining a bag-

Algorithm 2 Segment-level matching kernel (SLMK) for sequence of segment-level embeddings $K_{\text{SLMK}}(\widehat{\boldsymbol{\mathcal{H}}}_m, \widehat{\boldsymbol{\mathcal{H}}}_n)$

Inputs:

(i) Utterance-level representations,
$\widehat{\boldsymbol{\mathcal{H}}}_m = (\widehat{\mathbf{h}}_{m1}, \widehat{\mathbf{h}}_{m2}, .., \widehat{\mathbf{h}}_{mj}, .., \widehat{\mathbf{h}}_{mL})$
$\widehat{\boldsymbol{\mathcal{H}}}_n = (\widehat{\mathbf{h}}_{n1}, \widehat{\mathbf{h}}_{n2}, .., \widehat{\mathbf{h}}_{nj}, .., \widehat{\mathbf{h}}_{nL})$

(ii) L: total number of segments.

Procedure:

for segment number $j = 1$ **to** L **do**

Compute matching score between embeddings of j^{th} segment $\widehat{\mathbf{h}}_{mj} = [\widehat{h}_{mj1}, \widehat{h}_{mj2}, .., \widehat{h}_{mjd}, .., \widehat{h}_{mjD}]$ and $\widehat{\mathbf{h}}_{nj} = [\widehat{h}_{nj1}, \widehat{h}_{nj2}, .., \widehat{h}_{njd}, .., \widehat{h}_{njD}]$ as:

$$S_j = \sum_{d=1}^{D} min(\widehat{h}_{mjd}, \widehat{h}_{njd}) \tag{4}$$

end for

Compute SLMK score between $\widehat{\boldsymbol{\mathcal{H}}}_m$ and $\widehat{\boldsymbol{\mathcal{H}}}_n$ as:

$$K_{\text{SLMK}}(\widehat{\boldsymbol{\mathcal{H}}}_m, \widehat{\boldsymbol{\mathcal{H}}}_n) = \sum_{j=1}^{L} S_j \tag{5}$$

Outputs:

(i) $K_{\text{SLMK}}(\widehat{\boldsymbol{\mathcal{H}}}_m, \widehat{\boldsymbol{\mathcal{H}}}_n)$.

of-code-words representation for each segment. Motivated by the ability of sequence-kernels to utilize the local-level class-specific contents in the speech, we propose a segment-level matching kernel (SLMK) based SVM classifier to classify the sequence of segment-level embeddings. This SLMK is very simple compared to the sequence-kernels in [15], [16] and [17], as it has to simply compute the similarity between two fixed-length sequences of segment-level embeddings while preserving the order of segments. The proposed approach for computing the similarity between two utterances, $\widehat{\boldsymbol{\mathcal{H}}}_m$ and $\widehat{\boldsymbol{\mathcal{H}}}_n$, using SLMK is given in Algorithm 2. As each segment-level embedding is a pseudo-probabilistic representation (due to ℓ_1-normalization), a histogram intersection matching (Eq. 4) is used to match corresponding segments from two sequences [19]. The final matching score (SLMK) is obtained as the sum of the matching scores at the segment-level as given in Eq. 5.

Note that, the proposed SLMK is a valid kernel because of the following reason. The similarity score computed in SLMK is based on histogram intersection matching kernel [19] which is a valid positive semi-definite kernel and the sum of valid positive semi-definite kernels is also a valid positive semi-definite kernel.

3. DATABASE

The LID dataset used in this study contains a set of closely related 9 Indian languages, provided by IIIT-Hyderabad [12, 13]. The details about the number of hours of speech data, the number of male and female speakers in both train and test sets is given in Table 1. This corpus contains read speech samples recorded in a controlled environment with a sampling rate of 16 kHz. We have divided larger speech files into smaller files such that all speech samples used in this experiment have a duration between 2 to 4 seconds. The training dataset contains 50465 files and testing dataset contains 16025 files. We have downsampled all speech files to 8 kHz in our experiments. This dataset is available upon request, for non-commercial and academic research purpose.

Many languages in this corpus are closely related. Majority of the phonemes are common among these languages [12]. For example, south Indian languages like Kannada, Malayalam and Telugu belong to the Dravidian language family and have many similar words. Similarly, Assamese, Bengali, Gujarati, Hindi and Punjabi belong to the Indo-Aryan language family. Since these languages are closely related, the correct identification of a language is very challenging.

Table 1: Details about the Indian languages used in the study along with duration (**Hours**), number of male (**#M**) and female (**#F**) speakers.

Language	**Train**			**Test**		
	Hours	**#M**	**#F**	**Hours**	**#M**	**#F**
Assamese	12.40	22	11	1.94	3	3
Bengali	9.91	24	35	1.53	15	15
Gujarati	9.71	115	75	2.18	37	36
Hindi	10.96	41	28	3.23	16	19
Kannada	10.08	21	16	0.99	10	4
Malayalam	10.08	7	6	3.07	9	7
Manipuri	5.31	5	6	2.50	3	3
Odia	9.81	31	31	2.45	9	9
Telugu	10.43	21	21	3.15	4	4

To simulate the real-world speech samples containing various types of indoor and outdoor noises, we have added the audio samples from DCASE-2017 scene classification development dataset[1] [20] to the clean speech samples. We have used the audio samples from 4 scenes (out of total 15), namely, Lakeside beach, Bus, Car, and City center as background noise types. We divided these 4 scenes into 2 sets. In set 1 (Lakeside beach and Bus), only 70% of samples from each class are used for corrupting the clean speech in training dataset. Remaining 30% of samples are used for corrupting the test dataset. This represents the test dataset under **seen** background conditions as the background noise conditions in these samples are already shown to the system during the training. The samples from set 2 (Car and City center) scene classes are added only to testing samples (resembling **unseen** types of background noise). These audio examples were originally recorded at 44.1 kHz sampling rate with a

[1] http://www.cs.tut.fi/sgn/arg/dcase2017/challenge/index

binaural microphone. In our experiments, these samples have been converted to monaural followed by low-pass filtering to retain frequency components only upto 4 kHz and then down-sampled to 8 kHz. Due to the addition of these acoustic scene samples to the clean speech, the resulting SNR of the speech varies between -0.5db to 19.22db with a mean of 7.34db.

4. EXPERIMENTS AND RESULTS

The performance of systems in this paper are reported in terms of two metrics: accuracy and the C_{avg} (given in NIST Language Recognition Evaluation 2015 [21]). Lower values of C_{avg} indicates better performance. We used Pytorch [22] and LIBSVM [23] tools for implementing DNNs and SVM classifiers respectively in this work.

4.1. Experiments on clean speech

In this section, we evaluate the performance of the proposed method for clean speech and compare with state-of-the-art approaches.

4.1.1. Baseline systems

The first baseline system is a x-vector based LID system [9]. This system consists of 5 feed-forward layers at the front-end to process the features at the frame-level followed by a statistical pooling layer to compute the mean and standard deviation. These statistics are then concatenated and further processed by a segment-level layer to get 512-dimensional x-vector. These x-vectors are then classified using a Gaussian back-end [9]. The system is trained using all clean speech samples in the training set. The result obtained is presented in the 1^{st} row of Table 2.

We have also implemented the Gaussian mixture model based segment-level pyramid match kernel (GSPMK) based LID system [17] that preserves the order of LID-seq-senones during the classification. To compute GSPMK, each speech utterance represented as a sequence of LID-seq-senones is repeatedly divided to form a pyramid of increasingly finer segments. Then GSPMK between a pair of varying length sequence of LID-seq-senones is computed by matching the corresponding segments at every level of the pyramid. The final GSPMK value between a pair of sequence of LID-seq-senones is computed as a weighted sum of the number of new matches found at different levels of the pyramid of segments [17]. The result obtained for the GSPMK based LID system is given in 2^{nd} row of Table 2.

4.1.2. LID using end-to-end LRF-Net

The LRF-Net is basically a self-attention based LID system as explained in section 2.1. The LRF estimator block in the LRF-Net has a dense layer with 100 nodes followed by a layer with a single unit to produce LRF (attention) weights. The final classification layer has 9 nodes to represent the languages. Categorical cross-entropy loss is used to train the network. The weights of the BNF extractor at the front-end are kept unchanged during the training. Performance of LRF-Net is given in 3^{rd} row of Table 2.

Table 2: Performance of baseline systems and LRF-Net. The best performance is marked in bold.

LID system	$C_{avg} \times 100$	Accuracy(%)
x-vector	2.74	94.29
GSPMK+SVM	2.69	94.65
LRF-Net	**2.58**	**95.20**

It is seen that all 3 systems have performed almost equally on clean speech. GSPMK with SVM (GSPMK+SVM) has performed slightly better compared to x-vector based system by utilizing the local-level LID-specific contents in the LID-seq-senones. The end-to-end LRF-Net has provided the best performance among the baseline systems. This indicates that even in clean speech, the utterance-level representation computed as a weighted average of the LID-seq-senones carries slightly more language-discriminative contents than the traditional approaches.

4.1.3. LID using proposed LRF-based representation with SLMK-based SVM

Here, we evaluate the effectiveness of the proposed utterance-level representation with SLMK-based SVM classifier. The LID-specific features and corresponding LRF values are obtained using the pretrained LRF-Net. We experimented by varying the number of segments (L) used to represent the utterance. Results obtained by varying the number of LID-seq-senones selected in each segment (k) are given in Table 3. Since some speech samples have as low as 192 frames of speech (after voice activity detection), we limited the maximum number of segments to 20. Also, we have not reported the results whenever the total number of frames to be considered ($L \times k$) are greater than 192 as few test samples will not satisfy this condition.

From Table 3 it is observed that, the performance of the proposed LRF-based representation with SLMK-based SVM is very sensitive to the number of segments and number of LID-seq-senones selected in each segment. In general, increase in the number of segments has led to better performance. The system has performed slightly better than the LRF-Net in one case (with L=16 and k=10). Last column in Table 3 shows the results when all LID-seq-senones in a segment are considered irrespective of their LRF values. Both LRF-Net and LRF-based representation with SLMK-based SVM (in 3 cases) have performed slightly better than this system by considering the relevance of individual LID-seq-senones.

Table 3: $C_{avg} \times 100$ obtained by varying number of segments (L) and number of LID-seq-senones in each segment (k). Last column indicates the performance obtained when all LID-seq-senones in a segment are used irrespective of their LRF values.

L	Number of frames in each segment (k)									
	1	2	5	10	15	25	50	100	150	All
1	48.03	47.12	45.10	38.41	32.42	28.23	20.04	9.45	4.53	3.22
2	44.56	43.16	35.66	27.38	22.23	19.34	8.60	-	-	3.13
4	34.90	31.02	25.39	16.40	10.23	7.54	-	-	-	3.10
8	25.10	21.14	14.51	5.75	2.63	-	-	-	-	2.92
16	11.31	9.35	4.10	**2.56**	-	-	-	-	-	2.98
20	9.12	7.30	2.70	-	-	-	-	-	-	2.96

From the results in Table 2 and 3, it can be seen that the proposed LRF-based representation method has provided only a slight improvement in performance compared to the baseline systems and LRF-Net in the case of clean speech. It indicates that, the proposed selection of feature vectors based on their LRF values is not much effective in clean speech case.

However, in the case of noisy speech, some feature vectors might have been significantly effected by noise. Hence, selection of the feature vectors based on their relevance to LID might be beneficial in this case. This motivates us to evaluate the performance of the proposed method on speech samples with different types of real-world background noise.

Table 4: $C_{avg} \times 100$ for different types of background noise.

LID system	noise type in test samples			
	seen noise		unseen noise	
	Beach	Bus	City	Car
x-vector	17.35	15.22	18.45	19.21
GSPMK+SVM	17.02	14.80	17.96	18.88
LRF-rep+ SLMK-SVM-All	17.23	15.22	18.32	19.04
LRF-Net	14.32	12.08	15.57	16.50
LRF-rep+ SLMK-SVM	**13.40**	**11.36**	**14.52**	**15.64**

4.2. Experiments on noisy speech

Here, we use the speech samples with different types of real-world background noise obtained by adding DCASE-2017 scene samples to the clean speech. We used a balanced training dataset containing 50% of speech samples with Beach and other 50% of samples with Bus type of noise. The results obtained for the proposed LRF-based utterance representation with SLMK-based SVM system (LRF-rep+SLMK-SVM) tested on speech with Beach and Bus types of noise (with L=16 and k=10) are shown in 5^{th} row of Table 4. The results obtained on speech with unseen types of noise from set 2 (City center and Car) are also shown on the right side of the Table 4. The performance of baseline systems are given in the first 2 rows of Table 4. 3^{rd} row in Table 4 shows the results obtained when all frames in the segment are considered irrespective of their LRF values (LRF-rep+SLMK-SVM-All). The performance of end-to-end LRF-Net is given in the 4^{th} row. Since the training dataset used in this case contains more complexity than the dataset with clean speech samples, the number of nodes in all hidden layers of LRF-Net has been doubled.

It is seen that both LRF-Net and proposed LRF-based representation with SVM systems (LRF-rep+SLMK-SVM) have performed significantly better than all other systems by considering the relevance of LID-seq-senones. Among these two, LRF-rep+SLMK-SVM has performed better than LRF-Net by utilizing the local-level LID-specific contents in the speech. In general, all these systems are sensitive to unseen types of background noise. Since both baseline systems and LRF-rep+SLMK-SVM-All do not have any explicit mechanism to handle the noise, they have performed poorly compared to both LRF-Net and LRF-rep+SLMK-SVM. The relevance based selection of LID-seq-senones and utilization of locally available LID-specific information have enabled the LRF-rep+SLMK-SVM system to perform comparatively better even in unseen background noise conditions.

5. CONCLUSIONS

We proposed a novel method of representing speech utterance using only frames with significant LID-specific contents. The significance of individual frame-level feature is decided based on its language relevance factor obtained using a self-attention based DNN. This representation preserves the locally available information in the input to some extent. Results obtained show that the proposed representation of the utterance used with the SLMK-based SVM classifier performs similarly to the state-of-the-art approaches on clean speech and it performs better than state-of-the-art systems in noisy speech.

In the future work, we will explore different methods to improve the robustness of the LRF-based LID system to unseen background noise conditions.

6. REFERENCES

[1] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[2] Pavel Matejka, Le Zhang, Tim Ng, Harish Sri Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang, "Neural network bottleneck features for language identification," in *Proceedings of Odyssey*, 2014, vol. 2014, pp. 299–304.

[3] Radek Fer, Pavel Matějka, František Grézl, Oldřich Plchot, Karel Veselỳ, and Jan Honza Černockỳ, "Multilingually trained bottleneck features in spoken language recognition," *Computer Speech & Language*, vol. 46, pp. 252–267, 2017.

[4] Mitchell McLaren, Luciana Ferrer, and Aaron Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5575–5579.

[5] Najim Dehak, Pedro Torres-Carrasquillo, Douglas Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proceedings of INTERSPEECH*, 2011, pp. 857–860.

[6] Bing Jiang, Yan Song, Si Wei, Jun-Hua Liu, Ian Vince McLoughlin, and Li-Rong Dai, "Deep bottleneck features for spoken language identification," *PloS one*, vol. 9, no. 7, pp. e100795, 2014.

[7] 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.

[8] Jan Pešán, Lukáš Burget, and Jan Černockỳ, "Sequence summarizing neural networks for spoken language recognition," *Proceedings of Interspeech 2016*, pp. 3285–3288, 2016.

[9] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "Spoken language recognition using x-vectors," in *Proceedings of Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 105–111.

[10] Alicia Lozano-Diez, Oldřich Plchot, Pavel Matejka, and Joaquin Gonzalez-Rodriguez, "DNN based embeddings for language recognition," in *Proceedings of 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5184–5188.

[11] B. Padi, A. Mohan, and S. Ganapathy, "End-to-end language recognition using attention based hierarchical gated recurrent unit models," in *Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5966–5970.

[12] KV Mounika, Sivanand Achanta, HR Lakshmi, Suryakanth V Gangashetty, and Anil Kumar Vuppala, "An investigation of deep neural network architectures for language recognition in indian languages.," in *Proceedings of INTERSPEECH*, 2016, pp. 2930–2933.

[13] Ravi Kumar Vuddagiri, Hari Krishna Vydana, and Anil Kumar Vuppala, "Curriculum learning based approach for noise robust language identification using dnn with attention," *Expert Systems with Applications*, vol. 110, pp. 290–297, 2018.

[14] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey, "Self-attentive speaker embeddings for text-independent speaker verification.," in *Proceedings of INTERSPEECH*, 2018, pp. 3573–3577.

[15] Shikha Gupta, Aroor Dinesh Dileep, and Thenkanidiyoor Veena, "Segment-level pyramid match kernels for the classification of varying length patterns of speech using SVMs," in *Proceedings of 2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 2030–2034.

[16] Shikha Gupta, Thenkanidiyoor Veena, and Aroor Dinesh Dileep, "Segment-level probabilistic sequence kernel based support vector machines for classification of varying length patterns of speech," in *Neural Information Processing*, Cham, 2016, pp. 321–328, Springer International Publishing.

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

[18] 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 *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 283–287.

[19] Annalisa Barla, Francesca Odone, and Alessandro Verri, "Histogram intersection kernel for image classification," in *Proceedings 2003 international conference on image processing (Cat. No. 03CH37429)*. IEEE, 2003, vol. 3, pp. III–513.

[20] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen, "Dcase 2017 challenge setup: Tasks, datasets and baseline system," 2017.

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

[22] 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 *NIPS-W*, 2017.

[23] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, April 2011, Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.