INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

A new time-frequency attention mechanism for TDNN and CNN-LSTM-TDNN, with application to language identification

Xiaoxiao Miao[1,2,3], *Ian McLoughlin*[1], *Yonghong Yan*[2,3,4]

[1]School of Computing, The University of Kent, Medway, UK
[2]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Xinjiang Key Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences
xm39@kent.ac.uk

Abstract

In this paper, we aim to improve traditional DNN x-vector language identification (LID) performance by employing Convolutional and Long Short Term Memory-Recurrent (CLSTM) Neural Networks, as they can strengthen feature extraction and capture longer temporal dependencies. We also propose a two-dimensional attention mechanism. Compared with conventional one-dimensional time attention, our method introduces a frequency attention mechanism to give different weights to different frequency bands to generate weighted means and standard deviations. This mechanism can direct attention to either time or frequency information, and can be trained or fused singly or jointly. Experimental results show firstly that CLSTM can significantly outperform a traditional DNN x-vector implementation. Secondly, the proposed frequency attention method is more effective than time attention, particularly when the number of frequency bands matches the feature size. Furthermore, frequency-time score merging is the best, whereas frequency-time feature merge only shows improvements for small frequency dimension.

Index Terms: Language Identification, DNN x-vector, CLSTM, Time attention, Frequency attention

1. Introduction

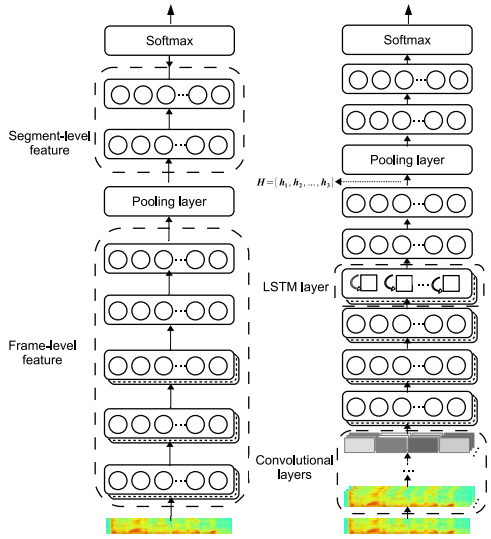
The development of Deep Neural Networks (DNN) [1] in recent years, has enabled DNN-based language identification (LID) to obtain increasingly good results. This usually involves DNNs replacing one or more of the components of a conventional Gaussian Mixture Model (GMM) i-vector-based system (i-vector extraction from GMM super vector and classifier training at the back-end) [2, 3]. In the feature domain, some researchers [4, 5] have used a phoneme dependent deep bottleneck feature (DBF) extractor, obtained from the lower layers of a deep bottleneck network (DBN) that has been well trained for an automatic speech recognition (ASR) task. In the model domain, for both LID and Speaker Recognition (SRE), novel total variability (TV) modelling methods have been proposed based on phonetic-aware DNNs [6, 7]. In these studies, instead of GMM posterior probability, DNN output posteriors are exploited to obtain sufficient statistics. Thus the DBF DNN i-vector [8, 9, 10] method was proposed. This combines a DBF for extracting robust features with the posteriors of the DNN for improved model capability, obtaining more and better phoneme information for the TV modelling, further enhancing LID performance. These advances clearly demonstrated the relevance of phonetic-aware ASR-trained DNNs for the LID task.

In addition, end-to-end LID systems based on DNNs have been recently proposed, which abandon the traditional LID system framework but combine the individual components instead. In 2014, Google researchers [11] incorporated feature extraction, feature transformation, and classification into a single NN model. Later researchers took advantage of different NN structures, including Time-Delay Neural Network (TDNN) [12], Long Short Term Memory-Recurrent Neural Network (LSTM-RNN) [13]. However, all of them use frame level features as input and predict frame level labels, requiring post-processing to obtain utterance labels, which may not be very effective.

Ideally, an end-to-end method should take frame level features as input and directly produce an utterance label. Obviously, the utterance representation is obtained by means of a pooling layer. For example, statistics layer/x-vector [14], spatial pyramid pooling (SPP) [15], a learnable dictionary coding layer [16, 17], which directly learn language category information from the underlying acoustic features. The most important merit of end-to-end systems is that they abandon the acoustic model; they have no need of elaborate phone labels, but still can achieve similar or better performance. In most prior work, these pooling layers assign equal weight to each frame-level feature. However, [18, 19, 20, 21] proposed using frame-level weights for LID/SRE that are learned by a structured attention mechanism, then incorporated into a weighted statistics pooling layer to get one utterance representation. Results are encouraging and show that an attention mechanism can be beneficial. This paper continues to explore end-to-end x-vector based LID. We first improve traditional systems by adopting an input CNN to strengthen feature extraction, then LSTM to model temporal dependencies to learn long-range discriminative features over the input sequence, yielding a CLSTM system, the similar system has shown good performance in speech recognition [22]. We then make two further contributions; (a) while current attention-based methods only consider information along the time axis [18, 19, 20, 21], we believe that frequency also contains discriminative regions, so we create and evaluate a novel frequency attention mechanism. (b) we then propose a two dimensional time-frequency attention mechanism where frame-level and frequency band-sensitive attentions can be trained or fused jointly.

The remainder of the paper is organized as follows: Section 2 describes the baseline systems and the proposed changes; Section 3 reports experimental results and then Section 4 concludes our work and discusses future issues.

http://dx.doi.org/10.21437/Interspeech.2019-1256



(a) *TDNN* (b) *CNN-LSTM-TDNN*

Figure 1: *Architecture of baseline and improved LID systems.*

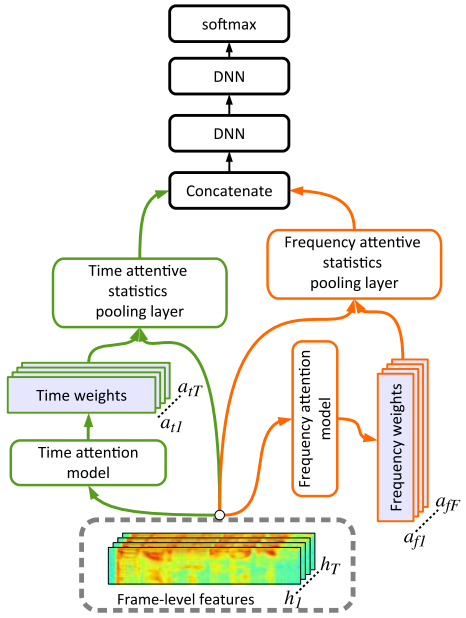


Figure 2: *Proposed time and frequency attention mechanisms.*

2. language identification systems

2.1. GMM/DNN/DBF DNN i-vector

The GMM/DNN/DBF DNN i-vector baselines that we adopt all need to compute sufficient statistics:

$$N_k(s) = \sum_{t=1}^{T_s} p(k \mid \mathbf{x}_{s,t}, \phi) \quad (1)$$

$$\mathbf{F}_k(s) = \sum_{t=1}^{T_s} p(k \mid \mathbf{x}_{s,t}, \phi)\mathbf{y}_{s,t} \quad (2)$$

$$\mathbf{S}_k(s) = \sum_{t=1}^{T_s} p(k \mid \mathbf{x}_{s,t}, \phi)\mathbf{y}_{s,t}\mathbf{y}_{s,t}^{\top} \quad (3)$$

In a conventional GMM Supervector approach, all frames of features in the training dataset are grouped together to estimate a universal background model (UBM). For GMM i-vector [3], ϕ represents the parameters of the GMM UBM, $p(k\ |)$ corresponds to k-th GMM occupancy probability, $x_{s,t}$ is same with $y_{s,t}$ that is the acoustic feature of the t-th frame of utterance s that has L frames. Compared with GMM i-vector, the only difference for DNN i-vector [10, 8] is that ϕ represents the parameters of the pre-trained ASR DNN, $p(k\ |)$ corresponds to k class posteriors from the ASR DNN. Compared with DNN i-vector, DBN i-vector [10, 8] sets $y_{s,t}$ to be the DBF vector from the t-th frame of utterances.

These sufficient statistics are all that are needed to train subspace T and extract the i-vector, followed by back-end classification using multi-class logistic regression training.

2.2. DNN x-vector

The baseline end-to-end LID x-vector system based on a Time-Delay Neural Network TDNN [14] structure is shown in Fig. 1(a). Frame-level features centering on the current frame with small extended context, are input to the first five layers. The statistical pooling layer accumulates all frame-level outputs, calculates the mean and standard deviation, and obtains a segment-level fixed-dimension representation. Segment-level statistics are then passed to two additional fully connected hidden layers and finally a softmax output layer.

2.3. CLSTM x-vector

The structure of the proposed CLSTM x-vector system is shown in Fig. 1(b). The initial CNN can extract local feature descriptors automatically from input frames with context. To some extent, the convolution layer of the CNN operates as a sliding window to act as an automatic local feature extractor. It learns a temporal ordered feature representation automatically under backward propagation directed by the loss function.

LSTMs have a powerful ability to handle long-term dependencies, which we believe can be discriminative for languages. We therefore add one LSTM layer between the TDNN layers. Several variants of LSTM have been proposed for RNNs. The baseline LSTM architecture we chose [23] is one that has shown good performance in the related ASR task.

2.4. Attention-based x-vector

2.4.1. Time Attention

It is often the case that frame-level features from some frames are more important for discriminating languages than others in a given utterance. Recent studies [18, 19, 20, 21] have applied attention mechanisms to SRE/LID for the purpose of frame selection, by automatically calculating the importance of each frame. We therefore use an attention model in conjunction with the CLSTM network to calculate a scalar score for each frame-level feature. In this way, utterance-level features extracted from a weighted mean vector focus on important frames to become more language discriminative. We call this time attention, and show it in the green coloured path (left) in Fig. 2.

Given input frame $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, , \ldots, \boldsymbol{x}_T\}$, where T is segment duration, the output of the hidden layer below the attention layer is $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T\}$. Let the dimension of each h be d_h so the size of $\boldsymbol{H}$ is $d_h \times T$. The time attention

Table 1: *Performance results for time and frequency attention.*

	3s			10s			30s		
System	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**
GMM i-vector	43.33	18.49	16.12	19.32	9.31	7.04	8.83	4.18	2.50
DNN i-vector	31.42	13.46	10.75	9.82	4.28	3.24	4.08	1.39	1.11
DBF DNN i-vector	23.73	9.35	7.73	8.06	3.17	4.94	2.46	1.11	0.78
DNN x-vector	25.90	10.31	9.03	11.17	3.56	3.38	5.79	1.75	1.71
DNN x-vector time	24.84	9.96	9.12	11.03	3.67	3.33	6.67	1.86	1.71
CLSTM	19.51	7.09	6.67	6.02	1.64	1.66	2.46	0.81	0.78
CLSTM time	19.46	7.14	6.76	5.38	1.68	1.76	2.69	0.70	0.83
CLSTM fre2D	19.69	7.32	6.90	5.05	1.64	1.71	1.90	0.49	0.64
CLSTM fre8D	19.51	6.60	6.58	5.14	1.48	1.66	1.90	0.53	0.69
CLSTM fre16D	19.69	6.84	6.67	5.05	1.39	1.62	2.13	0.53	0.60
CLSTM fre23D	19.14	6.47	6.48	4.91	1.40	1.48	1.71	0.42	0.55
CLSTM fre32D	19.05	6.29	6.16	4.77	1.33	1.43	1.85	0.43	0.55

mechanism takes the whole hidden representation $\boldsymbol{H}$ as input, and outputs an annotation matrix $\boldsymbol{T}_A$ as follows:

$$\mathbf{T}_A = softmax(g(\boldsymbol{H}^T \boldsymbol{W}_1^t)\boldsymbol{W}_2^t) \tag{4}$$

where $\boldsymbol{W}_1^t$ is a matrix of size $d_h \times d_a$, with d_a being the dimension of the attention vector. $\boldsymbol{W}_2^t$ is a vector of size $d_a \times 1$; $g()$ is some activation function, ReLU in this paper. $softmax()$ is performed column-wise and each column vector of $\boldsymbol{T}_A$ is an annotation vector that represents the weights for different h_t. Finally the weighted means $\boldsymbol{E}_t$ is obtained by

$$\mathbf{E}_t = \boldsymbol{H}\boldsymbol{T}_A \tag{5}$$

2.4.2. Frequency Attention

The traditional time attention mechanism [18, 19, 20, 21] relates positions on the time axis to temporal dependencies. However, we believe language features can be discriminative in both time and frequency, since different languages display varying correlations of different frequencies with time. Therefore, we propose that attending to frequency may be beneficial to the language model. Motivated by this, we propose a modified attention block, illustrated in the orange right hand path in Fig. 2,

$$\mathbf{F}_A = softmax(g(\boldsymbol{H}^T \boldsymbol{W}_1^f)\boldsymbol{W}_2^f) \tag{6}$$

where $\boldsymbol{W}_1^f$ is a matrix of size $d_h \times d_a$; $\boldsymbol{W}_2^f$ is a vector of size $d_a \times d_f$, where d_f can be 2, 4, 8, 16, 23, 32 in the following experiments; $g()$ is again a ReLU activation function. $softmax()$ is performed rank-wise in this case and each rank of $\boldsymbol{F}_A$ is an annotation vector that represents the weights for different frequency bands $\boldsymbol{h}_b$. For example, if the number of hidden nodes is 1500 and d_f is 2, there are two frequency bands $[1 : 750]$, $[751 : 1500]$. Suppose $\boldsymbol{F}_A$=[a_{f1}, a_{f2}] computed by (6) , then $a_{f1} \times h \in [1 : 750]$, and $a_{f2} \times h \in [751 : 1500]$. Mean and variance are then computed from the frequency weighted h.

2.4.3. Combination methods

- Feature-level combination

  Two types of attention for capturing temporal and spectral dependencies are shown in Fig. 2: the green path (left) attends to the time axis while the orange one (right) attends to the frequency axis. The final outputs of attention are concatenated and fed into fully connect layers.

- Score-level combination

  Another approach is a score level fusion that combines scores calculated by time attention $\boldsymbol{s}^t(u)$ and frequency attention $\boldsymbol{s}^f(u)$. For example, the fused score $\boldsymbol{s}(u)$ of utterance u can be calculated as follows:

  $$\boldsymbol{s}(u) = (1-\alpha)\boldsymbol{s}^f(u) + \alpha\boldsymbol{s}^t(u)$$

  Where $\boldsymbol{s}()$ is the overall scoring function.

3. Evaluation

3.1. Database and experimental setup

3.1.1. Database

The ASR DNN is trained on roughly 1000 hours of clean English telephone speech from Fisher. For the LID task, we conducted experiments using NIST LRE07 which is a closed-set language detection task spanning 14 languages. The experiments used the LID training corpus including Callfriend datasets, LRE03, LRE05, SRE08 datasets, and development data for LRE07. The experimental LID test corpus was the NIST LRE07 test dataset separated into 30s, 10s and 3s conditions. Each condition has 2158 utterances. We also used training data augmentation [14] to increases the amount and diversity of the existing training data, including additive noises (MUSAN dataset) and reverberation (RIR dataset).

3.1.2. Experimental setup

For the GMM i-vector system, raw audio is converted to 7-1-3-7 based 56 dimensional SDC features, and a frame-level energy-based VAD selects features corresponding to speech frames. A 2048 component full covariance GMM UBM is trained, along with a 600 dimensional i-vector extractor, followed by length normalization and multi-class logistic regression.

A nine-layer ASR DNN is trained with cross entropy, from a 40×11 input layer (40-dimensional PLP features concatenation over a context of the current frame with the preceding and following 5 frames). Input is followed by linear discriminant analysis (LDA). The hidden layers have 3000 nodes followed by Pnorm nonlinear activation (with 300 nodes) and normalization, except that the bottleneck layer (the fourth hidden layer) has 390 nodes; the output of Pnorm is 39 dimensional and the

Table 2: *Performance results for the feature-level combination of time and frequency attention.*

	3s			10s			30s		
System	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**
CLSTM time	19.46	7.14	6.76	5.38	1.68	1.76	2.69	0.70	0.83
CLSTM fre23D	19.14	6.47	6.48	4.91	1.40	1.48	1.71	0.42	0.55
CLSTM time fre2D	18.95	6.57	6.48	4.96	1.47	1.62	1.71	0.46	0.55
CLSTM time fre8D	19.42	7.01	6.76	5.42	1.52	1.62	2.09	0.59	0.69
CLSTM time fre16D	18.95	6.39	6.62	5.14	1.44	1.57	1.71	0.56	0.69
CLSTM time fre23D	19.51	6.43	6.39	5.00	1.66	1.66	2.13	0.50	0.60
CLSTM time fre32D	20.25	7.13	6.34	5.14	1.51	1.57	1.90	0.53	0.64

Table 3: *Performance results for the score-level combination of time and frequency attention.*

	3s			10s			30s		
System	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**
CLSTM time	19.46	7.14	6.76	5.38	1.68	1.76	2.69	0.70	0.83
CLSTM fre23D	19.14	6.47	6.48	4.91	1.40	1.48	1.71	0.42	0.55
CLSTM time fre2D feature	18.95	6.57	6.48	4.96	1.47	1.62	1.71	0.46	0.55
CLSTM time fre2D score	18.16	7.46	6.20	4.54	1.59	1.48	1.90	0.54	0.64
CLSTM time fre8D score	17.89	7.36	6.02	4.96	1.43	1.34	1.95	0.57	0.64
CLSTM time fre16D score	18.12	7.19	6.34	4.91	1.54	1.43	1.76	0.62	0.64
CLSTM time fre23D score	17.42	7.28	6.16	4.45	1.59	1.43	1.85	0.50	0.60
CLSTM time fre32D score	18.58	7.24	5.97	4.49	1.49	1.39	2.09	0.45	0.55

BNFs are extracted from the subsequent normalization. The output layer has 5560 nodes.

For the x-vector system, the features are 23 dimensional MFCC with a frame-length of 25ms, mean-normalized over a sliding window of up to 3 seconds. An energy-based speech activity detector, identical to that used in the baseline systems, filters out non-speech frames. The DNN x-vector configuration follows [14], CLSTM x-vector has two convolutional layers with 3×3 filters followed by ReLU and batch normalization. The number of filters was 128 and 256 respectively, the cell-dim of LSTM is 1024, the recurrence of dimension is 256. For the time and frequency attention models, the number of hidden nodes was 64 and we used ReLU activation functions followed by batch normalization. All the experiments are carried out using Kaldi [24].

3.2. Experimental results

Table 1 lists results from various traditional LID methods, and the proposed CLSTM system, showing time attention and frequency attention of various dimensions separately. Firstly, we note that the CLSTM x-vector system clearly outperforms DNN x-vector methods. Then, incorporating time attention can improve both methods, corroborating existing published results for DNN x-vector [14]. Secondly, frequency attention is able to improve on time attention, although it is weakly sensitive to feature dimension. Best performance is obtained with 23 dimension features – which matches the size of the MFCC input features.

Table 2 shows results for the feature-level combination of time and frequency attention, and reveals that combining both can yield further advantage. In particular, 2 and 16 dimension feature merging performs well for the 3s condition, and 2 dimensions is best for other conditions. Score-level merging of time and frequency attention systems is presented in Table 3, and seen to outperform the feature-level merge of Table 2. In fact all dimension systems in Table 3 can outperform either feature or time attention alone, particularly for the most challenging 3s condition.

4. Conclusions

This paper first presented and evaluated a new end-to-end LID architecture named CLSTM, employing a CNN front-end for a deep neural network structure with LSTM for extracting time sequencing information. Performance exceeds that of traditional DNN x-vector architectures on the LRE2007 task. We then evaluated an attention mechanism in the time domain, in accordance with previously developed x-vector based attention mechanisms, noting system performance improvements on x-vector as well as the new architecture. Next we proposed a frequency domain attention mechanism which yielded a small performance improvement over a time attention mechanism on LRE2007, and then the novel approach of extending the attention mechanism to two dimensions, encompassing both time and frequency domain discriminative features. Evaluating both feature-level and score-level merger of the two dimensions, we noted that score level merger performed best. The most challenging 3s task yielded particularly good results, improving on the performance of state-of-the-art systems for all tested durations.

5. Acknowledgements

This work is partially supported by the National Key Research and Development Program (Nos. 2016YFB0801203, 2016YFB0801200) , the National Natural Science Foundation of China (Nos. 11590774, 11590770), the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No.2016A03007-1) Thanks to China Scholarship Council for funding to conduct this research at the University of Kent, Medway, UK.

6. References

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[2] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, 2006.

[3] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Twelfth annual conference of the international speech communication association*, 2011.

[4] B. Jiang, Y. Song, S. Wei, M.-G. Wang, I. McLoughlin, and L.-R. Dai, "Performance evaluation of deep bottleneck features for spoken language identification," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 143–147.

[5] Y. Song, X. Hong, B. Jiang, R. Cui, I. McLoughlin, and L.-R. Dai, "Deep bottleneck network based i-vector representation for language identification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[7] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[8] F. Richardson, D. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923*, 2015.

[9] B. Jiang, Y. Song, S. Wei, J.-H. Liu, I. V. McLoughlin, and L.-R. Dai, "Deep bottleneck features for spoken language identification," *PLoS one*, vol. 9, no. 7, p. e100795, 2014.

[10] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[11] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5337–5341.

[12] D. Garcia-Romero and A. McCree, "Stacked long-term TDNN for spoken language recognition." in *Prof. Interspeech*, 2016, pp. 3226–3230.

[13] J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, and P. J. Moreno, "Automatic language identification using long short-term memory recurrent neural networks," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[14] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," *submitted to Odyssey*, 2018.

[15] M. Jin, Y. Song, I. McLoughlin, and L.-R. Dai, "LID-senones and their statistics for language identification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 171–183, 2018.

[16] W. Cai, Z. Cai, W. Liu, X. Wang, and M. Li, "Insights into end-to-end learning scheme for language identification," *arXiv preprint arXiv:1804.00381*, 2018.

[17] W. Cai, Z. Cai, X. Zhang, X. Wang, and M. Li, "A novel learnable dictionary encoding layer for end-to-end language identification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5189–5193.

[18] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[19] W. Geng, W. Wang, Y. Zhao, X. Cai, B. Xu, C. Xinyuan *et al.*, "End-to-end language identification using attention-based recurrent neural networks," *Proc. Interspeech*, 2016.

[20] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[21] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Proc. Interspeech*, pp. 3573–3577, 2018.

[22] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[23] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *arXiv preprint arXiv:1402.1128*, 2014.

[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.