Circuits, Systems, and Signal Processing
https://doi.org/10.1007/s00034-019-01286-9

SHORT PAPER

A New Time–Frequency Attention Tensor Network for Language Identification

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

Received: 11 April 2019 / Revised: 8 October 2019 / Accepted: 10 October 2019

Abstract
In this paper, we aim to improve traditional DNN x-vector language identification performance by employing wide residual networks (WRN) as a powerful feature extractor which we combine with a novel frequency attention network. Compared with conventional time attention, our method learns discriminative weights for different frequency bands to generate weighted means and standard deviations for utterance-level classification. This mechanism enables the architecture to direct attention to important frequency bands rather than important time frames, as in traditional time attention methods. Furthermore, we then introduce a cross-layer frequency attention tensor network (CLF-ATN) which exploits information from different layers to recapture frame-level language characteristics that have been dropped by aggressive frequency pooling in lower layers. This effectively restores fine-grained discriminative language details. Finally, we explore the joint fusion of frame-level and frequency-band attention in a time–frequency attention network. Experimental results show that firstly, WRN can significantly outperform a traditional DNN x-vector implementation; secondly, the proposed frequency attention method is more effective than time attention; and thirdly, frequency–time score fusion can yield further improvement. Finally, extensive experiments on CLF-ATN demonstrate that it is able to improve discrimination by regaining dropped fine-grained frequency information, particularly for low-dimension frequency features.

Keywords Language identification · DNN x-vector · Time–frequency attention tensor network · Cross-layer frequency tensor attention network

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. Also thanks to the Data Science Research Group and the University of Kent School of Computing for supporting this research.

✉ Xiaoxiao Miao
xm39@kent.ac.uk; miaoxiaoxiao@hccl.ioa.ac.cn

Extended author information available on the last page of the article

Published online: 28 October 2019

1 Introduction

Language identification (LID) is part of a pre-processing procedure performed before automatic speech recognition in a multi-lingual context, or when doing language-specific post-processing. The main task of LID is to identify which language is being spoken using differentiated information extracted from the speech signals, and to do so with speed and accuracy. Much work has been done in this field in recent years, and most traditional solutions are based on i-vector utterance representations, most commonly Gaussian mixture model (GMM) i-vector-based systems [7,9]. In these systems, i-vectors are learned from GMM supervectors in an unsupervised way without any language labels, followed by classifier training.

Thanks to developments in deep neural networks (DNN) [15], DNN-based i-vector LID has led to remarkable improvements, still within the conventional i-vector-based framework. This firstly relies on effective acoustic modelling. In the front-end feature domain, Jiang et al. [16] and Song et al. [29] developed discriminative acoustic–phonetic features called deep bottleneck features (DBF). These are extracted using a deep bottleneck network (DBN) that has been well trained for an associated automatic speech recognition (ASR) task [28], demonstrating extremely good performance for LID. In the back-end model domain, several authors [17,19,20] proposed using the output posteriors from a pre-trained DNN for ASR to obtain phonetic-aware Baum–Welch statistics instead of GMM posterior probabilities. These together produce the successful DBF DNN i-vector [25,26] method, which combines a DBN for front-end frame-level feature extraction with the posteriors of the DNN for back-end utterance-level modelling, further enhancing LID performance. These advances clearly demonstrated that phonetic-aware ASR-trained DNNs are effective for LID tasks, which is consistent with the fact that different languages include different phoneme sets and combination statistics.

However, ASR-trained DNNs rely heavily on acoustic models, needing elaborate phone labels, and multi-stage training strategies. Thus, end-to-end LID systems based on DNNs have recently gained steady interest. These abandon the acoustic model and instead directly learn LID information, combining the individual components of i-vector-based systems into end-to-end schemes [10,12,21]. These systems take frame-level features as input and predict frame-level labels. Some post-processing is then required to obtain utterance-level labels from the frame-level posteriors. To be more effective and accurate, researchers have recently developed end-to-end methods that input frame-level features and directly produce utterance labels without additional post-processing, e.g. using a statistics layer in the x-vector system [27], spatial pyramid pooling (SPP) [18], or incorporating a learnable dictionary coding layer [4,5].

Other recent studies used attention mechanisms. A neural network armed with an attention mechanism is able to focus on useful information for certain tasks while ignoring or diminishing the importance of others. This attention vector can be regarded as a series of changeable connections that let the forward information and the backward information flow more effectively. It has previously produced significant improvement in image captioning [32], machine translation [8], speech recognition [3]. Attention mechanisms were then introduced into the LID and speaker recognition (SRE) fields [6,11,23,34] to obtain weighted frame-level feature vectors. These focus on important frames rather than assigning equal weights to each frame-level feature and are shown to perform well.

This paper starts with an end-to-end x-vector-based LID framework. We firstly aim to improve performance by adopting a wide residual network (WRN) [30,33] structure, as opposed to the commonly used narrow and very deep counterparts (e.g. the original ResNet [14] for image recognition). WRNs decrease depth and increase the width of the convolutional layers by adding more feature maps to each residual block, addressing the slow-speed training problems that occur in very deep residual networks. After characterizing that system, we make three further contributions:

(a) Current attention-based methods only consider information along the time axis [6,11,23,34]. Some studies [1,2,22] have demonstrated f0 range differences between languages, and we therefore believe that languages can be discriminated through their patterns in dominant frequency ranges. We therefore create and evaluate a novel frequency attention network (F-ATN) which dynamically adjusts the weight of different frequency bands to improve discrimination.

(b) Frequency pooling methods are important for convolutional networks. They compress the dimension or size of feature maps to decrease feature resolution. During this process, some language information is lost or is not well represented in low-resolution frequency features. To solve this problem, we propose optimizing F-ATN by introducing a cross-layer frequency attention tensor network (CLF-ATN) which allows fine-grained frequency information to bypass aggressive pooling operations.

(c) We investigate a combined use of frequency and time attention information to give more detailed language-discriminative information. We consider a two-dimensional time–frequency attention mechanism where frame-level and frequency-band-sensitive attentions can be fused jointly in the belief that the frequency- and time-discriminative adjustments are complementary.

2 Language Identification Systems

2.1 GMM/DNN/DBF DNN i-Vector

The GMM/DNN/DBF DNN i-vector baselines systems all rely on statistics collected over frame-level features in an utterance. Sufficient statistics will allow the back-end classifiers to discriminate between languages. Zero-, first- and second-order statistics from all tested systems and features can be described using three equations, where the component variables x, y and $p(, \phi)$ map to different information in each system (described below):

$$N_k(s) = \sum_{t=1}^{T_s} p(k \mid x_{s,t}, \phi) \tag{1}$$

$$F_k(s) = \sum_{t=1}^{T_s} p(k \mid x_{s,t}, \phi) y_{s,t} \tag{2}$$

$$S_k(s) = \sum_{t=1}^{T_s} p(k \mid x_{s,t}, \phi) y_{s,t} {y_{s,t}}^{\top} \tag{3}$$

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 [9], ϕ represents the parameters of the GMM UBM, $p(k \mid)$ corresponds to the k-th GMM occupancy probability, $x_{s,t}$ are the acoustic feature of the t-th frame of utterance s that has L frames (and are the same as $y_{s,t}$ in the GMM UBM). In the DNN i-vector system [25,26], ϕ now represents the parameters of the pre-trained ASR DNN, and $p(k \mid)$ corresponds to its k class posteriors. In the DBN i-vector system [25,26], $y_{s,t}$ becomes the DBF vector from the t-th frame of utterances. For each system a subspace is trained from the sufficient statistics to extract the i-vector which is used in the back-end classifier for multi-class logistic regression training.

2.2 DNN x-Vector

The baseline end-to-end LID x-vector system is based on a time-delay neural network (TDNN) [27] structure. Frame-level features centred on the current frame, along with a small extended context, are the input to the first five-layer block. A statistical pooling layer then 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 which finally feed into a softmax output layer.

2.3 WRN x-Vector

The structure of the proposed WRN x-vector system is shown in the bottom part of Fig. 1. Wide residual networks decrease depth and increase the width of residual networks, addressing the very slow-speed training problems that occur with very deep residual networks. Wide residual networks have been found, in general, to function well as powerful feature extractors. The baseline WRN architecture we chose [31] is one that has shown good performance in the related ASR task.

2.4 Attention-Based x-Vector

2.4.1 Time Attention Network

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 [6,11,23,34] have applied attention mechanisms to SRE/LID for the purpose of frame selection, by automatically calculating the importance of each frame. We therefore apply an attention model to a WRN network. This calculates a scalar score for each frame-level feature in an utterance. In this way, utterance-level features extracted from a weighted mean vector focus on important frames to improve language discrimination. We call

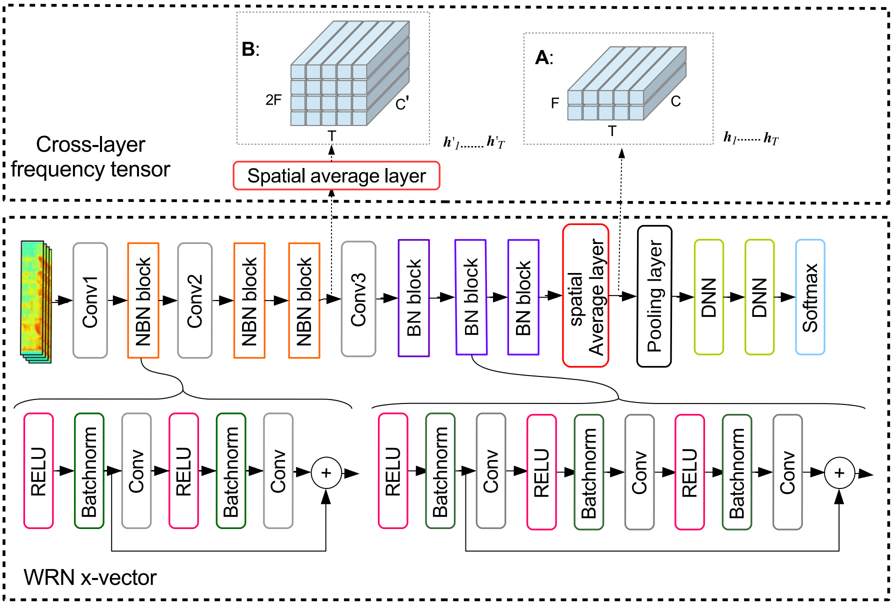


Fig. 1 Proposed WRN system for LID showing expanded detail of the structure of the repeated non-bottleneck (NBN) and bottleneck (BN) blocks below and the attention tensor above

this *time attention* and show it in the green-coloured path (left) in Fig. 2b which includes this as part of the time–frequency attention mechanism described later (i.e. at present, ignore the orange path to the right).

Given an input frame vector $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, , \ldots, \mathbf{x}_T\}$, T represents segment duration, and the output of the hidden layer before the attention layer is $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T\}$. The dimension of each $\mathbf{h}_t$ is d_h so the size of $\mathbf{H}$ is then $d_h \times T$. Time attention uses the whole of $\mathbf{H}$, the hidden representation, as input. The output is then $\mathbf{T}_A$, called an annotation matrix:

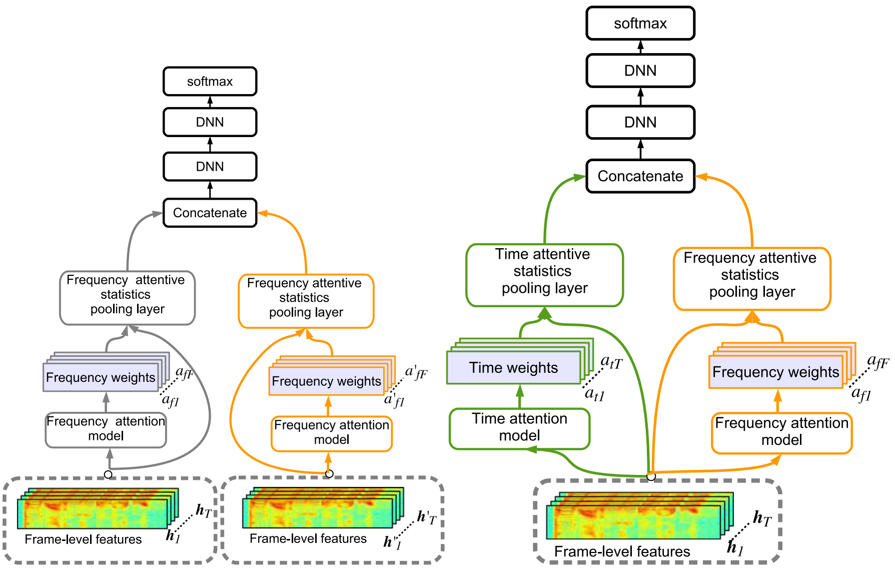
$$\mathbf{T}_A = softmax(ReLU(\mathbf{H}^T \mathbf{W}_1^t)\mathbf{W}_2^t) \tag{4}$$

where $\mathbf{W}_1^t$ is a $d_h \times d_a$ sized matrix where d_a is the attention vector dimension. $\mathbf{W}_2^t$ is a $d_a \times 1$ sized vector. The ReLU used here could be replaced by other nonlinear activation functions. The $softmax()$ is performed column-wise so that each column vector of $\mathbf{T}_A$ is an annotation vector that represents the weights for different $\mathbf{h}_t$. The weighted means $\mathbf{E}$ are then finally obtained using;

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

2.4.2 Frequency Attention Network

A traditional time attention mechanism [6,11,23,34] relates positions along the time axis to temporal dependencies. However, we believe that language features are also dis-



(a) CLF-ATN **(b)** TF-ATN

Fig. 2 A block diagram showing the proposed **a** cross-layer and **b** time–frequency attention architectures

criminative in frequency. This is motivated by the observation that different languages display different correlations of frequency over time. Therefore, we propose creating a *frequency attention* that may be beneficial to the language model—effectively this proposal means that we change the orientation of the attention mechanism from the time-domain axis to the frequency domain axis. Motivated by this, we propose a modified attention block, illustrated in the orange right-hand path in Fig. 2b,

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

where $\mathbf{W}_1^f$ is a $d_h \times d_a$ sized matrix, $\mathbf{W}_2^f$ is a $d_a \times d_f$ sized vector, where the dimension d_f varies between 2 and 32 in the following experiments. As with Eq. (4), ReLU could be replaced by a different nonlinear activation function if desired. Again, a $softmax()$ is performed, but this time it operates rank-wise: Each row of $\mathbf{F}_A$ is an annotation vector that represents the weights for different frequency bands $\mathbf{h}_b$. For example, if the number of hidden nodes is 1500 and d_f is 4, there are four frequency bands [1 : 375], [376 : 750], [751 : 1125] and [1126 : 1500]. Then if $\mathbf{F}_A$=[a_{f1}, a_{f2}, a_{f3}, a_{f4}], it implies that $a_{f1} \times h \in [1 : 375]$, $a_{f2} \times h \in [376 : 750]$, $a_{f3} \times h \in [751 : 1125]$, and $a_{f4} \times h \in [1126 : 1500]$. Mean and variance statistics are computed as usual, but from the frequency weighted h rather than unweighted or time-weighted h.

2.4.3 Cross-Layer Frequency Tensor Attention Network

In this study, we also investigate using multi-scale frame-level features to calculate weights. We employ two different attention models by inputting the low-level and high-level frame feature, respectively, which is shown in the top part of Fig. 1. The output of the hidden layer before the pooling layer is a high-level low-resolution frame feature $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T\}$, and the low-resolution annotation $\mathbf{A}$ is then tensor of size $[F \times T] \times C$. The Conv3 is a single convolutional layer without a nonlinearity, performing frequency downsampling. Therefore, the output before the Conv3 layer is a low-level high-resolution frame feature $\mathbf{H}' = \{\mathbf{h}'_1, \mathbf{h}'_2, \ldots, \mathbf{h}'_T\}$ and the high-resolution annotation $\mathbf{B}$ is a tensor of size $[[2F] \times T] \times C'$.

To make use of the frame features from different layers, we therefore employ two different single frequency attention models to generate two sets of the attentive frequency weights, by attending to low-resolution annotations and high-resolution annotations, respectively. These are then concatenated to allow the system to produce cross-layer attentive statistics, shown in Fig. 2a.

2.4.4 Time–Frequency Attention Network

In the belief that latent language-discriminative information is contained in both the time and frequency domains, we construct a system which calculates time attention $s^t(u)$ and frequency attention $s^f(u)$ separately. Scoring is performed using the time-weighted $\mathbf{h}$ and the frequency-weighed $\mathbf{h}$, respectively, and then we combine results in a score-level fusion, shown in Fig. 2b. For example, the fused score $s(u)$ of utterance u can be calculated as follows:

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

where $s()$ is the overall scoring function and α is a balancing parameter which allows a trade-off in setting the relative importance of time and frequency to the LID task. In the following experiments, $\alpha = 0.5$ to equally weight time and frequency importance.

3 Testing Methodology

3.1 Corpus

The baseline ASR DNN used in this evaluation is trained on roughly 1000 h of clean English telephone speech from Fisher. For the LID task, we conduct experiments using NIST LRE07 which is a well-established 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 30-s, 10-s, and 3-s conditions, with each condition having 2158 utterances. We also used training data augmentation [27] to increase the amount and diversity of the existing training data, including additive noise (MUSAN dataset) and reverberation (RIR dataset).

3.2 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. All the utterances are split into short segments of no more than 120 s long. 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 (comprising 40-dimensional PLP features concatenated 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 the Pnorm is 39-dimensional, and the BNFs are extracted from the subsequent normalization. The output layer has 5560 nodes, and the dimension of the i-vector is 600. The experiments for ASR DNN and i-vector extraction are all carried out using Kaldi [24].

The features are 23-dimensional MFCCs with a frame-length of 25 ms, mean-normalized over a sliding window of up to 3 s. 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 [27]. The WRN x-vector configuration is given in Table 1 and is similar to the ASR system presented in [31]. The WRN structure, shown in Fig. 1, is shallow, with a convolutional layer at the beginning of the network without a nonlinearity followed by three non-bottleneck (NBN) blocks, with a single convolutional layer of 3 × 3 filters after the first and third blocks (again without a nonlinearity) to perform downsampling. These are followed by three bottleneck (BN) layers before spatial averaging and projection. Each NBN block consists of six layers, comprising a ReLU, batch normalization layer, then a convolution, all repeated twice, and with an internal feed-forward piecewise addition path which allows some information to bypass the internal convolutional layers. The BN blocks are more complex, each comprising 9 layers that repeat ReLU, batch normalization and convolution three times. Another internal feed-forward piecewise addition path is provided to again allow some information to bypass the convolutional layers.

Table 1 Detailed feature sizes and layer-wise structure of the WRN system

Layer type	Output	Filter	Downsample	Channel	Blocks
Conv1	$23 \times T$	3×3	False	12	–
NBN block	$23 \times T$	–	False	12	2
Conv2	$12 \times T$	3×3	True	32	–
NBN block	$12 \times T$	–	False	32	2
Conv3	$6 \times T$	3×3	True	256	–
BN blocks	$6 \times T$	–	False	256/128/256	3
Spatial average layer	$6 \times T$	–	False	256	–

Table 2 Performance results for various i-vector baselines

System	3 s		10 s		30 s	
	C_avg	EER	C_avg	EER	C_avg	EER
GMM i-vector	18.49	16.12	9.31	7.04	4.18	2.50
DNN i-vector	13.46	10.75	4.28	3.24	1.39	1.11
DBF+DNN	9.35	9.73	3.17	4.94	1.11	0.78
DNN x-vector	9.16	9.03	3.05	2.96	1.28	1.44
WRN x-vector	9.22	8.38	2.78	2.59	1.21	1.43

The difference between the WRN x-vector system and the cross-layer tensor system CLF-ATN is highlighted in Fig. 1, which shows the attention information collected at the top of the figure from the output of the final NBN block and from the output of the final BN block. In each case the output is subject to a spatial averaging layer. The former output has dimension $12 \times T$ per channel, whereas the latter has reduced dimension of $6 \times T$ per channel. Each block of information is then post-processed and used for classification in the network shown in Fig. 2a.

4 Results

4.1 Single Feature Performance

4.1.1 Baseline Performance

We evaluate traditional LID systems alongside the proposed WRN systems from Sect. 2, with the results listed in Table 2 in terms of two common scoring mechanisms, C_avg [13] and EER (where the false acceptance rate equals the missed detection rate), for 3-s, 10-s, and 30-s, tasks.

Examining these results, it is clear that the best performing method for the 30-s tasks is the DBF DNN i-vector approach utilizing an ASR DNN to extract acoustic–phonetic discriminative DBFs from which it can obtain phonetic-aware statistics using the output posteriors of the ASR DNN. The DNN x-vector system achieves better performance for the 3-s and 10-s tasks, while the WRN x-vector yields further improvement for the 3-s and 10-s tasks (except 3 s C_avg performance), as it strengthens feature extraction to learn directly from LID labels. These results lend some confidence to the idea that GMM/DNN/DBF DNN i-vector methods are not able to reliably estimate i-vectors from short utterances, i.e. when statistics may be insufficient.

4.1.2 Time Attention and Frequency Attention

We then evaluate the proposed time attention and frequency attention methods (with a span of dimensions tested) in the same way. Results are shown in Table 3 alongside the WRN x-vector baseline. Firstly, we can see immediately that incorporating time attention (T-ATN) can outperform the WRN x-vector system at all timescales for

Table 3 Performance results for frequency attention on various wide residual networks (WRN)

System	3 s		10 s		30 s	
	C_avg	EER	C_avg	EER	C_avg	EER
WRN x-vector	9.22	8.38	2.78	2.59	1.21	1.43
T-ATN	8.89	7.92	2.98	2.40	1.32	1.15
F-ATN-2D	8.64	7.92	2.45	2.17	0.84	0.92
F-ATN-4D	8.68	8.43	2.78	2.50	0.77	0.88
F-ATN-8D	8.81	8.24	2.62	2.54	0.92	1.01
F-ATN-16D	8.79	8.38	2.93	2.59	0.74	0.83
F-ATN-32D	8.51	8.15	2.74	2.40	1.00	0.96

Table 4 Performance results for cross-layer time and frequency attention on various wide residual networks (WRN)

System	3 s		10 s		30 s	
	C_avg	EER	C_avg	EER	C_avg	EER
T-ATN	8.89	7.92	2.98	2.40	1.32	1.15
CLT-ATN	9.11	8.43	2.43	2.50	1.19	1.25
F-ATN-2D	8.64	7.92	2.45	2.17	0.84	0.92
CLF-ATN-2D	8.24	7.83	2.65	2.54	1.05	1.06
CLF-ATN-4D	8.45	8.24	2.38	2.27	0.88	0.88
CLF-ATN-8D	8.33	8.52	2.21	2.17	0.86	0.88
CLF-ATN-16D	8.65	8.35	2.63	2.58	0.80	0.97
CLF-ATN-32D	8.44	8.15	2.85	2.68	1.12	1.11

EER performance. Secondly, the frequency attention mechanism (F-ATN) alone yields improvement over the time attention mechanism, although it is weakly sensitive to feature dimension, performing best overall with two-dimensional features (F-ATN-2D), except for C_avg performance over 3 s, C_avg and EER performance over 30 s.

4.1.3 Cross-Layer Frequency Tensor Network

The cross-layer frequency tensor network is explored in Table 4. Results are given separately for T-ATN, its cross-layer variant CLT-ATN, F-ATN, and several cross-layer variants. For convenience, F-ATN-2D was selected for comparison from the previous evaluation (Table 3). However, there is no guarantee that 2D will also be the best choice for operating cross-layer, and hence, we evaluate a range of dimensions for the cross-layer frequency systems.

CLT-ATN does not appear to show consistent further improvement on the whole, although several dimensions of CLF-ATN can outperform the 10-s and 30-s results for F-ATN. We surmise that this is because there is no time downsampling for WRN, and hence, the time attention tensors from different layers have not lost any time information. By contrast, CLF-ATN is able to preserve more language detail from the

Table 5 Performance results for time–frequency attention on various wide residual networks (WRN)

System	3 s		10 s		30 s	
	C_avg	EER	C_avg	EER	C_avg	EER
TF-ATN-2D	9.01	7.36	2.87	2.17	1.13	1.01
TF-ATN-4D	8.83	7.46	2.75	2.08	1.05	0.92
TF-ATN-8D	8.72	7.64	2.56	**1.94**	0.88	**0.78**
TF-ATN-16D	9.08	7.50	2.80	2.22	0.88	0.97
TF-ATN-32D	8.79	**7.27**	2.89	2.22	1.06	0.97
DBF+DNN	9.35	7.73	3.17	4.94	1.11	**0.78**
CLF-ATN-8D	**8.33**	8.52	**2.21**	2.17	**0.86**	0.88

The bold values indicate the best performance for each test condition

frequency domain by having the higher resolution earlier feature, combined with the lower resolution but more refined later feature.

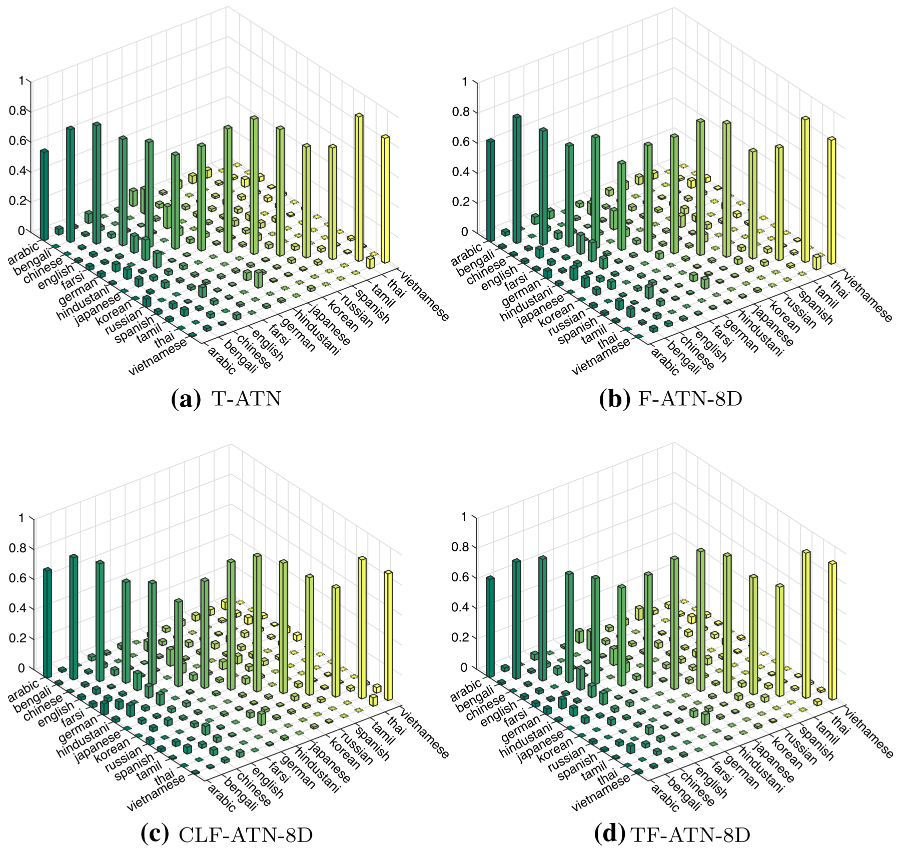
4.1.4 Feature Score Fusion

As a final step of evaluation, Table 5 examines the fusion of classification scores from time and frequency domain feature systems. Again, the frequency domain results are given for a range of different frequency dimensions. At the bottom of the table, we provide a comparison to the state-of-the-art results from the DBF DNN i-vector system and from the best performing cross-layer tensor attention network of Table 4. Overall best results are presented in bold font. Clearly, these results reveal that TF fusion can yield further advantage at all timescales in terms of EER but not C_avg, where the cross-layer tensor system significantly outperforms all other systems. In summary, the eight-dimensional frequency feature fusion with time attention (TF-ATN-8D) provides the overall best EER result while the eight-dimensional cross-layer frequency tensor network (CLF-ATN-8D) yields the overall best results for C_avg. Exploiting both time and frequency information definitely improves score, indicating that there may be some complementarity in the information they extract, but slightly different arrangements of time–frequency combination perform best in the two scoring mechanisms.

4.1.5 Further Analysis

Further analysis is conducted to examine confusion matrices of various systems in Fig. 3. Clockwise from top left, bar plots highlight confusion patterns for time attention, frequency attention, cross-layer tensor attention and time–frequency attention score fusion, with all frequency networks operating with a dimension of 8 on the most challenging 3-s task. Presented with normalized scores, fully correct results would result in a diagonal of 14 full-height bars. We have rotated plots to provide a clearer view of off-diagonal elements.

The first thing to note from Fig. 3 is that all systems perform similarly in terms of true-positives (i.e. the diagonals)—this is borne out by comparing their 3-s C_avg scores from the previous tables, which are very similar at 12.69, 12.55, 12.21, and



(a) T-ATN

(b) F-ATN-8D

(c) CLF-ATN-8D

(d) TF-ATN-8D

Fig. 3 Confusion matrices for various time and frequency attention on the 3-s LID task

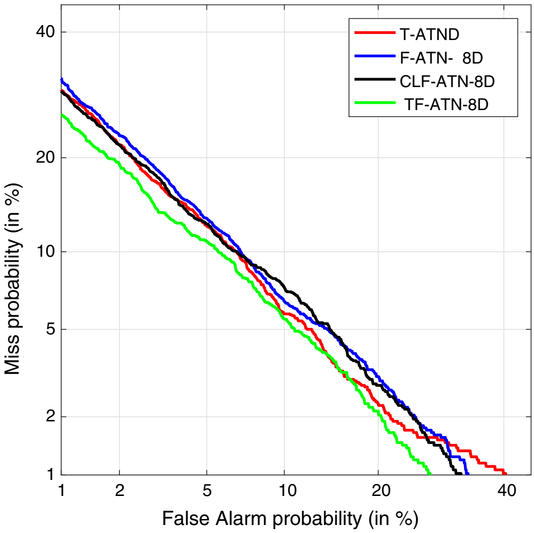
11.15, respectively. The plots show that the class performance distribution within those scores is similar. What is much more different though, is the scattering of errors. While some errors (e.g. Hindustani–Tamil or Vietnamese–Thai) are ever-present due to the similar nature of the languages, some larger errors are entirely absent in some systems (e.g. Hindustani–Japanese in F-ATN but absent in T-ATN, and vice versa with Korean–Arabic, along the back row of bars). Such complementarity between languages provides the resource that the TF fusion systems exploit to improve performance.

Finally, Fig. 4 presents DET curves from the same systems on the same task to indicate that the advantage enjoyed by the TF feature score fusion system over other systems extends to all operating points on the false-positive and false-negative continuum.

5 Conclusion

This paper first presented a new end-to-end LID architecture named WRN x-vector that utilize a WRN front-end for extracting language-discriminative information and

Fig. 4 DET curves of the time and frequency attention systems for the 3-s LID task



then using traditional statistics pooling methods and fully connected output classifiers. Performance was shown to exceed that of traditional DNN x-vector architectures on the LRE2007 task. We then proposed incorporating a time attention mechanism in the WRN x-vector system, which was shown to further improve performance. Next, we proposed an attention mechanism operating in the frequency domain, which we called F-ATN. This yielded a small performance improvement over the time attention mechanism on LRE2007, but to further improve results we proposed a novel cross-layer frequency tensor attention network (CLF-ATN). Results on several dimensions of CLF-ATN demonstrated that it can outperform F-ATN. We also investigated combining time and frequency domain-discriminative features. Evaluating score-level fusion of the two attention mechanisms, we noted the best performance improvement, especially for EER performance. Confusion matrices showed that the time attention and frequency attention were effective in combatting different classes of false-error, a complementarity that was exploited effectively in the time–frequency system. An evaluation of DET curves showed that the best performing TF-ATN-8D system outperformed the other systems over a range of trade-off points. The novel frequency attention mechanism proposed in this paper combined with time-domain attention and a multi-resolution cross-layer approach on a wide residual network, which itself was found to perform well for an LID task, yield results which improve upon the current state-of-the-art DBF DNN i-vector approach.

References

1. E.P. Altenberg, C.T. Ferrand, Fundamental frequency in monolingual English, bilingual English/Russian, and bilingual English/Cantonese young adult women. J. Voice **20**(1), 89–96 (2006)
2. S.N. Awan, P.B. Mueller, Speaking fundamental frequency characteristics of white, African American, and Hispanic kindergartners. J. Speech Lang. Hear. Res. **39**(3), 573–577 (1996)
3. D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2016), pp. 4945–4949
4. W. Cai, Z. Cai, W. Liu, X. Wang, M. Li, Insights into end-to-end learning scheme for language identification (2018). arXiv preprint arXiv:1804.00381
5. W. Cai, Z. Cai, X. Zhang, X. Wang, 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
6. W. Cai, J. Chen, M. Li, Exploring the encoding layer and loss function in end-to-end speaker and language recognition system (2018). arXiv preprint arXiv:1804.05160
7. W.M. Campbell, D.E. Sturim, D.A. Reynolds, Support vector machines using GMM supervectors for speaker verification. IEEE Signal Process. Lett. **13**(5), 308–311 (2006)
8. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation (2014). arXiv preprint arXiv:1406.1078
9. N. Dehak, P.A. Torres-Carrasquillo, D. Reynolds, R. Dehak, Language recognition via i-vectors and dimensionality reduction, in *Twelfth Annual Conference of the International Speech Communication Association* (2011)
10. D. Garcia-Romero, A. McCree, Stacked long-term TDNN for spoken language recognition, in *Prof. Interspeech* (2016), pp. 3226–3230
11. 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, in *Proceedings of Interspeech* (2016)
12. J. Gonzalez-Dominguez, I. Lopez-Moreno, H. Sak, J. Gonzalez-Rodriguez, 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)
13. N.L. Group et al., The 2007 NIST language recognition evaluation plan (LRE07) (2007)
14. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 770–778
15. G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
16. B. Jiang, Y. Song, S. Wei, J.H. Liu, I.V. McLoughlin, L.R. Dai, Deep bottleneck features for spoken language identification. PLoS One **9**(7), e100795 (2014)
17. B. Jiang, Y. Song, S. Wei, M.G. Wang, I. McLoughlin, L.R. Dai, Performance evaluation of deep bottleneck features for spoken language identification, in *2014 9th International Symposium on Chinese Spoken Language Processing (ISCSLP)* (IEEE, 2014), pp. 143–147
18. M. Jin, Y. Song, I. McLoughlin, L.R. Dai, LID-senones and their statistics for language identification. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(1), 171–183 (2018)
19. P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, J. Alam, Deep neural networks for extracting Baum–Welch statistics for speaker recognition, in *Proceedings of Odyssey* (2014), pp. 293–298
20. Y. Lei, N. Scheffer, L. Ferrer, M. McLaren, A novel scheme for speaker recognition using a phonetically-aware deep neural network, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2014), pp. 1695–1699
21. I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, 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
22. I. Mennen, F. Schaeffler, G. Docherty, Cross-language differences in fundamental frequency range: a comparison of English and German. J. Acoust. Soc. Am. **131**(3), 2249–2260 (2012)
23. K. Okabe, T. Koshinaka, K. Shinoda, Attentive statistics pooling for deep speaker embedding (2018). arXiv preprint arXiv:1803.10963

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, EPFL-CONF-192584* (IEEE Signal Processing Society, 2011)
25. F. Richardson, D. Reynolds, N. Dehak, Deep neural network approaches to speaker and language recognition. IEEE Signal Process. Lett. **22**(10), 1671–1675 (2015)
26. F. Richardson, D. Reynolds, N. Dehak, A unified deep neural network for speaker and language recognition (2015). arXiv preprint arXiv:1504.00923
27. D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, S. Khudanpur, Spoken language recognition using x-vectors (2018) **(submitted to Odyssey)**
28. Y. Song, X. Hong, B. Jiang, R. Cui, I. McLoughlin, L.R. Dai, Deep bottleneck network based i-vector representation for language identification, in *Sixteenth Annual Conference of the International Speech Communication Association* (2015)
29. Y. Song, B. Jiang, Y. Bao, S. Wei, L.R. Dai, I-vector representation based on bottleneck features for language identification. Electron. Lett. **49**(24), 1569–1570 (2013)
30. Y. Wang, X. Deng, S. Pu, Z. Huang, Residual convolutional CTC networks for automatic speech recognition (2017). arXiv preprint arXiv:1702.07793
31. Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, S. Khudanpur, Backstitch: counteracting finite-sample bias via negative steps, in *Interspeech* (2017), pp. 1631–1635
32. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, in *International conference on machine learning* (2015), pp. 2048–2057
33. S. Zagoruyko, N. Komodakis, Wide residual networks (2016). arXiv preprint arXiv:1605.07146
34. Y. Zhu, T. Ko, D. Snyder, B. Mak, D. Povey, Self-attentive speaker embeddings for text-independent speaker verification, in *Proceedings of Interspeech* (2018), pp. 3573–3577

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

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

Ian McLoughlin
ivm@kent.ac.uk

Yonghong Yan
yonghong.yan@hccl.ioa.ac.cn

[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, Beijing, China

[3] University of Chinese Academy of Sciences, Beijing, China

[4] Xinjiang Key Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Science, Ürümqi, China