Computer Speech & Language 60 (2020) 101026

Contents lists available at ScienceDirect

Computer Speech & Language

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

State-of-the-art speaker recognition with neural network embeddings in NIST SRE18 and Speakers in the Wild evaluations

Jesús Villalba[a,*], Nanxin Chen[a], David Snyder[a,b], Daniel Garcia-Romero[b], Alan McCree[b], Gregory Sell[b], Jonas Borgstrom[c], Leibny Paola García-Perera[a], Fred Richardson[c], Réda Dehak[d], Pedro A. Torres-Carrasquillo[c], Najim Dehak[c]

[a] *Center for Language and Speech Processing, Johns Hopkins University, Baltimore MD, USA*
[b] *Human Language Technology Center of Excellence, Johns Hopkins University, Baltimore MD, USA*
[c] *MIT Lincoln Laboratory, Lexington MA, USA*
[d] *LSE-EPITA, Villejuif, France*

ARTICLE INFO

Article History:
Received 6 May 2019
Revised 10 September 2019
Accepted 26 September 2019
Available online 9 October 2019

Keywords:
Speaker recognition
Embeddings
X-Vectors
NIST SRE18
SITW
Domain adaptation
Evaluations
Calibration

ABSTRACT

We present a thorough analysis of the systems developed by the JHU-MIT consortium in the context of NIST speaker recognition evaluation 2018. In the previous NIST evaluation, in 2016, i-vectors were the speaker recognition state-of-the-art. However now, neural network embeddings (a.k.a. x-vectors) rise as the best performing approach. We show that in some conditions, x-vectors' detection error reduces by 2 w.r.t. i-vectors. In this work, we experimented on the Speakers In The Wild evaluation (SITW), NIST SRE18 VAST (Video Annotation for Speech Technology), and SRE18 CMN2 (Call My Net 2, telephone Tunisian Arabic) to compare network architectures, pooling layers, training objectives, back-end adaptation methods, and calibration techniques. x-Vectors based on factorized and extended TDNN networks achieved performance without parallel on SITW and CMN2 data. However for VAST, performance was significantly worse than for SITW. We noted that the VAST audio quality was severely degraded compared to the SITW, even though they both consist of Internet videos. This degradation caused strong domain mismatch between training and VAST data. Due to this mismatch, large networks performed just slightly better than smaller networks. This also complicated VAST calibration. However, we managed to calibrate VAST by adapting SITW scores distribution to VAST, using a small amount of in-domain development data.

Regarding pooling methods, learnable dictionary encoder performed the best. This suggests that representations learned by x-vector encoders are multi-modal. Maximum margin losses were better than cross-entropy for in-domain data but not in VAST mismatched data. We also analyzed back-end adaptation methods in CMN2. PLDA semi-supervised adaptation and adaptive score normalization (AS-Norm) yielded significant improvements. However, results were still worse than in English in-domain conditions like SITW.

We conclude that x-vectors have become the new state-of-the-art in speaker recognition. However, their advantages reduce in cases of strong domain mismatch. We need to investigate domain adaptation and domain invariant training approaches to improve performance in all conditions. Also, speech enhancement techniques with a focus on improving the speaker recognition performance could be of great help.

*Corresponding author.
E-mail address: jvillalba@jhu.edu (J. Villalba).

https://doi.org/10.1016/j.csl.2019.101026
0885-2308/

1. Introduction

The National Institute of Standards and Technology (NIST) regularly conducts speaker recognition evaluations (SRE) to assess the state-of-the-art of the technology (Doddington, 2000). These evaluations focus on the speaker detection task, i.e., given one or more enrollment recordings and a test recording, we need to decide whether the enrollment speaker is also present in the test. Along the years, NIST has been increasing the difficulty of the evaluation conditions. First SRE campaigns were only centered on telephone conversational speech (Martin and Przybocki, 2001; Przybocki et al., 2007). In SRE08-12, NIST introduced far-field microphone interview speech (Brandschain et al., 2010; Villalba et al., 2013; Martin et al., 2014). SRE16 brought significant changes (Sadjadi et al., 2017). Although it focused again on telephone speech, for the first time, the data was non-English speech collected outside North America. This was a major difficulty since the training data was mainly English speech collected in the US. Just a small amount of unlabeled adaptation data was provided to correct distribution shift due to language and channel mismatch. For the recent SRE18 (Sadjadi et al., 2019), NIST decided to maintain the non-English condition. This time, they selected Arabic language collected in Tunisia through public switched telephone network (PSTN) and voice over IP (VOIP). This data was extracted from the *Call My Net 2* corpus (CMN2). Furthermore, NIST added a new condition including speech from amateur Internet videos extracted from the Video Annotation for Speech Technology corpus (VAST) (Tracey and Strassel, 2018). As consequence, VAST spans a wide range of quality levels, including noise, reverberation and other artifacts that complicate speaker verification. These recordings usually contain multiple speakers so diarization was required to isolate the speaker of interest. At the same time, other organizations have promoted evaluations using their own datasets. For example, SRI International organized the Speakers In The Wild evaluation using speech from Internet videos (McLaren et al., 2016b).

In this paper, we present a thorough analysis of the systems developed by the JHU-MIT consortium in the context of NIST SRE18. All systems consisted of a neural network embedding with some form of global temporal pooling (a.k.a. x-vector) (Snyder et al., 2018b) or i-vector (Dehak et al., 2011) followed by some form of probabilistic linear discriminant analysis (PLDA) (Kenny, 2010) back-end. We will show how the x-vector approach has become the new state-of-the-art for speaker recognition. We base our analysis on experiments on SITW, NIST SRE18 VAST and CMN2 datasets. The JHU-MIT systems used two different training/development setups, referred as JHU-CLSP[1] and JHU-HLTCOE[2] setups. We focus on the JHU-CLSP setup for our analysis, since most of our systems were based on it. The contributions of the paper are the following. We introduce the factorized TDNN (F-TDNN) network topology, recently proposed for speech recognition (Povey et al., 2018), for x-vector speaker recognition. We compare F-TDNN with other network topologies such us TDNN (Snyder et al., 2017; 2018b), extended TDNN (E-TDNN) (Snyder et al., 2019), and ResNet (2D convolutions) (He et al., 2015) in the common setup. We also compare pooling methods-mean plus standard deviation, learnable dictionary encoder (LDE) (Cai et al., 2018b) and multi-head attention- with the common network topology. We introduce *Additive Angular Margin* loss, recently proposed for face recognition (Deng et al., 2019), to speaker recognition and compare with softmax and angular softmax (Liu et al., 2017) cross-entropy. We analyze the impact of different domain adaptation methods in the back-end, when adapting from English to Tunisian Arabic condition in the CMN2 dataset. Furthermore, one of the main difficulties of the VAST data was the domain mismatch with respect to previous evaluations, which, among other things, complicates score calibration. We show how to perform calibration with limited amount of in-domain data (37 files). Finally, we analyze the effectiveness of score-level fusion with these systems.

The rest of the paper is organized as follows. Section 2 introduces work related to embeddings for speaker and language recognition. Section 3 presents the x-vector approach including network topologies, pooling methods and training objectives. Sections 4 and 5 remind the i-vector model and PLDA. Section 6 describes the training, development and evaluation data. Section 7 describes the experimental setup including acoustic features, network and back-end hyper-parameters, diarization setup, metrics, calibration and fusion. Section 9 analyzes the results for SITW and SRE18 VAST data and Section 10 for SRE18 CMN2 data. Finally, Section 11 presents the conclusions.

2. Related work

Two decades ago, state-of-the-art speaker recognition was based on the GMM-UBM approach (Reynolds et al., 2000). In this approach, a universal background model (UBM) represents the acoustic features distribution of a large population of speakers. In enrollment phase, GMM of individual speakers are adapted from the UBM. Meanwhile in the test phase, we compute the likelihood ratio of the test acoustic features given the enrollment and the universal GMMs. This method has the inconvenience of being sensitive to inter-session variability. Furthermore, the i.i.d. assumption and the fact that each recording has different length produce badly calibrated likelihoods. These issues force us to resort to score normalization methods to improve results. The joint factor analysis (JFA) approach improves the way in which the target GMM is estimated and made trial evaluation more robust to inter-session variability (Kenny et al., 2007). However, score normalization is still needed to achieve good performance. With the advent of i-vectors (Dehak et al., 2011), each recording is compressed into a unique fixed length embedding. This embedding becomes the new feature for downstream classifiers. First, cosine scoring with linear discriminant analysis (LDA) is proposed to compare enrollment and test i-vectors. Later, authors introduce generative probabilistic LDA (PLDA) (Kenny, 2010; Brümmer and De Villiers, 2010). PLDA has the advantage of producing well-calibrated likelihood ratios without requiring score normalization-when training and evaluation data are drawn from the same domain. Also, discriminative PLDA (DPLDA) (Burget et al., 2011) and Pairwise Support Vector Machines (PSVM) are proposed (Cumani et al., 2011) as an alternative to generative models.

[1] Johns Hopkins University Center for Speech and Language Processing
[2] Johns Hopkins University Human Language Technology Center of Excellence

Recently, there has been a surge of interest in applying neural networks to speaker recognition. The first successful attempt consists in using a DNN trained for ASR to compute the Gaussian component responsibilities of the i-vector model (DNN i-vector) (Lei et al., 2014). Each senone unit of the ASR model is associated with one Gaussian of the i-vector model. This obliges to use a very large i-vector model with more than 3000 components. In order to reduce the number of GMM components, using a bottleneck layer from the ASR network as feature is proposed (BNF i-vector) (Matejka et al., 2014; Richardson et al., 2015). Bottleneck features (BNF) are appended to the acoustic features, and then fed to a standard GMM-UBM i-vector. Including the BNF helps the EM algorithm to produce better clusters of the feature space, more correlated to phonetic information.

More recent works approach the idea of completely removing GMM-UBM and i-vector models and compute sequence embeddings based solely on deep networks. The first works in this area focus on text-dependent speaker recognition. In Variani et al. (2014), a neural network is optimized to classify acoustic features into a set of training speakers. In evaluation phase, an embedding is obtained by averaging the representations from the last hidden layer. This approach is termed as d-vector. In Heigold et al. (2016), d-vectors are improved by classifying a full sequence during training. The frame-level representations are aggregated using a LSTM layer, which generates the sequence embedding in the last time step. For text-independent speaker recognition, Snyder et al. (2016) uses a Siamese network trained to discriminate between target and non-target trials (verification loss). Frame level representations are aggregated by a pooling layer computing their mean and standard deviation. This method requires more than 100k training speakers to improve w.r.t. i-vectors. The work in Zhang and Koishida (2017) proposes a similar scheme but using triplet loss to increase the margin between embeddings of different speakers. In Snyder et al. (2017), the authors propose to use embeddings networks with global temporal pooling trained on categorical cross entropy loss rather than verification loss. This approach, termed as x-vector, provides a moderate improvement over i-vectors on NIST SRE10. Later, performance is improved by aggressively augmenting the x-vector training data with additive and convolutional noise (Snyder et al., 2018b). Augmentation allows x-vectors to surpass the i-vector performance definitively. x-Vectors are also useful for language identification (McCree et al., 2018; Snyder et al., 2018a; Villalba et al., 2018) and speaker diarization (Garcia-Romero et al., 2017; Sell et al., 2018).

Several variants of x-vectors have been published studying alternative pooling methods and training objectives. Rezaur rahman Chowdhury et al. (2018) proposes to use attention to compute a weighted average of frame level representations. The authors compare several attention weights computation methods. Zhang et al. (2017a) proposes to use multiple attention heads. The learnable dictionary encoder (LDE) pooling layer (Cai et al., 2018b) assumes multi-modal frame level deep representations, i.e., distributed in multiple clusters, and uses a GMM to obtain an embedding per Gaussian component. Single component embedding are stacked together to form a super-vector, which is projected to a lower dimension to produce the final embedding. Also, angular softmax (Liu et al., 2017) is proposed to maximize angular margin between speaker embeddings.

3. Neural network embeddings

In this work, we unified under the term *x-vectors* all neural embeddings which include some form of global temporal pooling and are trained to identify the speakers in a set of training recordings (Snyder et al., 2017; 2018b). x-Vector networks are divided into three parts. First, an encoder network extracts frame level representations from acoustic features such MFCC or filter-banks. This is followed by a global temporal pooling layer that aggregates the frame-level representation into a single vector per utterance. Finally, a feed forward classification network processes this single vector to calculate speaker class posteriors. Typically, in the evaluation phase, the embedding (x-vector) is extracted from the first affine transform after the pooling layer. Meanwhile, the rest of layers after the embedding layer are discarded. Different x-vector systems are characterized by different encoder architectures; pooling methods and training objectives. In the following sections, we describe the topologies, pooling layers and training losses that we used in our investigation.

3.1. Encoder networks

3.1.1. TDNN

Time delay neural networks (Waibel et al., 1989) are the baseline encoder architecture that you can find in Kaldi x-vector examples (Snyder et al., 2018b). Table 1 summarizes the TDNN x-vector network used in our experiments. Each feature frame is

Table 1
Baseline TDNN x-vector architecture.

Layer	Layer type	Context	Size
1	TDNN-ReLU	$t-2$:$t+2$	512
2	TDNN-ReLU	t−2, t, $t+2$	512
3	TDNN-ReLU	$t-3$, t, $t+3$	512
4	Dense-ReLU	t	512
5	Dense-ReLU	t	1500
6	Pooling (mean+stddev)	Full-seq	2×1500
7	Dense (x-vector)-ReLU		512
8	Dense-ReLU		512
9	Dense-Softmax		Num. spks.

processed by a sequence of time-delay layers (Peddinti et al., 2015). Time delay layers are equivalent to one dimensional dilated convolutions. Therefore, frame-level representations at each layer aggregate information from a context of past and future frames. The deeper we go into the network, the wider the context. By dilating the convolution, we can obtain very large contexts while using small kernels. The total context in this network was 15 frames.

The pooling layer computes mean and standard deviation of the TDNN output over time to obtain a unique representation per recording. Afterwards, the pooling output is projected to a lower dimension to calculate the speaker embedding. The output of the network are posterior probabilities for the training speakers.

3.1.2. E-TDNN

The Extended TDNN architecture (E-TDNN) has been introduced in Snyder et al. (2019). The authors optimized this architecture to balance the trade off between performance and network parameters. Table 2 summarizes the E-TDNN topology. The two main differences w.r.t. TDNN are a slightly wider temporal context of the TDNN (22 frames) and interleaving dense layers in between the convolutional layers (equivalent to the 1×1 convolutions used in computer vision architectures). The x-vector is extracted from layer 12 prior to the ReLU non-linearity. This architecture has been found to greatly outperform the baseline TDNN in the SITW and SRE16 benchmarks.

3.1.3. F-TDNN with skip connections

The factorized TDNN (F-TDNN) with skip connections (Povey et al., 2018) topology is depicted in Fig. 1. The F-TDNN reduces the number of parameters of the network by factorizing the weight matrix of each TDNN layer into the product of two low-rank matrices. Singular value decomposition (SVD) is the usual way of factorizing layers in already trained neural networks. Instead of using SVD, F-TDNN factors are trained from a random start. The first of those factors is constrained to be semi-orthogonal, which helps to ensure that we do not lose information when projecting from the high dimension to the low-rank dimension.

Povey et al. (2018) also found that, instead of factorizing each TDNN layer into a convolution times a feed-forward layer, it is better to factorize the layer into two convolutions with smaller kernel size. For example, instead of using context $(-2, 0, 2)$ in the first low-rank factor and no context in the second factor, it is better to use context $(-2,0)$ in the first factor and $(0, +2)$ in the second factor.

As in other architectures, we introduced skip connections. This means that some layers receive as input, not only the previous layer but also the output from other prior layers. The prior layers were concatenated to the input of the current layer like in DenseNet (Huang et al., 2017; Han et al., 2018), instead of being added like in ResNet (He et al., 2015). Skip connections alleviate the vanishing gradient problem and facilitate creating deeper networks. We created skip connections between the low-rank interior features of the F-TDNN. Thus, we had less trainable parameters than concatenating using the full-size layers.

Table 3 summarizes the layers of our largest F-TDNN x-vector, which consists of F-TDNN layers of size 1024 with bottlenecks (inner features) of size 256. We also experimented with smaller F-TDNN with the same topology but reducing the F-TDNN layer sizes and inner sizes.

3.1.4. Resnet 2D

Here, the TDNN encoders are replaced by the 34 layers residual network (ResNet34) described in He et al. (2015). Also, MFCC features are replaced by log-Mel filter-banks. ResNet is based on two dimensional convolutions (2D-CNN), instead of 1D-CNN. ResNet consists of residual blocks like the one in Fig. 2. The residual block is composed of two 2D convolutions separated by ReLU non-linearity; The input to the block is added to the output of the second convolution and another ReLU is applied. Table 4 summarizes the topology of the ResNet34 encoder. There are 4 types of residual blocks with different number of channels in the 2D convolutions. Each type of block is repeated multiple times as indicated in the *Blocks* column of the table. Each time that we

Table 2
Extended TDNN x-vector architecture.

Layer	Layer type	Context	Size
1	TDNN-ReLU	$t-2$:$t+2$	512
2	Dense-ReLU	t	512
3	TDNN-ReLU	$t-2$, t, $t+2$	512
4	Dense-ReLU	t	512
5	TDNN-ReLU	$t-3$, t, $t+3$	512
6	Dense-ReLU	t	512
7	TDNN-ReLU	$t-4$, t, $t+4$	512
8	Dense-ReLU	t	512
9	Dense-ReLU	t	512
10	Dense-ReLU	t	1500
11	Pooling (mean+stddev)	Full-seq	2×1500
12	Dense (x-vector)-ReLU		512
13	Dense-ReLU		512
14	Dense-Softmax		Num. spks.

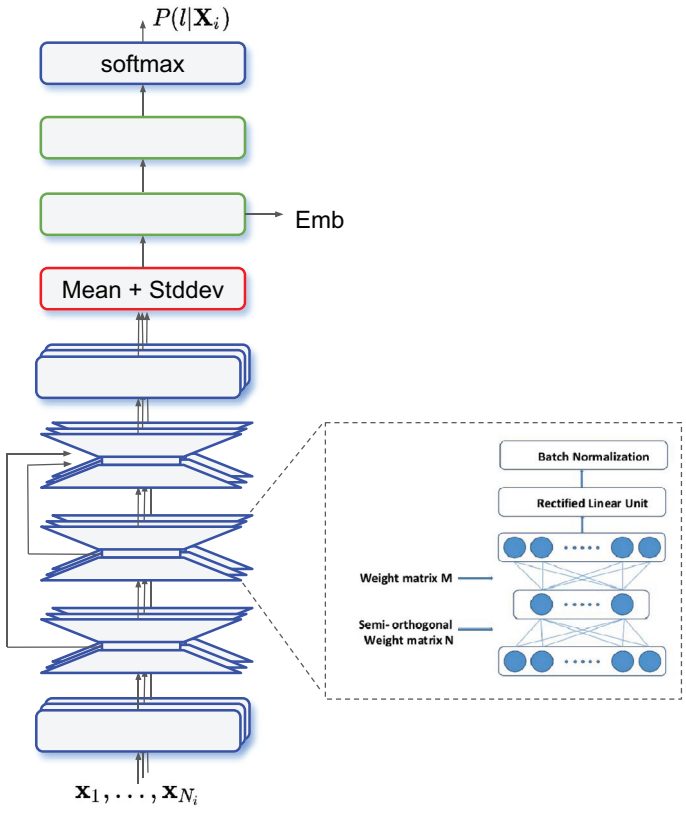


Fig. 1. Factorized TDNN x-vector architecture.

Table 3
Factorized TDNN x-vector architecture.

Layer	Layer type	Context factor1	Context factor2	Skip conn. from layer	Size	Inner size
1	TDNN-ReLU	$t-2{:}t+2$			512	
2	F-TDNN-ReLU	$t-2, t$	$t, t+2$		1024	256
3	F-TDNN-ReLU	t	t		1024	256
4	F-TDNN-ReLU	$t-3, t$	$t, t+3$		1024	256
5	F-TDNN-ReLU	t	t	3	1024	256
6	F-TDNN-ReLU	$t-3, t$	$t, t+3$		1024	256
7	F-TDNN-ReLU	$t-3, t$	$t, t+3$	2, 4	1024	256
8	F-TDNN-ReLU	$t-3, t$	$t, t+3$		1024	256
9	F-TDNN-ReLU	t	t	4, 6, 8	1024	256
10	Dense-ReLU	t	t		2048	
11	Pooling (mean+stddev)	full-seq			2×2048	
12	Dense (x-vector)-ReLU				512	
13	Dense-ReLU				512	
14	Dense-Softmax				N. spks.	

increase the number of channels of the convolution, we downsample the filter-bank and time dimensions by 2. Finally, average pooling is applied in the filter-bank dimension to get a single vector per frame.

On top of the encoder we can apply several temporal pooling methods. With this architecture, we compared three pooling methods and three training losses, described below. This was implemented in Pytorch while the other encoders were implemented in Kaldi.

3.2. Pooling methods

The original x-vector framework (Snyder et al., 2018b) assumes that the frame-level TDNN representations before pooling are uni-modal. That is, the distribution of the frame representations concentrates around a single peak value, which is different for

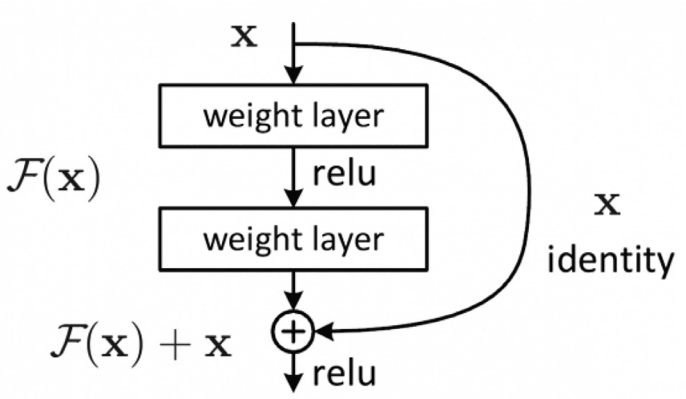


Fig. 2. 2D convolutional residual block used as basic block in ResNet. Figure taken from He et al. (2015).

Table 4
ResNet34 encoder architecture. F is the feature dimension ($F=23$ for 8 kHz systems, $F=40$ for 16 kHz systems), and T is the sequence length.

Layer	Output Size	Downsample	Channels	Blocks	Kernel
conv1	$F \times T$	False	16	–	7×7
resblock-1x	$F \times T$	False	16	3	3×3
resblock-2x	$F/2 \times T/2$	True	32	4	3×3
resblock-3x	$F/4 \times T/4$	True	64	6	3×3
resblock-4x	$F/8 \times T/8$	True	128	3	3×3
average	$1 \times T/8$	–	128	–	

each utterance. Thus, to pool those representations, we just compute their mean and standard deviation. However, other pooling methods have been proposed. The learnable dictionary encoder (LDE) pooling was first introduced in the context of texture recognition (Zhang et al., 2017b) and then adapted to language (Cai et al., 2018a) and speaker recognition (Cai et al., 2018b). The LDE layer assumes that frame level representations are distributed in C clusters and it learns a dictionary of cluster centers. This is essentially the same as we do in the GMM-i-vector paradigm. Given a representation $\mathbf{x}_t$, the responsibility of cluster c is obtained as,

$$w_{t,c} = \frac{\exp\left(-\frac{1}{2} s_c \parallel \mathbf{x}_t - \boldsymbol{\mu}_c \parallel^2 + b_c\right)}{\sum_{c=1}^{C} \exp\left(-\frac{1}{2} s_c \parallel \mathbf{x}_t - \boldsymbol{\mu}_c \parallel^2 + b_c\right)} \tag{1}$$

where s_c is an isotropic precision; and the bias b_c includes the log-weight and log-normalizing constant of the Gaussian. The bias term was not included in the original paper, but we found that it slightly improves the results.

Then, we compute an embedding per component by calculating the deviation of the component mean from its center

$$\mathbf{e}_c = \frac{\sum_{t=1}^{T} w_{t,c}(\mathbf{x}_t - \boldsymbol{\mu}_c)}{\sum_{t=1}^{T} w_{t,c}} \qquad c = 1, \cdots, \mathrm{C} \tag{2}$$

and we concatenate the embeddings for all the components $\mathbf{e} = (\mathbf{e}_1^{\mathrm{T}}, \cdots, \mathbf{e}_{\mathrm{C}}^{\mathrm{T}})^{\mathrm{T}}$. This embedding has the same role as the super-vector in GMM-i-vectors. This super-vector is projected to a lower dimension to obtain the final embedding. This projection has the same role as the total variability matrix in i-vectors.

We also evaluated *multi-head attention* pooling. There are multiple scoring functions to compute the attention weights (Rezaur rahman Chowdhury et al., 2018). We used a function similar to LDE, in which the attention to frame $\mathbf{x}_t$ from a head c is given by

$$w_{t,c} = \frac{\exp(-s_c \parallel \mathbf{x}_t - \boldsymbol{\mu}_c \parallel)}{\sum_{t=1}^{T} \exp(-s_c \parallel \mathbf{x}_t - \boldsymbol{\mu}_c \parallel)} \ . \tag{3}$$

The difference with LDE is that attention weights are normalized to sum up to one in the time dimension, instead of the component dimension. Also, we got better results using L1 distance than L2 distance. Then, the embedding for head c is just given by $\mathbf{e}_c = \sum_{t=1}^{T} w_{t,c}(\mathbf{x}_t - \boldsymbol{\mu}_c)$. Again, we concatenate the embeddings from each head to form a super-vector.

3.3. Objective functions

3.3.1. Softmax cross-entropy

Categorical cross-entropy is the usual x-vector objective, given by

$$L=-\sum_{i=1}^{M}\log P(y_i=t_i|\mathbf{X}_i) \tag{4}$$

where $\mathbf{X}_i$ are the acoustic features, y_i is the predicted speaker and t_i the true speaker label for recording i. The speaker posterior is calculated using the softmax function as

$$P(y_i=t_i|\mathbf{X}_i)=\frac{\exp\left(\mathbf{W}_{t_i}^{\mathrm{T}}\mathbf{z}_i+b_{t_i}\right)}{\sum_j\exp\left(\mathbf{W}_j^{\mathrm{T}}\mathbf{z}_i+b_j\right)} \tag{5}$$

where $\mathbf{z}_i$ is the input to the last network layer; and $\mathbf{W}_j$ and b_j are weights and bias for class j.

3.3.2. Angular softmax

We compared softmax cross-entropy with angular softmax (A-Softmax) loss, introduced for face recognition (Liu et al., 2017). To derive A-Softmax from cross-entropy, we first need to write $\mathbf{W}_j\mathbf{z}_i+b_j=\|\mathbf{W}_j\|\,\|\mathbf{z}_i\|\cos(\theta_{j,i})+b_j$, where $0\le\theta_{j,i}\le\pi$ is the angle between $\mathbf{W}_j$ and $\mathbf{z}_i$. Next, we enforce $\|\mathbf{W}_j\|=1$ and $b_j=0\ \forall j$ so the input to the softmax function is just $\|\mathbf{z}_i\|\cos(\theta_{j,i})$. Now, we incorporate an angular margin between classes to enhance the discriminative power of learned embeddings. Given $\mathbf{z}$ from class 1 and given θ_1 and θ_2 the angles between $\mathbf{z}$ and classes 1 and 2, we need $\cos(\theta_1)>\cos(\theta_2)$ –or equivalently $\theta_1<\theta_2$– to correctly classify $\mathbf{z}$. However, during training, we could require $\cos(m\theta_1)>\cos(\theta_2)$ –or equivalently $\theta_1<\theta_2/m$– with $m>1$, to consider $\mathbf{z}$ correctly classified. Imposing stronger requirements for correct classification, generates an angular classification margin between embeddings of different classes. Thus, the speaker posterior is calculated as

$$P(y_i=t_i|\mathbf{X}_i)=\frac{\exp\left(\|\mathbf{z}_i\|\cos(m\theta_{t_i,i})\right)}{\exp\left(\|\mathbf{z}_i\|\cos(m\theta_{t_i,i})\right)+\sum_{j\neq t_i}\exp\left(\|\mathbf{z}_i\|\cos(\theta_{j,i})\right)} \tag{6}$$

where $\theta_{t_i,i}$ has to be in the interval $[0,\pi/m]$, so the cosine argument is in $[0,\pi]$. To avoid this restriction, we need to define a monotonically decreasing function $\phi(\theta_{t_i,i})$ equal to $\cos(m\theta_{t_i,i})$ for $\theta_{t_i,i}\in[0,\pi/m]$ and lower than –1 for $\theta_{t_i,i}>\pi/m$. A function that satisfies this requirement is the pice-wise function $\phi(\theta_{t_i,i})=(-1)^k\cos(m\theta_{t_i,i})-2k$ for $\theta_{t_i,i}\in\left[\frac{k\pi}{m},\frac{(k+1)\pi}{m}\right]$ and $k\in[0,m-1]$. Then, the speaker posterior is finally calculated as

$$P(y_i=t_i|\mathbf{X}_i)=\frac{\exp\left(\|\mathbf{z}_i\|\,\phi(\theta_{t_i,i})\right)}{\exp\left(\|\mathbf{z}_i\|\,\phi(\theta_{t_i,i})\right)+\sum_{j\neq t_i}\exp\left(\|\mathbf{z}_i\|\cos(\theta_{j,i})\right)}\,. \tag{7}$$

The angular margin between classes increases when we set $m>1$, and it is zero when $m=1$. We observed that it is convenient to pre-train the model with softmax cross-entropy before applying A-Softmax objective.

3.3.3. Additive angular margin softmax

Additive angular margin softmax (AAM-Softmax) has been recently proposed for face recognition (Deng et al., 2019). This loss is very similar to A-Softmax. It also enforces $\|\mathbf{W}_j\|=1$ and $b_j=0$ like in A-Softmax but it also normalizes $\|\mathbf{z}_i\|$ by l_2 normalization and re-scale to a constant value s. In this manner the predictions only depend on the angle between the feature and the class weight. It also substitutes the multiplicative angular margin by an additive margin. Thus, the speaker posterior is computed as

$$P(y_i=t_i|\mathbf{X}_i)=\frac{\exp\left(s\cos(\theta_{t_i,i}+m)\right)}{\exp\left(s\cos(\theta_{t_i,i}+m)\right)+\sum_{j\neq t_i}\exp\left(s\cos(\theta_{j,i})\right)}\,. \tag{8}$$

The additive angular margin has better geometric attribute than other losses because the angular margin has the exact correspondence to the geodesic distance.

4. i-Vectors

The i-vector paradigm (Dehak et al., 2011) is an extension of the GMM-UBM approach (Reynolds et al., 2000), where a speech segment is modeled by a Gaussian mixture model (GMM). The i-vector approach assumes that the super-vector mean $\mathbf{M}$ of the segment GMM can be written as

$$\mathbf{M}=\mathbf{m}+\mathbf{T}\boldsymbol{\phi} \tag{9}$$

where $\mathbf{m}$ is the super-vector of UBM means, $\mathbf{T}$ is a low-rank matrix and ϕ is a standard normal distributed vector. $\mathbf{T}$ defines the total variability space, i.e. the directions in which we can move the UBM means to adapt it to a specific segment.

Using this model, we can compute the posterior distribution of ϕ given the segment acoustic features. This posterior is Gaussian distributed, and the mean of this distribution is referred as *i-vector* in the literature. Thus, we can model a variable length sequence with a single feature vector. The i-vector embedding becomes a new feature for pattern classification algorithms like PLDA (Kenny, 2010; Villalba and Brummer, 2011) and PSVM (Cumani et al., 2012).

5. PLDA back-end

Probabilistic linear discriminant analysis (PLDA) has been the state-of-the-art back-end for speaker verification since its introduction (Brümmer and De Villiers, 2010; Kenny, 2010). For this research, we used the simplified PLDA model (SPLDA), which performs as good as the full PLDA. This is a generative model where an *i*/x-vector ϕ_{ij} from the session *j* of the speaker *i* is written as

$$\boldsymbol{\phi}_{ij} = \boldsymbol{\mu} + \mathbf{V}\mathbf{y}_i + \varepsilon_{ij} \tag{10}$$

where μ is a speaker independent term, $\mathbf{V}$ is a low-rank matrix of eigen-voices, $\mathbf{y}_i$ is the speaker factor vector, and ε_{ij} is an offset vector that accounts for the variability between different sessions of the same speaker. The speaker factor contains the speaker information and it is assumed to be standard normal distributed *a priori*. The prior distribution of ε_{ij} is a full covariance Gaussian

$$\varepsilon_{ij} \sim \mathcal{N}(\varepsilon_{ij}|\mathbf{0}, \mathbf{S}_w) \tag{11}$$

where $\mathbf{S}_w$ is the within-class covariance. Meanwhile, the between class covariance can be computed as $\mathbf{S}_b = \mathbf{V}\mathbf{V}^{\mathrm{T}}$.

PLDA is scored by computing the ratio between the likelihood of the trial i/x-vectors given the target hypothesis and the corresponding likelihood given the non-target hypothesis. If the speaker in the enrollment and test i/x-vectors is the same ($\mathcal{M}_1$), both *i*/x-vectors share the same speaker factor $\mathbf{y}$ but have different channel offsets. On the other hand, if they belong to different speakers, they also have different speaker factors. Thus, the ratio is computed as

$$R(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) = \frac{P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathcal{M}_1)}{P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathcal{M}_0)} = \frac{\int P(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2|\mathbf{y}_1)P(\mathbf{y}_1)\ \mathrm{d}\mathbf{y}_1}{\int P(\boldsymbol{\phi}_1|\mathbf{y}_1)P(\mathbf{y}_1)\ \mathrm{d}\mathbf{y}_1 \int P(\boldsymbol{\phi}_2|\mathbf{y}_2)P(\mathbf{y}_2)\ \mathrm{d}\mathbf{y}_2}\ . \tag{12}$$

Note that the speaker identity variables are integrated out. Instead of computing point estimates for $\mathbf{y}$ and comparing the enrollment and test identity variables, we compute the likelihood that both are generated by the same $\mathbf{y}$ regardless of what is the value of $\mathbf{y}$. This method takes into account the uncertainty about the value of $\mathbf{y}$.

The parameters of the PLDA model can be trained by maximum likelihood and minimum divergence iterations. Mathematical derivations of the EM algorithm for different PLDA flavors can be found in Brummer (2010), Villalba (2014), Sizov et al. (2014). Since the PLDA log-likelihood ratio reduces to a quadratic function, discriminative training has also been proposed to estimate the model parameters (Burget et al., 2011; Cumani et al., 2011; 2012).

Usually, it is convenient to pre-process i-vectors by a first linear discriminant dimension reduction, followed by centering, whitening and length normalization (Garcia-Romero and Espy-Wilson, 2011). Initial LDA reduces the number of parameters of the downstream PLDA model. Length normalization helps to alleviate the fact that embedding distribution is heavy tailed. Meanwhile, centering and whitening assure that the normalized *i*/x-vectors are evenly distributed in the unit hyper-sphere.

In recent evaluations, we usually encounter domain mismatch between training and evaluation domain, due to either channel or language. Bayesian adaptation has been proposed to adapt out-of-domain PLDA models to the target domain (Villalba and Lleida, 2012; 2014). In this approach, the out-of-domain model is adopted as prior and limited target-domain data is used to compute the posterior distribution for model parameters. The model parameters that maximize the posterior are the adapted model. In this work, we adopted a simpler approach where between/within-class covariances of the adapted model are a weighted sum of the out-of-domain $\mathbf{S}_{\text{out}}$ and in-domain $\mathbf{S}_{\text{in}}$ covariances (Garcia-Romero et al., 2014),

$$\mathbf{S}_{\text{adapt}} = \alpha\mathbf{S}_{\text{in}} + (1-\alpha)\mathbf{S}_{\text{out}}\ . \tag{13}$$

We can compute the adapted eigen-voice matrix $\mathbf{V}$, from the eigen-decomposition of the adapted between-class covariance.

6. Datasets

6.1. NIST SRE18

The NIST speaker recognition evaluation 2018 (SRE18) is the latest of the series of evaluations conducted by the US National Institute of Standards and Technology (NIST) (Sadjadi et al., 2019). The data for this evaluation comes from two sources: the *Call My Net 2* (CMN2) and *Video Annotation for Speech Technology* (VAST) (Tracey and Strassel, 2018) corpora.

The CMN2 corpus consists of public switched telephone network (PSTN) and voice over IP (VOIP) conversational telephone speech collected in Tunisia. The language for all the calls was Tunisian Arabic. The recruited speakers (called claque speakers) made multiple calls to family and friends using different handsets in a variety of locations (noisy restaurant, office, etc.). This data condition is most challenging, given that most data available to train speaker recognition models consists of English language recorded in the US. For this condition, two sets of development data were provided. The first one, denoted as SRE18-CMN2-dev, consists of 25 speakers ($\sim$ 10 recordings/speaker). It was used to monitor performance and train fusion and calibration. It has 125 enrollment

models and 1566 test segments to build a total of 108,095 trials. The second set, denoted as SRE18-dev-unlabeled, consists of 2332 segments without speaker labels. However, it provides metadata like the speaker's telephone number−though several speakers may use the same phone number. We used this dataset for the PLDA back-end adaptation. Telephone numbers were used as pseudo-speaker labels for a first PLDA adaptation. The adapted PLDA was used to cluster the data and obtain refined speaker labels; which we used for the final PLDA adaptation. The CMN2 evaluation set consists of 188 speakers with 940 models, 12,135 test segments and 2,063,007 trials. Segment durations for this corpus were uniformly sampled between 10 and 60 s.

The VAST corpus consists of amateur video content harvested from the web (Tracey and Strassel, 2018). Given that, a wide range of acoustic conditions may be expected. All videos are in English and its duration ranges from a few seconds to several minutes. Also, videos may contain multiple speakers, so diarization is needed to isolate the target speaker. For the enrollment side, ground truth diarization marks were provided. For this condition, only 10 speakers (2−4 recordings/speaker) were provided for development (SRE18-VAST-dev). With this data, we could create 10 models, 27 test segments and 270 trials. This number of trials is not enough to train calibration because at the operating point of interest ($P_{\mathcal{T}}=0.05$) there were only 2−3 errors. Hence, we decided to calibrate on the SITW eval core-multi, described below, since it also consists of multi-speaker video speech. However, we found that SITW and SRE18-VAST-dev score distributions (mainly the non-target distribution) did not match, resulting in bad VAST calibration. We discovered that adding some VAST data into back-end adaptation and score-normalization we could bring those score distributions closer. We applied diarization to SRE18-VAST-dev to obtain single speaker segments, denoted as SRE18-VAST-dev-diar, and used those for adaptation. The issue of VAST adaptation will be discussed more in detail in Section 9.5. The VAST evaluation set consists of 101 speakers with 101 models, 315 test segments and 31,815 trials.

6.2. Speakers in the Wild

We also experimented on the *Speakers In The Wild* dataset developed by SRI International (McLaren et al., 2016b). As VAST, this dataset consists of audio from video collected in different conditions including real noise, reverberation, intra-speaker variability and compression artifacts. The database consists of recordings from 299 speakers ($\sim$ 8 recordings/speaker), usually public persons. For a given speaker, there may be considerable mismatch in audio conditions between recordings, having speech from high quality studio-based interviews and from raw audio captured on, for example, a camcorder. This dataset is divided into development and evaluation parts. We focused on monitoring performance on the evaluation part and we used the development part for domain adaptation and score normalization. Each part consists of two enrollment conditions:

- Core: enrollment recordings contain a single speaker. Speech varies between 6 and 180 s. For the eval part, it contains 1201 recordings.
- Assist: enrollment recordings contain multiple speakers with a short annotated segment indicating the speaker of interest. For the eval part, it contains 2968 recordings.

Also, it counts with two test conditions:

- Core: test recordings contain a single speaker. Speech varies between 6 and 180 s. For the eval part, it contains 1201 recordings.
- Multi: test recordings contain multiple speakers with a short annotated segment indicating the speaker of interest. For the eval part, it contains 2274. Speech varies between 6 seconds to 10 min.

For this work, we just tested the core vs core (721,788 trials) (denoted just as core) and core vs multi (2,010,683 trials) conditions, since they are comparable to the SRE18 VAST condition. We will not analyze calibration of SITW eval, since it is straightforward, and we will just focus on the usage of SITW eval to calibrate VAST, which is more challenging. For domain adaptation and score normalization, we diarized the multi-speaker signals of the dev part and combined with the single speaker signals (SITW-dev-core), to create the SITW-dev-diar set.

6.3. Training data

The datasets used for training included Switchboard (SWB) phase1-3 and cellular1-2 (Godfrey et al., 1992); NIST SRE04-10 as prepared by the SRE16 Kaldi recipe[3]; NIST SRE12 telephone data (SRE12-tel) and phone-calls recorded through far-field microphone (SRE12-phnmic) (NIST Multimodal Information Group, 2012); Speaker in the Wild dev core (SITW-dev-core)−as explained in previous section−; Mixer6 telephone (MX6-tel) and microphone phone-calls (MX6-phnmic) (Brandschain et al., 2010); and VoxCeleb 1 and 2 (Nagrani et al., 2017; Chung et al., 2018). We did not use interview part of Mixer6 and SRE12 to avoid dealing with removing the interviewer. VoxCeleb 1 and 2 are large scale datasets collected from open-source media. An automatic pipeline obtains YouTube videos from celebrities and confirms the identity of the speaker using face recognition. Some speakers in VoxCeleb overlap with those in SITW so they were removed from the VoxCeleb dataset. In the original VoxCeleb distributions videos are split into short continuous subsegments only containing the voice of the target speaker. We concatenated the

[3] https://github.com/kaldi-asr/kaldi/blob/master/egs/sre16/v2

subsegments belonging to the same original video into a unique segment. Using concatenated VoxCeleb helps to balance the weight of each video in the x-vector training, and avoids including within-session variability in the within-class covariance of the PLDA. We denote the concatenated version of VoxCeleb as VoxCelebCat.

We built 8 kHz and 16 kHz versions of our systems. To train embeddings for 8 kHz, we used all the datasets enumerated above. Databases originally at 16 kHz were downsampled to 8 kHz. Recordings shorter than 4 s and speakers with less than 8 recordings were discarded. In total, we obtained 735,018 utterances from 12,872 speakers. For the 16 kHz systems, we trained embeddings on VoxCelebCat, SITW-dev-core, MX6-phnmic and SRE12-phnmic. In total, this set contained 436,815 recordings from 7936 speakers. To train PLDA for the telephone condition, we used only telephone signals from NIST SRE04-12. This set contained 175,116 recordings from 4585 speakers. To train PLDA for SITW and VAST, we used VoxCelebCat and SITW-dev-core. This set consisted of 418,711 recordings from 7304 speakers.

All datasets were augmented with noise from MUSAN corpus[4] and reverberation using impulse responses from small and medium size rooms in the Aachen impulse response database (AIR)[5]. We added babble (13−20 dB), music (5−15 dB) and generic noises (0−15 dB). We also combined reverberation with additive noise. The number of augmented segments was around twice the size of the original dataset. We combined the original data and the augmented data so that total dataset was around 3 $\times$ the original size.

Other datasets were used for back-end adaptation and score normalization. For adaptation to the video conditions, we used SITW-dev-diar, described in Section 6.2; and SRE18-VAST-dev-diar, described in Section 6.1. SITW-dev-diar was used to center the SITW eval set; and SITW-dev-diar plus SRE18-VAST-dev-diarized, denoted as SITW-SRE18-dev-diar, was used to center SRE18 VAST and for score normalization of the SITW/VAST conditions. We used SRE18-dev-unlabeled, also described in Section 6.1, for centering, PLDA adaptation and score normalization for the Tunisian Arabic condition.

The above setup was designed by people at JHU-CLSP. We focus on this setup since most of our systems were based on it. However, in Sections 9.4 and 10.4, we point out the difference with the setup designed by researchers at JHU-HLTCOE.

7. Experimental setup

7.1. Feature extraction

x-Vectors systems based on time delay networks used 23 MFCC for 8 kHz; and 40 MFCC for 16 kHz. Systems based on ResNets used 23 and 40 log-Mel filter-banks for 8 and 16 kHz respectively. GMM i-vector systems used 23 or 40 MFCC with first and second derivatives. Features were short-time centered before silence removal with a 3 seconds sliding window.

Bottleneck features (BNF) for BNF i-vectors were trained on 1800 hours of Fisher English. The bottleneck network was trained with Kaldi NNet2 (Povey et al., 2011). It consisted of 7 hidden layers, the 6th layer was an 80-dimension linear bottleneck layer; the rest were TDNN layers with p-norm activations and input/output dimension equal to 3500/350. The output layer was a softmax that classifies 5577 senone acoustic units. BNF were used just to compute Gaussian responsibilities but not to compute first order sufficient statistics, where we just used MFCC with deltas, following (McLaren et al., 2016a). Note that, though the BNF network is trained only on Fisher, the i-vector GMM and T matrix are trained on the same data as the rest of i-vector and x-vector systems.

We used the Kaldi energy VAD to remove silence frames. This VAD makes frame-level decisions, classifying a frame as speech or non-speech based on the average log-energy in a given window. We also tested some neural network based VADs, but we did not observe any significant improvement.

7.2. x-Vector networks

x-Vectors based on time-delay networks were implemented using the Kaldi toolkit (Povey et al., 2011). We compared TDNN, extended TDNN and factorized TDNN encoders. The networks were trained using natural gradient descent optimizer with periodic model averaging (Povey et al., 2015). For 16 kHz networks, 3 epochs were enough for convergence. For 8 kHz networks, since 8 kHz data doubled the size of the 16 kHz data, 2 epochs was enough for convergence. x-Vectors based on ResNet34 encoder were implemented in PyTorch (Paszke et al., 2017). They were trained in single GPU using Adam optimizer (Kingma and Ba, 2015) and exponential learning rate decay (Vaswani et al., 2017).

Table 5 summarizes the computational resources for different networks used in our experiments. The architecture for TDNN, E-TDNN and largest F-TDNN are shown in Tables 1−3. For smaller F-TDNN, the network architecture was the same as in Table 3 but with smaller layer sizes, as indicated in the table. ResNet has the lowest number of parameters (8M) and encoder architecture is summarized in Table 4. Number of floating point operations is given for an utterance of 3 s duration, which is the average duration used for training.

7.3. Back-end

Before the PLDA classifier, we pre-processed the speaker embeddings using LDA, centering, whitening and length normalization. We tuned different back-ends for the SITW/VAST condition and the CMN2 condition.

[4] http://www.openslr.org/resources/17
[5] http://www.openslr.org/resources/28

Table 5
Computation resources of different x-vector architectures.

Architecture	TDNN	E-TDNN	F-TDNN	F-TDNN	F-TDNN	F-TDNN	ResNet34
Hidden layer size	512	512	512	600	725	1024	16−128
F-TDNN inner size	–	–	128	150	180	256	–
Num. params. ($\times 10^6$)	8.5	10	9	10	11	17	8
MFLOP (3 sec. utt)	821	1374	790	1046	1463	2798	865

For SITW/VAST, we trained LDA, centering, whitening and SPLDA on VoxCeleb plus SITW-dev-core data, as described in Section 6.3. LDA dimension was 200 and SPLDA had 150 eigenvoices. In the evaluation phase, SITW was centered on SITW-dev-diar. Meanwhile, the centering for the VAST was adapted from SITW-dev-diar to SRE18-VAST-dev-diar by *maximum a posteriori* (MAP) as

$$\boldsymbol{\mu}_{\text{MAP}} = \frac{N_{\text{VAST}}}{N_{\text{VAST}}+r}\boldsymbol{\mu}_{\text{VAST}} + \frac{r}{N_{\text{VAST}}+r}\boldsymbol{\mu}_{\text{SITW}} \tag{14}$$

where μ_{MAP}, μ_{VAST}, μ_{SITW}, are the adapted, VAST and SITW embedding means; N_{VAST} is the number of embeddings in the SRE18-VAST-dev-diar set, and the MAP relevance factor $r = 14$.

We used adaptive S-Norm (AS-Norm) with SITW-SRE18-dev-diar as cohort. For SITW, we selected the 500 top cohort segments. For VAST, we selected the top 120 cohort segments. When applying score normalization on the VAST dev, we have target trials in the cohort score matrices that we do not want to use to compute the normalizing parameters. We assumed that the top 7 segments were target speakers and did not use them to perform the normalization. As there may be several speakers in the test recording, we used diarization to obtain single speaker subsegments from each recording. We scored the enrollment segment against all the test subsegments and selected the maximum score.

For CMN2, we trained LDA, centering, whitening and SPLDA on telephone data from previous evaluations, as described in Section 6.3. For processing CMN2 recordings (unlabeled/dev/eval), we used the centering computed on the SRE18-dev-unlabeled. We also adapted the SPLDA to the SRE18 unlabeled data in two steps. First, we adapted SPLDA using the telephone numbers in the meta-data as speaker labels. Second, we used the adapted SPLDA to compute the affinity matrix between SRE18-dev-unlabeled segments and perform agglomerative hierarchical clustering (AHC) (Reynolds and Torres-Carrasquillo, 2005) to obtain new speakers labels. The number of speakers for AHC was tuned minimizing the SRE18 CMN2 dev Cprimary. Finally, we used the new speaker labels to adapt SPLDA again from the original out-of-domain SPLDA. The within-class and between-class covariances of the adapted model were a weighted sum of the out-of-domain $\mathbf{S}_{\text{out}}$ (weight=0.6) and in-domain $\mathbf{S}_{\text{in}}$ (weight=0.4) covariances. We also used adaptive S-Norm (AS-Norm) using SRE18 unlabeled as cohort. We used the top 400 cohort segments to compute the normalization parameters of each trial.

7.4. Diarization

For diarization of SITW multi and VAST test, we used a similar setup to the Kaldi x-vector CALLHOME diarization recipe[6], which is based on Sell et al. (2018). We used the 16 kHz F-TDNN x-vector, to compute embeddings using a 1.5 s sliding window with 0.75 s of window-shift. We obtained sliding window embeddings for VAST, SITW and VoxCelebCat without augmentation. We used VoxCelebCat x-vectors to train LDA dimensionality reduction to 120, centering and PLDA. We scored all x-vectors in a given recording against each other and applied AHC on the score matrix. We tuned the stopping threshold for AHC to optimize performance on SITW eval core and core-multi sets. We assumed that the target speaker would have a significant amount of speech in the test segment. For that reason, we discarded all the speaker clusters with less than 10 s duration unless all clusters in the segment are shorter than that.

7.5. Evaluation metric

We compare results based on the NIST SRE18 primary metric (Cprimary) (Sadjadi et al., 2019), which is the normalized detection cost function (DCF) (Doddington, 2000). For the SITW/VAST, the DCF operating point was defined by a target prior $P_T = 0.05$. For CMN2, the primary cost was the average of DCF at two operating points, $P_T = 0.01$ and $P_T = 0.005$. We also report equal error rate (EER) results.

8. Fusion and calibration

Fusion and calibration were performed using linear logistic regression with the Bosaris toolkit (Brümmer and De Villiers, 2011). To select the best fusion combination, we implemented a greedy fusion scheme. First, we calibrated all the systems and selected the best one given the lowest actual cost. We fixed the best system and evaluated all the two-systems fusions that

[6] https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v2

Table 6
x-Vector encoder analysis SITW/VAST.

System	SITW EVAL CORE			SITW EVAL CORE-MULTI			SRE18 DEV VAST			SRE18 EVAL VAST		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
16 kHz systems												
BNF-i-vector	5.77	0.257	0.262	6.02	0.260	0.260	11.52	**0.185**	**0.222**	17.46	0.508	0.571
TDNN(8.5M)	3.40	0.185	0.188	3.86	0.191	0.191	**3.70**	0.337	0.424	12.06	0.468	0.578
E-TDNN(10M)	2.74	0.162	0.165	3.20	0.171	0.172	**3.70**	0.305	0.305	13.02	0.442	0.527
F-TDNN(9M)	2.39	0.144	0.150	2.79	0.153	0.153	4.53	0.309	0.383	11.75	0.412	0.508
F-TDNN(10M)	2.37	0.135	0.138	2.86	0.145	0.146	**3.70**	0.337	0.420	**10.79**	0.403	0.503
F-TDNN(11M)	2.05	0.137	0.140	2.57	0.145	0.147	**3.70**	0.305	0.387	11.11	0.409	0.487
F-TDNN(17M)	**1.89**	**0.124**	**0.126**	**2.33**	**0.135**	**0.137**	7.00	0.370	0.498	12.06	**0.388**	**0.474**
ResNet(8M)	3.01	0.187	0.191	3.47	0.198	0.198	**3.70**	0.412	0.498	11.43	0.464	0.554
8 kHz systems												
GMM-i-vector	8.22	0.384	0.393	8.67	0.386	0.387	18.52	0.486	0.568	20.32	0.543	0.750
BNF-i-vector	7.80	0.353	0.365	8.42	0.352	0.354	14.81	0.412	0.568	17.90	0.533	0.638
TDNN(8.5M)-sre16	5.21	0.278	0.284	5.60	0.287	0.287	11.11	0.300	0.691	13.33	0.475	0.636
TDNN(8.5M)	3.58	0.197	0.202	3.93	0.206	0.207	7.41	0.296	0.535	12.93	0.431	0.596
E-TDNN(10M)	2.90	0.172	0.175	3.29	0.183	0.183	7.41	0.337	0.461	12.60	0.410	0.561
F-TDNN(11M)	2.84	0.158	0.163	3.18	0.165	0.166	7.41	**0.222**	0.461	12.06	0.385	0.52
F-TDNN(17M)	**2.46**	**0.148**	**0.151**	**2.83**	**0.155**	**0.156**	**4.53**	0.259	**0.383**	**11.75**	**0.377**	**0.514**

include the best system. Thus, we selected the best fusion of two systems. We fixed those two system and then added a third system, and so on. To reduce the chances of over-fitting, in each step, we prioritized fusions with only positive weights.

For VAST, we trained fusion/calibration on SITW eval-core multi on operating point $P_T = 0.05$. For CMN2, we trained on SRE18 dev CMN2 set on operating point $P_T = 0.01$. Although the average operating point for CMN2 is $P_T = 0.075$, we decided to use a higher target prior to have more false alarm errors and obtain a more robust calibration.

9. Results and discussion for SITW/VAST

9.1. x-Vector encoder

We analyze results for different encoder architectures in the x-vector network on the SITW and SRE18 VAST datasets. We also compare them with GMM-UBM and BNF i-vectors. Table 6 compares the results in terms of Cprimary and EER. The table is divided into two blocks. The upper block shows results for systems developed at 16 kHz sampling frequency while the lower block shows results for 8 kHz systems. We compare x-vectors with TDNN, extended TDNN and factorized TDNN; and 2D convolutional ResNet. We compare several F-TDNN versions with different hidden layer sizes, as summarized in Table 5. We include the number of parameters of each network in parenthesis. All networks used mean plus standard deviation pooling and were trained with softmax cross-entropy objective. All results are with automatic diarization as described in 7.4, using F-TDNN(17M) to extract diarization x-vectors. Thus, all networks were compared using the same diarization references.

First, we focus on the SITW results. For both 8 and 16 kHz, x-vectors were significantly better than i-vectors. For 8 kHz, we compared GMM-UBM and BNF i-vectors. BNF i-vectors min. Cprimary was just 4% better than GMM-UBM i-vectors. We think that BNF did not improve more because of domain mismatch between video speech and BNF training data (telephone). For 16 kHz, we just ran BNF i-vectors since there was not much difference with GMM i-vectors. For 16 kHz systems, the small TDNN (8.5M) had min. Cprimary 26% relative better than i-vectors. Meanwhile, the largest F-TDNN(17M) improved by 50% w.r.t. i-vectors. We experimented with F-TDNN with reduced number of parameters to compare with E-TDNN. For similar number of parameters (9-10M), F-TDNN performed 10−15% better than E-TDNN. We think that reduction in number of parameters per layer obtained by the weight matrix factorization and the skip connections helped to train the network better. Furthermore, F-TDNN(10M) only degraded min. Cprimary by 7% w.r.t. F-TDNN(17M). Original TDNN x-vector performed worse than E-TDNN and F-TDNN. ResNet is the model with the lowest number of parameter and performed similar to the original TDNN. However, ResNet is not fully comparable with the rest given that it was trained using the Pytorch setup, which does not have the usual Kaldi tricks like model averaging. x-Vectors at 16 kHz performed better than x-vectors at 8 kHz indicating that the networks were able to take advantage of the speaker discriminant information present above 4 kHz. However, 8 kHz x-vectors were still competitive. F-TDNN(17M) 8 kHz was only 13% worse than its 16 kHz equivalent. The comparison between 8 kHz encoders followed a trend similar to the 16 kHz systems. F-TDNN(17M) had 6% better min. Cprimary than F-TDNN(11M); F-TDNN(11M) improved by 10% w.r.t. E-TDNN. We also compared the small training setup in SRE16 Kaldi recipe[7] with our new setup using TDNN network. The SRE16 setup did not include VoxCeleb or SRE12 data. The new setup provided about 30% of improvement.

Note that the results for the core-multi condition are close to ones in the core-core condition. This means that diarization worked really well on this data. We will analyze diarization in more detail in Section 9.3.

[7] https://github.com/kaldi-asr/kaldi/blob/master/egs/sre16/v2

Now, we focus on the VAST results for 16 kHz systems. The first observation is that the performance in VAST dev is not correlated with the performance in eval. This is because, we could create only 270 trials with the dev data, which gave just 2−3 errors in the target operating point. We also observe that performance in VAST eval is much worse than in SITW core-multi. EER is about 5 times worse and Cprimary is about 3 times worse. We think that this is because VAST presents much more challenging acoustic conditions. While VoxCeleb and SITW are based on audio from celebrities, which in many occasions will be recorded by professionals, VAST mainly consists of amateur content. Doing some random listening of VAST audios, we noted children crying, strong impulse noises, recordings in open crowded spaces, etc. This mismatch between the training and VAST data derives in smaller differences between encoder architectures. The best x-vector was still much better than i-vector−31% relative in terms of EER, 24% better in terms of Cprimary-but relative improvement is smaller than for SITW. Min. Cprimary of F-TDNN(17M) was 17% better than for TDNN and 12% better than E-TDNN. Performance differences between F-TDNN sizes were less than 4%, not significant in our opinion. For 8 kHz systems, the performance difference between encoders is even smaller than for 16 kHz. Min. Cprimary for F-TDNN(17M) was only 12% better than TDNN, and 8% better than for E-TDNN.

The conclusion from these experiments is that, overall, F-TDNN was the best encoder architecture for our setup. Version with 10 million parameters can be a good choice balancing computational cost and performance. x-Vectors clearly surpassed i-vectors for all conditions and encoder architectures. The results in VAST eval indicate that x-vectors with large number of parameters over-fit to the training domain, where they get superior performance, but they perform just slightly better than smaller networks in mismatched data.

9.2. x-Vector pooling and training objective

In this section, we focus on comparing pooling methods and training objectives using the ResNet encoder architecture. Table 7 shows the results. ResNet was not competitive w.r.t. other encoders in previous section (using just mean+stddev pooling). However, using other pooling methods, it can improve and reach performance similar to that of the Kaldi F-TDNN. We compared mean plus standard deviation pooling (mean+std), learnable dictionary encoder (LDE) and multi-head attention (MHAtt). LDE/MHAtt used 64 clusters/heads, decided empirically. In Cai et al. (2018b), mean+stddev pooling is compared to LDE64. The output of the ResNet has 128 channels, so when we apply mean+stddev pooling we obtain a 2×128 dimensional vector while LDE provides a 64×128 dimensional super-vector. We argue that it seems fairer to compare pooling super-vectors of the same size. Thus, we added a linear layer projecting the ResNet output from 128 to 4096 to get a pooling super-vector of the same size as LDE (8192). For SITW, mean+stddev pooling with large pooling vector significantly outperformed the original version with small pooling vector. We think that, in the same way that support vector machines use the kernel trick to project vectors into high dimensional space and make them linearly separable, by projecting the ResNet representations to high dimension they become more adequate for the mean+stddev pooling. However, we still obtained the best results by using LDE, which clearly suggests that ResNet frame-level representations are multi-modal. Min. Cprimary for LDE was about 25% better than mean+stddev with 4096 dimension. On the other hand, multi-head attention performed poorly compared to LDE and just slightly better than mean +stddev with dimension 128. For VAST, LDE also performed significantly better than the others, about 12%. Also, it matched the performance of Kaldi F-TDNN(17M) shown in the previous section.

Regarding training objectives, in SITW, A-Softmax and AAM-Softmax improved w.r.t. softmax cross-entropy. However, in VAST, the three losses obtained similar Cprimary. This suggests that domain mismatch diminishes the advantages of the angular margin criteria. We also provide the result with LDE with 32 components, which we submitted for the actual evaluation. Using 64 component was better, as shown in Cai et al. (2018b).

9.3. Diarization effect

Table 8 compares results with and without speaker diarization. Diarization was effective in SITW. In the core condition, where there is only one speaker in enrollment and test recordings, it did not worsen the results. In the core-multi condition, it improved EER and Cprimary by about 20%. However for VAST, the difference between both was not significant. Considering that VAST is

Table 7
Analysis of x-vector pooling and training objective on SITW/VAST. All systems are 16 kHz.

System	SITW EVAL CORE			SITW EVAL CORE-MULTI			SRE18 DEV VAST			SRE18 EVAL VAST		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
Softmax loss												
ResNet-mean+std-128	3.67	0.231	0.235	4.2	0.238	0.239	**3.7**	0.374	0.576	13.02	0.494	0.676
ResNet-mean+std-4096	3.01	0.187	0.191	3.47	0.198	0.198	**3.7**	0.412	0.498	11.43	0.464	0.554
ResNet-MHAtt64	3.91	0.228	0.234	4.37	0.241	0.242	**3.7**	0.300	0.346	12.7	0.465	0.595
ResNet-LDE64	**2.50**	**0.146**	**0.15**	**2.82**	**0.154**	**0.154**	**3.7**	**0.193**	**0.230**	**9.60**	**0.408**	**0.498**
A-softmax loss												
ResNet-LDE32	**2.16**	0.136	0.142	2.63	0.145	0.146	**3.7**	0.226	0.424	10.79	0.412	0.516
ResNet-LDE64	2.21	**0.122**	**0.122**	**2.57**	**0.131**	**0.132**	**3.7**	**0.111**	0.424	9.96	0.417	**0.479**
AAM-softmax												
ResNet-LDE64	**2.11**	0.130	0.136	**2.57**	0.143	0.144	3.7	0.305	0.461	**9.28**	**0.399**	**0.470**

Table 8
Diarization effect on SITW/VAST.

System	SITW EVAL CORE			SITW EVAL CORE-MULTI			SRE18 EVAL VAST		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
F-TDNN(17M) w/o diar	**1.89**	**0.121**	0.127	2.99	0.169	0.169	**11.28**	0.404	0.512
F-TDNN(17M) with diar	**1.89**	0.124	**0.126**	**2.33**	**0.135**	**0.137**	12.06	**0.388**	**0.474**

also a multi-speaker scenario, we expected to obtain a significant improvement from the diarization. We have tried to tune diarization hyper-parameters for VAST but so far, we haven't observed any improvement. We conclude, that domain mismatch also broke diarization performance. This is an issue that deserves further investigation.

9.4. Comparison with HLTCOE setup

So far, we have done our analysis based on the JHU-CLSP training setup, since most of our systems were based on it. In this section, we compare JHU-CLSP and JHU-HLTCOE setup using common E-TDNN Kaldi x-vector architecture. Table 9 summarizes the differences between both setups. Table 10 presents results for three systems: E-TDNN trained with HLTCOE setup using HLTCOE back-end, same E-TDNN using JHU-CLSP back-end; and E-TDNN trained with JHU-CLSP setup using JHU-CLSP back-end. We observe minor differences between back-ends, since both were generative PLDA trained mostly on VoxCeleb. However, the performance of the HLTCOE E-TDNN was significantly better. E-TDNN-HLTCOE was about 20% better than E-TDNN-CLSP in SITW and about 10% in VAST eval. E-TDNN-HLTCOE performed also similar to larger F-TDNN. There are some differences between both setups that could explain this (Snyder et al., 2019). The most important are: more aggressive data augmentation multiplying training size $\times 6$ instead of $\times 3$; larger batch size of 128 segments, instead of 64; and 6 training epochs instead of 3. We think that tuning the training setup in this way, we could also obtain some gains in other E-TDNN and F-TDNN systems. However, given that this is a much larger setup, doing that for all our systems is not feasible because of the computing cost.

9.5. Calibration

9.5.1. Eval calibration

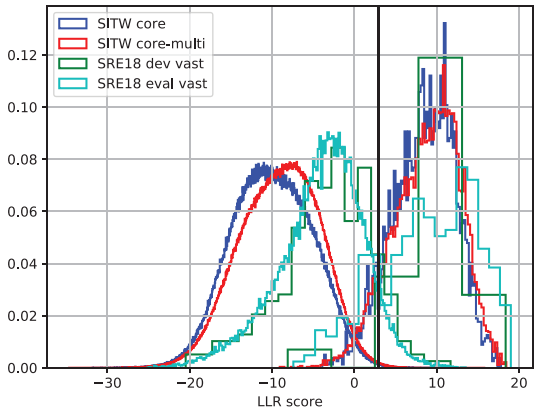
In the first stages of our research for SRE18, we assumed that VAST condition would be similar to SITW. Thus, we did not include any VAST data for domain adaptation. We did not do any score normalization since it did not improve performance on SITW. With that setup, we obtained actual Cprimary larger than 1 on VAST dev. This result led us to compare the score distributions of SITW and VAST dev. Fig. 3a shows those distributions. SITW core and core-multi are depicted in red and blue; and VAST dev in green. The black vertical line marks the minimum risk Bayes decision threshold. By comparing green and red lines, we decided that SITW and VAST target score distributions were more or less well aligned. However, the VAST dev non-target distribution was shifted to the right w.r.t. SITW non-targets. As consequence, we obtained a huge false alarm rate for VAST dev. Note

Table 9
Comparison CLSP vs HLTCOE setup configurations.

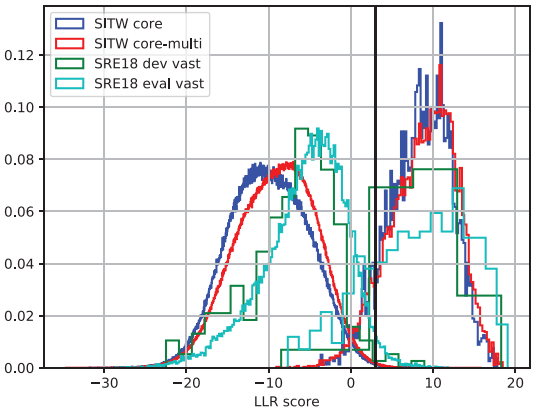
		Training Data						Augmentations						Aug. Size	Epochs	PLDA
		SWB	SRE04-10	SRE12	MX6	VoxCeleb	SITW -dev	Musan Noise	Musan Music	Musan Babble	MX6 Babble	Codecs	Reverb			
16 kHz	CLSP			✓	✓	✓	✓	✓	✓	✓			✓	3	3	SPLDA
	HLTCOE					✓		✓	✓		✓	✓	✓	6	6	Full-rank PLDA
8 kHz	CLSP	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	3	3	SPLDA
	HLTCOE	✓	✓			✓		✓	✓		✓	✓	✓	6	6	Heavy-tailed PLDA

Table 10
Comparison CLSP vs HLTCOE setup on SITW/VAST.

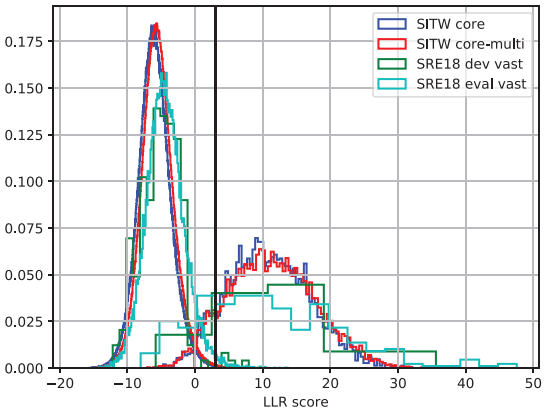
System	SITW EVAL CORE			SITW EVAL CORE-MULTI			SRE18 DEV VAST			SRE18 EVAL VAST		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
HLTCOE back-end												
E-TDNN-HLTCOE	**1.99**	0.138	0.141	**2.26**	0.135	0.137	**3.7**	0.337	0.498	**11.11**	0.402	**0.452**
CLSP back-end												
E-TDNN-HLTCOE	2.21	**0.125**	**0.129**	2.54	**0.130**	**0.132**	**3.7**	**0.296**	0.309	**11.11**	**0.398**	0.475
E-TDNN-CLSP	2.74	0.162	0.165	3.20	0.171	0.172	**3.7**	0.305	**0.305**	13.02	0.442	0.527
F-TDNN(17M)	**1.89**	**0.124**	**0.126**	**2.33**	0.135	0.137	7.0	0.370	0.498	12.06	**0.388**	**0.474**



(a) Centering with SITW, without score normalization.



(b) Centering with SITW+VAST, without score normalization.



(c) Centering with SITW+VAST, with score normalization.

Fig. 3. Score histograms for SITW and VAST for different back-ends.

that the false alarm rate is the area under the non-target distribution integrated from the threshold to infinity. Thus, while false alarms for SITW have small contribution to the Cprimary calculation, for VAST, false alarms have the main contribution to the cost, causing miscalibration.

We decided to use VAST dev data for adaptation to improve calibration. Since VAST dev only contains 10 speakers, we decided that it was not large enough to perform PLDA adaptation. We decided to use it only on the centering step, previous to length normalization. We adapted the SITW centering to VAST using *maximum a posteriori*. We used a relevance factor $r=14$, which we tuned by visual inspection of the VAST dev and SITW score distributions. With this procedure, we obtained the score distributions in Fig. 3b. We still observe some distribution shift, but the false alarm rate has been greatly reduced.

In the next step, we decided to add adaptive S-Norm (AS-Norm) score normalization with a pool of SITW-dev and VAST-dev cohort segments. The number of top cohort segments selected by the adaptive method was tuned to 500 for normalizing SITW eval and 120 for VAST dev/eval. This tuning was needed because of the unbalance between the number of cohort segments from SITW and VAST. Restricting the top cohort segments for VAST reduces the weight of SITW segments in the normalization. With this method, we obtained the score distributions in Fig. 3c. Now, the score distributions of SITW and VAST dev are better aligned; and the false alarm rate is much smaller.

After applying these steps, we calibrated in the usual way using the SITW eval core-multi scores.

At evaluation time, we were not sure whether this calibration method would be effective, since we did not know the VAST eval distribution. Now, having the keys, we can also add the VAST eval distributions to the analysis (cyan line). We can observe that, although the VAST dev was a very small set, the score distribution was very similar to the VAST eval. Thanks to that, this method was effective, and we could obtain good performance in the evaluation. All the systems shown in Tables 6–10, were calibrated with this method.

In Fig. 3, we do not show the case with AS-Norm and without centering adapted to VAST because it is not visually distinguishable from Fig. 3c. We concluded that AS-Norm compensates for most of the domain mismatch. However, having adaptation in the centering still gave some additional improvement.

9.5.2. Post-eval calibration

As we have seen in Fig. 3, the score distributions of VAST dev and eval are well aligned. Based on that, at post-eval, we tried to improve calibration by adapting the scores of SITW to the SRE18 VAST dev. The idea consists in making the SITW target and non-target score distributions overlap with the SRE18 VAST dev score distributions. To do so, we first assumed that the score distributions are approximately Gaussian. That is, we assumed that they can be described by their means and variances. Following, we adapted the parameters of the SITW score distributions to VAST using *maximum a posteriori* (MAP). Separately for targets and non-targets, we obtain MAP adapted means and variances by

$$\mu_{\text{MAP}} = \frac{N_{\text{VAST}}}{N_{\text{VAST}}+r}\mu_{\text{VAST}} + \frac{r}{N_{\text{VAST}}+r}\mu_{\text{SITW}} \tag{15}$$

$$\sigma^2_{\text{MAP}} = \frac{N_{\text{VAST}}}{N_{\text{VAST}}+r}\sigma^2_{\text{VAST}} + \frac{r}{N_{\text{VAST}}+r}\sigma^2_{\text{SITW}} + \frac{rN_{\text{VAST}}}{(N_{\text{VAST}}+r)^2}(\mu_{\text{VAST}}-\mu_{\text{SITW}})^2 \tag{16}$$

where μ_{SITW}, and σ^2_{SITW} are the maximum likelihood mean and variance of the SITW distribution (prior distribution); μ_{VAST} and σ^2_{VAST} are the mean and variance of the SRE18 VAST score distribution (adaptation data); μ_{MAP}, σ^2_{MAP} are the adapted mean and variance (posterior distribution); N_{VAST} is the number of VAST dev trials; and r is the MAP relevance factor. To adapt the target distribution, we tuned $r=12$ for targets and $r=100$ for non-targets. Next, we transformed the SITW scores s_{SITW} with

$$s_{\text{MAP}} = \frac{\sigma_{\text{MAP}}}{\sigma_{\text{SITW}}}(s_{\text{SITW}}-\mu_{\text{SITW}}) + \mu_{\text{VAST}} \tag{17}$$

to make the SITW distribution to have the same mean and variance as the adapted score distribution. Finally, we used the adapted SITW scores to train the calibration by linear logistic regression.

Table 11 compares the calibration method that we used in the evaluation phase with this new method. The *Act Cp eval* column shows actual cost using the pre-eval calibration method, using VAST data for centering and AS-Norm as described in Section 9.5.1. The *Act Cp post-eval* column shows actual cost combining the eval and post-eval calibration methods. That is, we applied AS-Norm to scores, adapted SITW scores to VAST, and trained calibration on adapted SITW scores. For the JHU-MIT primary fusion and F-TDNN, calibration improved around 15%; for JHU-HLTCOE E-TDNN, it improved by 10%. We also wanted to know whether the new calibration method allows us to remove AS-Norm. In the last line of the table, we include a F-TDNN(17M) system without AS-Norm and with/without new calibration method. We observe that by removing AS-Norm in old calibration method, the eval actual cost is 26% worse than using AS-Norm. By adding the new calibration method, it improved by 26%, but it was still 14% worse than combining AS-Norm with the new calibration method. The conclusion is that we need to combine all the above techniques to achieve the optimum performance.

9.6. Fusion

Table 13 presents fusion results. All the systems were calibrated with the post-eval method explained in Section 9.5.2. The first line shows the JHU-MIT primary system submitted to NIST SRE18. The rest of the table shows fusions of post-eval systems based only on the JHU-CLSP training setup. These fusions are the best single system, and the best fusion of 2, 3 and four systems. We observe a significant improvement by fusing two systems. However, the gain obtained by fusing more than two systems was minimal. Table 12 lists the systems included in each fusion. The greedy fusion scheme selected ResNets with LDE and A-softmax

Table 11
Calibration Analysis for SITW/VAST. All systems used AS-Norm except indicated otherwise.

System	SRE18 EVAL VAST			
	EER	Min Cp	Act Cp eval	Act Cp post-eval
Primary JHU-MIT	10.16	0.357	0.431	0.369
E-TDNN-HLTCOE	11.11	0.402	0.452	0.409
F-TDNN(17M)	12.06	0.388	0.474	0.402
F-TDNN(17M) w/o AS-Norm	11.49	0.426	0.645	0.471

Table 12
Submission system fusion summary.

Submission	VAST	CMN2
JHU-MIT Primary	F-TDNN(17M)-16k + F-TDNN(17M)-8k + E-TDNN-HLTCOE-16k + ResNet-LDE32-ASoftmax-16k	E-TDNN(10M)-HLTCOE-8k + ResNet-MHAtt-ASoftmax-DPLDA + TDNN(8.5M)-8k
Best 1	ResNet-LDE64-ASoftmax-16k	ResNet-MHAtt-ASoftmax-GPLDA-8k
Best 2	+ F-TDNN(17M)-8k	+ F-TDNN(17M)-8k
Best 3	+ F-TDNN(17M)-16k	+ TDNN-8k
Best 4	+ ResNet-LDE32-ASoftmax-16k	+ BNF-i-vector-8k

Table 13
Fusion results on SITW/VAST.

Submission	SITW EVAL CORE			SITW EVAL CORE-MULTI			SRE18 DEV VAST			SRE18 EVAL VAST		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
JHU-MIT Primary	**1.53**	**0.097**	**0.098**	**1.82**	**0.105**	**0.105**	**3.7**	**0.305**	0.465	10.18	**0.358**	**0.369**
Best 1	2.21	0.122	0.122	2.57	0.131	0.132	**3.7**	**0.111**	0.424	**9.96**	0.417	0.455
Best 2	1.67	0.101	0.106	2.02	0.112	0.113	**3.7**	0.259	**0.346**	10.05	0.37	0.405
Best 3	1.61	**0.097**	0.100	1.93	**0.107**	**0.107**	**3.7**	0.259	0.498	10.48	0.369	**0.394**
Best 4	**1.59**	0.098	**0.099**	**1.90**	**0.107**	**0.107**	**3.7**	0.296	0.461	10.48	**0.368**	0.396

loss and large F-TDNN systems. Interestingly, for the two-systems fusion, it didn't choose the second-best single system but the F-TDNN at 8 kHz. This confirms that fusion of more heterogeneous systems provides a larger improvement.

10. Results and discussion for CMN2

10.1. Individual systems

Now, we analyze the *Call My Net 2* (CMN2) condition in Tunisian Arabic language. Table 14 presents results for several i-vector and x-vector systems. All systems in the table, unless indicated otherwise, are 8 kHz, used mean+stddev pooling and generative simplified PLDA back-end. They also include PLDA adaptation and AS-Norm. BNF and GMM-UBM performed similarly. As expected, BNFs for English were not useful for Tunisian. All the embedding systems performed better than i-vectors even though all were trained mainly in English and neural networks are usually prone to over-fitting. The smaller TDNN(8M) improved actual Cprimary by 27% w.r.t. i-vectors; the largest F-TDNN improved by 50%. F-TDNN(11M) was only 6% worse than F-TDNN(17M). E-TDNN performed slightly worse than F-TDNN having similar number of parameters. Also, ResNet with multi-head attention pooling performed similar to E-TDNN.

We compared the new training setup with previous smaller setup in Kaldi SRE16 recipe. New setup moderately improved actual Cprimary by 13%. Also, we compared generative and discriminative PLDA in ResNet with multi-head attention. While DPLDA was the best in dev, it over-fitted and performed poorly in the eval.

10.2. Analysis of adaptation methods

Table 15 compares back-end adaptation methods on CMN2. It compares centering adaptation, PLDA and adaptive S-Norm (AS-Norm). For this analysis, we used the x-vector system with F-TDNN of 17M parameters. We obtained the best result by combining the three methods. Thus, we improved CMN2 eval EER and Cprimary by more than 30%. Centering by itself had a small

Table 14
Individual system results on CMN2. All systems are 8 kHz.

Systems	SRE18 DEV CMN2			SRE18 EVAL CMN2		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
GMM-i-vector	10.37	0.664	0.685	11.85	0.723	0.725
BNF-i-vector	10.51	0.639	0.657	11.69	0.710	0.712
TDNN(8.5M)-sre16	7.20	0.505	0.51	7.93	0.515	0.518
TDNN(8.5M)	5.76	0.384	0.392	6.68	0.446	0.447
E-TDNN(10M)	5.88	0.392	0.398	5.97	0.409	0.410
F-TDNN(11M)	**4.96**	**0.326**	**0.33**	5.30	0.370	0.371
F-TDNN(17M)	5.10	0.355	0.372	**4.95**	**0.346**	**0.349**
ResNet(8M)-MHAtt-SPLDA	5.46	**0.326**	**0.34**	5.64	0.392	0.395
ResNet(8M)-MHAtt-DPLDA	5.64	**0.319**	**0.337**	6.81	0.499	0.524

Table 15
Comparison of adaptation methods in CMN2 with x-vector based on F-TDNN(17M).

Adaptation	SRE18 DEV CMN2			SRE18 EVAL CMN2		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
No adapt	7.74	0.488	0.502	7.37	0.492	0.509
AS-Norm	6.25	0.404	0.410	05.97	0.385	0.387
Centering	7.37	0.477	0.488	6.80	0.473	0.482
Centering + AS-Norm	6.13	0.410	0.421	5.67	0.386	0.389
Centering + PLDA	5.38	0.386	0.399	5.1	0.404	0.409
Centering + PLDA + AS-Norm	**5.1**	**0.355**	**0.372**	**4.95**	**0.346**	**0.349**

Table 16
Results for CMN2 sub-conditions for x-vector based on F-TDNN(17M).

Systems	SRE18 DEV CMN2			SRE18 EVAL CMN2		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
Total	5.10	0.355	0.372	4.95	0.346	0.349
Male	6.81	0.391	0.443	5.15	**0.345**	**0.347**
Female	**3.70**	**0.292**	**0.301**	**4.74**	0.346	0.352
PSTN	**4.38**	**0.356**	**0.398**	**4.72**	**0.310**	**0.314**
VOIP	6.49	0.349	0.401	5.32	0.408	0.413
Same phone	**2.94**	**0.289**	**0.306**	**3.42**	**0.266**	**0.268**
Diff. phone	6.29	0.387	0.404	5.64	0.386	0.390
1 Enrol. segment	7.35	0.421	0.434	6.07	0.434	0.436
3 Enrol. segments	**2.77**	**0.275**	**0.309**	**3.78**	**0.257**	**0.262**

impact, 7% for EER and 5% for Cprimary. When we put adaptive PLDA or AS-Norm on top of centering, we obtained that both had similar improvement in terms of Cprimary−15-19% w.r.t. just centering. However, adaptive PLDA improved EER more than AS-Norm−25% versus 16%−. AS-Norm improved Cprimary by another 15% w.r.t. centering plus PLDA adaptation.

10.3. Analysis of CMN2 subconditions

CMN2 trial list can be split in several sub-conditions depending on gender, telephone channel, telephone handset used in enrollment and test, and number of enrollment segments. Table 16 breaks the results by subconditions. All our systems followed similar trends, so we just show results with F-TDNN(17M) x-vector. We consider the results in CMN2 eval more reliable than dev, since it has more trials per subcondition. We observe that male and female trials obtained similar performance. Public switched telephone network (PSTN) performed better than Voice over IP (VoIP), since we didn't have VoIP data in our training. Cprimary degraded by 24% because of this. Also, there was a significant degradation when target speakers used different telephone handset in enrollment and test. In this case, Cprimary degraded by 30%, which means that there is still margin for better channel compensation. Finally, having 3 enrollment utterances was about 40% better than just one in EER and Cprimary. There is still work to do to reduce this margin.

10.4. Comparison with HLTCOE setup

Again, we compare JHU-CLSP versus JHU-HLTCOE neural network training setup for CMN2, using common E-TDNN Kaldi x-vector architecture. Table 9 summarizes the differences between both setups. Table 17 presents results for three systems: E-TDNN trained with HLTCOE setup using HLTCOE back-end, same E-TDNN using JHU-CLSP back-end; and E-TDNN trained with JHU-CLSP setup using JHU-CLSP back-end. There is very small difference between back-ends. HLTCOE back-end Cprimary was less than 3% better than CLSP back-end. The main difference between both is that HLTCOE used heavy-tail PLDA (Silnova et al., 2018) without length normalization, instead of Gaussian SPLDA with length normalization. The difference between E-TDNN training setup is more significant. HLTCOE E-TDNN was 11% better than the CLSP one in terms of EER and actual cost. The same reasons exposed in Section 9.4−more aggressive augmentation, different bath-size/epochs− could explain this difference. Furthermore, HLTCOE augmented VoxCeleb with GSM telephone channel, which would adapt VoxCeleb better to the telephone condition.

10.5. Fusion

Table 18 presents fusion results for CMN2. Table 12 lists the systems included in each fusion. The first line shows the JHU-MIT primary system submitted to NIST SRE18. The rest of the table shows fusions of post-eval systems based only on the JHU-CLSP

Table 17
Comparison CLSP vs HLTCOE setup on CMN2.

Systems	SRE18 DEV CMN2			SRE18 EVAL CMN2		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
HLTCOE back-end						
E-TDNN-HLTCOE	**4.55**	**0.298**	**0.312**	**4.95**	**0.352**	**0.354**
CLSP back-end						
E-TDNN-HLTCOE	5.02	0.339	0.346	5.26	0.360	0.362
E-TDNN-CLSP	5.88	0.392	0.398	5.97	0.409	0.410

Table 18
Fusions on CMN2.

Systems	SRE18 DEV CMN2			SRE18 EVAL CMN2		
	EER	Min Cp	Act Cp	EER	Min Cp	Act Cp
JHU-MIT Primary	**4.09**	**0.249**	**0.256**	4.50	**0.312**	**0.313**
Best 1	5.46	0.326	0.34	5.64	0.392	0.395
Best 2	4.68	0.294	0.303	**4.57**	**0.316**	**0.318**
Best 3	4.63	**0.29**	**0.297**	4.60	0.318	0.319
Best 4	**4.58**	0.292	0.301	4.62	0.318	0.32

training setup. These fusions are the best single system, and the best fusion of 2, 3 and four systems. We didn't obtain any gain from fusing more than two systems. As for video, we obtained the best results fusing large F-TDNN and ResNet.

11. Conclusions

We analyzed in detail the systems developed by the JHU-MIT consortium in the context of NIST SRE18. Compared to previous NIST SRE16, where i-vectors were still the speaker recognition state-of-the-art, neural network embeddings (a.k.a. x-vectors) rise now as the new best performing approach. In some conditions, x-vectors with very large networks improved the detection cost of i-vectors by a factor of 2. We compared different encoder network architectures for x-vectors in SITW, SRE18 VAST and CMN2 conditions. For the JHU-CLSP training setup, factorized TDNN performed the best compared to other options with similar number of parameters for the three datasets evaluated. However, E-TDNN performed as good as F-TDNN using the HLTCOE setup, which included some differences like more aggressive augmentation and larger batch-size. These F/E-TDNN networks achieved performance without parallel on SITW and CMN2 data. Despite that SITW and VAST consist both of speech from video, VAST obtained 3 times worse Cprimary. We think that this is because VAST contains more amateur video content compared to SITW and VoxCeleb, which mainly consist of celebrity videos. In VAST, recording conditions may be more neglected, and equipment may be of lower quality. This translated into strong domain mismatch between SITW/VoxCeleb and VAST. Domain mismatch made x-vectors with large number of parameters to over-fit, performing just slightly better than smaller networks on the mismatched VAST data.

We also compared pooling methods. Learnable dictionary encoder performed the best. This suggests that representations learned by the x-vector encoder are multi-modal. We also compared softmax cross-entropy with angular softmax and additive angular margin softmax training criteria. Both angular margin criteria were better when there was no mismatch between training and evaluation data. However, in the VAST mismatched condition they lost its advantage and performed similar to softmax cross-entropy.

Domain mismatch caused that the score distributions of SITW and VAST were not well aligned. We needed those distributions to be aligned to be able to calibrate VAST using SITW. We managed to align both score distributions by using the limited VAST dev data in the centering and score-normalization steps. Later, we improved this by MAP adapting the SITW scores to VAST and calibrating on adapted SITW scores. We also observed that domain mismatch can break automatic diarization. In SITW, diarization improved significantly while in VAST, it did not.

We analyzed back-end adaptation methods in CMN2. PLDA semi-supervised adaptation and AS-Norm yielded significant improvements. However, results are still worse than for English in-domain conditions like SITW.

We can say that, to achieve optimal performance, the *know how* accumulated from previous evaluations played an important role for proper back-end adaptation and score calibration.

System fusions obtained improvements w.r.t. single systems, although most of the gain came from the fusion of just two competitive systems. We conclude that, with x-vector systems, mega-fusions are not required.

As final remark, x-vectors provide the new state-of-the-art performance in speaker recognition evaluations. However, their advantages reduce in cases of strong domain mismatch. Investigation about domain adaptation and domain invariant approaches are needed to be able to use x-vectors in any condition. Also, speech enhancement techniques with a focus on improving the speaker recognition objective, instead of audio quality, can be of great help. Finally, improving diarization is required for the multi-speaker scenario. In this area, there are still open problems like selecting the optimum number of speaker clusters, extracting more accurate embeddings from short duration sequences, where phonetic information dominates over the speaker information, and overlap detection.

Acknowledgment

This work is sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

Brandschain, L., Graff, D., Cieri, C., Walker, K., Caruso, C., 2010. The mixer 6 corpus: resources for cross-channel and text independent speaker recognition. In: Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC10, Valletta, Malta, pp. 2441–2444.

Brummer, N., 2010. EM for probabilistic LDA. Technical Report February. Agnitio Research, Cape Town, South Africa.https://sites.google.com/site/nikobrummer/EMforPLDA.pdf
Brümmer, N., De Villiers, E., 2010. The speaker partitioning problem. In: Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop. ISCA, Brno, Czech Republic, pp. 194–201.http://www.isca-speech.org/archive_open/archive_papers/odyssey_2010/papers/od10_034.pdf
Brümmer, N., De Villiers, E., 2011. The BOSARIS toolkit: theory, algorithms and code for surviving the new DCF. NIST SRE11 Speaker Recognition Workshop, pp. 1–23. https://sites.google.com/site/nikobrummer/bosaris_toolkit_full_paper.pdf.Atlanta, Georgia, USA
Burget, L., Plchot, O., Cumani, S., Glembek, O., Matejka, P., Brummer, N., 2011. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011. IEEE, Prague, Czech Republic, pp. 4832–4835. https://doi.org/10.1109/ICASSP.2011.5947437.
Cai, W., Cai, Z., Zhang, X., Wang, X., Li, M., 2018. A novel learnable dictionary encoding layer for end-to-end language identification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Calgary, Canada, pp. 5189–5193. https://doi.org/10.1109/ICASSP.2018.8462025. 1804.00385.
Cai, W., Chen, J., Li, M., 2018. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. Odyssey 2018 The Speaker and Language Recognition Workshop. ISCA, Les Sables d'Olonne, France, pp. 74–81. https://doi.org/10.21437/Odyssey.2018-11.http://arxiv.org/abs/1804.00385http://www.isca-speech.org/archive/Odyssey_2018/abstracts/26.html, doi:10.21437/Odyssey.2018-11, arXiv:1804.05160
Chung, J.S., Nagrani, A., Zisserman, A., 2018. VoxCeleb2: deep speaker recognition. In: Proceedings of the 19th Annual Conference of the International Speech Communication Association, INTERSPEECH 2018. ISCA, Hyderabad, India, pp. 1086–1090. https://doi.org/10.21437/Interspeech.2018-1929.
Cumani, S., Brummer, N., Burget, L., Laface, P., 2011. Fast discriminative speaker verification in the i-vector space. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011. IEEE, Prague, Czech Republic, pp. 4852–4855. https://doi.org/10.1109/ICASSP.2011.5947442.
Cumani, S., Glembek, O.O., Brummer, N., De Villiers, E., Laface, P., 2012. Gender independent discriminative speaker recognition in i-vector space. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012. IEEE, Kyoto, Japan, pp. 4361–4364. https://doi.org/10.1109/ICASSP.2012.6288885.
Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio, Speech Lang. Process. 19 (4), 788–798. https://doi.org/10.1109/TASL.2010.2064307.http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5545402&tag=1
Deng, J., Guo, J., Xue, N., Zafeiriou, S., 2019. ArcFace: additive angular margin loss for deep face recognition. In: Proceedings of Computer Vision and Pattern Recognition, CVPR 2019.http://arxiv.org/abs/1801.07698
Doddington, G.R., 2000. The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective. Speech Commun. 31 (2−3), 225–254. https://doi.org/10.1016/S0167-6393(99)00080-1.
Garcia-Romero, D., Espy-Wilson, C.Y., 2011. Analysis of I-vector length normalization in speaker recognition systems. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011. ISCA, Florence, Italy, pp. 249–252.http://www.isr.umd.edu/Labs/SCL/publications/conference/dgromero_is11_lnorm_final.pdf
Garcia-Romero, D., McCree, A., Shum, S.H., Brümmer, N., Vaquero, C., 2014. Unsupervised domain adaptation for i-vector speaker recognition. In: Proceedings of Odyssey 2014 - The Speaker and Language Recognition Workshop. ISCA, Joensuu, Finland, pp. 260–264.
Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., McCree, A., 2017. Speaker diarization using deep neural network embeddings. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017. IEEE, New Orleans, LA, USA, pp. 4930–4934.
Godfrey, J.J., Holliman, E.C., Jane, M., 1992. SWITCHBOARD: telephone speech corpus for research and development. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 1992, San Francisco, California, USA, 1, pp. 517–520.
Han, K., Chandrashekaran, A., Kim, J., Lane, I., 2018. Densely connected networks for conversational speech recognition. In: Proceedings of the 19th International Conference of International Speech Communication Association, Interspeech 2018. ISCA, Hyderabad, India, pp. 796–800. https://doi.org/10.21437/Interspeech.2018-1486.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition.http://arxiv.org/abs/1512.03385
Heigold, G., Moreno, I., Bengio, S., Shazeer, N., 2016. End-to-end text-dependent speaker verification. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016. IEEE, Shanghai, China.http://arxiv.org/abs/1509.08062
Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, USA, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.http://ieeexplore.ieee.org/document/8099726/
Kenny, P., 2010. Bayesian speaker verification with heavy-tailed priors. In: Proceedings of Odyssey 2010 - The Speaker and Language Recognition Workshop. ISCA, Brno, Czech Republic.http://www.crim.ca/perso/patrick.kenny/kenny_Odyssey2010.pdf
Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Joint factor analysis versus eigenchannels in speaker recognition. IEEE Trans. Audio, Speech Lang. Process. 15 (4), 1435–1447. https://doi.org/10.1109/TASL.2006.881693.http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4156202
Kingma, D.P., Ba, J., 2015. Adam: a method for stochastic optimization. In: Proceedings of the International Conference of Learning Representations, ICLR 2015, San Diego, CA, USA, pp. 1–15. http://arxiv.org/abs/1412.6980.
Lei, Y., Scheffer, N., Ferrer, L., McLaren, M., 2014. A novel scheme for speaker recognition using a phonetically-aware deep neural network. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014. IEEE, Florence, Italy, pp. 1714–1718.
Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L., 2017. SphereFace: deep hypersphere embedding for face recognition. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 6738–6746. https://doi.org/10.1109/CVPR.2017.713. 1704.08063.http://ieeexplore.ieee.org/document/8100196/
Martin, A.F., Greenberg, C.S., Stanford, V.M., Howard, J.M., Doddington, G.R., Godfrey, J.J., 2014. Performance factor analysis for the 2012 NIST speaker recognition evaluation. In: Proceedings of the 15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014. ISCA, Singapore, pp. 1135–1138.
Martin, A.F., Przybocki, M., 2001. The NIST speaker recognition evaluations: 1996−2001. In: Proceedings of Odyssey 2001 - The Speaker and Language Recognition Workshop. ISCA, Crete, Greece, pp. 225–254. https://doi.org/10.1016/S0167-6393(99)00080-1.
Matejka, P., Zhang, L., Ng, T., Mallidi, S.H., Glembek, O., Ma, J., Zhang, B., 2014. Neural network bottleneck features for language identification. In: Proceedings of Odyssey 2014 - The Speaker and Language Recognition Workshop. ISCA, Joensuu, Finland, pp. 3–8.http://cs.uef.fi/odyssey2014/program/pdfs/35.pdf
McCree, A., Snyder, D., Sell, G., Garcia-Romero, D., 2018. Language recognition for telephone and video speech : the JHU HLTCOE submission for NIST LRE17. In: Proceedings of Odyssey 2018 - The Speaker and Language Recognition Workshop, Les Sables d'Olonne, France.
McLaren, M., Castan, D., Ferrer, L., Lawson, A., 2016. On the issue of calibration in DNN-based speaker recognition systems. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association, INTERSPEECH 2016, San Francisco, California, USA, pp. 1825–1829. https://doi.org/10.21437/Interspeech.2016-1134.
McLaren, M., Ferrer, L., Castan, D., Lawson, A., 2016. The speakers in the wild (SITW) speaker recognition database. In: Proceedings of the 17th Annual Conference of the International Speech Communication Association, INTERSPEECH 2016, pp. 818–822. https://doi.org/10.21437/Interspeech.2016-1129.
Nagrani, A., Chung, J.S., Zisserman, A., 2017. VoxCeleb: a large-scale speaker identification dataset. In: Proceedings of the 18th Annual Conference of the International Speech Communication Association, Interspeech 2017. ISCA, Stockholm, Sweden, pp. 2616–2620. https://doi.org/10.21437/Interspeech.2017-950.
NIST Multimodal Information Group, 2012. The NIST year 2012 speaker recognition evaluation plan. Technical Report. NIST. http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplan-v17-r1.pdf
Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in PyTorch. NIPS 2017 Workshop Autodiff.
Peddinti, V., Povey, D., Khudanpur, S., 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In: Proceedings of the 16th Annual Conference of the International Speech Communication Association, INTERSPEECH 2015. ISCA, Dresden, Germany, pp. 3214–3218.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohamadi, M., Khudanpur, S., 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In: Proceedings of the 19th Annual Conference of the International Speech Communication Association, INTERSPEECH 2018, Hyderabad, India. http://danielpovey.com/files/2018_interspeech_tdnnf.pdf.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., 2011. The kaldi speech recognition toolkit. IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU, pp. 1–4.

Povey, D., Zhang, X., Khudanpur, S., 2015. Parallel training of DNNs with natural gradient and parameter averaging. In: Proceedings of the International Conference of Learning Representations, ICLR 2015. 1410.7455.https://arxiv.org/abs/1410.7455http://arxiv.org/abs/1410.7455

Przybocki, M., Martin, A.F., Le, A.N., 2007. NIST Speaker recognition evaluations utilizing the mixer corpora - 2004, 2005, 2006. IEEE Trans. Audio, Speech Lang. Process. 15 (7), 1951–1959. https://doi.org/10.1109/TASL.2007.902489.http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4291612

Reynolds, D.A., Quatieri, T.F., Dunn, R.B., 2000. Speaker verification using adapted gaussian mixture models. Digital Signal Process. 10 (1−3), 19–41. https://doi.org/10.1006/dspr.1999.0361.

Reynolds, D.A., Torres-Carrasquillo, P., 2005. Approaches and applications of audio diarization. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005. IEEE, Philadelphia, Pennsylvania, USA, pp. 953–956. https://doi.org/10.1109/ICASSP.2005.1416463.

Rezaur rahman Chowdhury, F.A., Wang, Q., Moreno, I.L., Wan, L., 2018. Attention-based models for text-dependent speaker verification. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Calgary, Canada, pp. 5359–5363. https://doi.org/10.1109/ICASSP.2018.8461587.

Richardson, F., Reynolds, D.A., Dehak, N., 2015. Deep neural network approaches to speaker and language recognition. IEEE Signal Process. Lett. 22 (10), 1671–1675. https://doi.org/10.1109/LSP.2015.2420092.

Sadjadi, S.O., Greenberg, C.S., Reynolds, D.A., Singer, E., Mason, L., Hernandez-Cordero, J., 2019. The 2018 NIST speaker recognition evaluation. Interspeech 2019 (submitted).

Sadjadi, S.O., Kheyrkhah, T., Tong, A., Greenberg, C.S., Reynolds, D.A., Singer, E., Mason, L., Hernandez-Cordero, J., 2017. The 2016 NIST speaker recognition evaluation. Interspeech 2017. ISCA, ISCA, pp. 1353–1357. https://doi.org/10.21437/Interspeech.2017-458.http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0458.html

Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., Manohar, V., Dehak, N., Povey, D., Watanabe, S., Khudanpur, S., 2018. Diarization is hard: some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. In: Proceedings of the 19th Annual Conference of the International Speech Communication Association, INTERSPEECH 2018, Hyderabad, India, p. 2808—-2812. https://doi.org/10.21437/Interspeech.2018-1893.

Silnova, A., Brümmer, N., Garcia-Romero, D., Snyder, D., Burget, L., 2018. Fast variational bayes for heavy-tailed PLDA applied to i-vectors and x-vectors. Interspeech 2018, Hyderabad, India, pp. 72–76. https://doi.org/10.21437/Interspeech.2018-2128. arXiv:1803.09153v1.

Sizov, A., Lee, K.A., Kinnunen, T., 2014. Unifying probabilistic linear discriminant analysis variants in biometric authentication. Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg, pp. 464–475. https://doi.org/10.1007/978-3-662-44415-3_47.

Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., Khudanpur, S., 2018. Spoken language recognition using X-vectors. Odyssey 2018, Les Sables d'Olonne, France.

Snyder, D., Garcia-Romero, D., Povey, D., Khudanpur, S., 2017. Deep neural network embeddings for text-independent speaker verification. In: Proceedings of the 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017. ISCA, Stockholm, Sweden, pp. 999–1003. https://doi.org/10.1109/SLT.2016.7846260. 1703.01898.http://www.danielpovey.com/files/2017_interspeech_embeddings.pdf

Snyder, D., Garcia-Romero, D., Sell, G., McCree, A., Povey, D., Khudanpur, S., 2019. Speaker recognition for multi-speaker conversations using X-vectors. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019. IEEE, Brighton, UK.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-Vectors : robust DNN embeddings for speaker recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018. IEEE, Alberta, Canada, pp. 5329–5333.

Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., Khudanpur, S., 2016. Deep neural network-based speaker embeddings for end-to-end speaker verification. In: Proceedings of the 2016 IEEE Spoken Language Technology Workshop (SLT). IEEE, San Diego, CA, USA, pp. 165–170. https://doi.org/10.1109/SLT.2016.7846260.

Tracey, J., Strassel, S., 2018. VAST : a Corpus of video annotation for speech technologies main corpus sub-corpora. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaky, Japan, pp. 4318–4321.

Variani, E., Lei, X., McDermott, E., Moreno, I.L., Gonzalez-Dominguez, J., 2014. Deep neural networks for small footprint text-dependent speaker verification. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, Florence, Italy, pp. 4052–4056. https://doi.org/10.1109/ICASSP.2014.6854363. 1408.5882v1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. Advances in Neural Information Processing Systems, NIPS 2017, p. 5998—-6008. 1706.03762.

Villalba, J., 2014. Advances on Speaker Recognition in non Collaborative Environments. University of Zaragoza Ph.D. thesis.

Villalba, J., Brummer, N., 2011. Towards fully Bayesian speaker recognition: integrating out the between-speaker covariance. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association, Interspeech 2011. ISCA, Florence, Italy, pp. 505–508. https://www.isca-speech.org/archive/archive_papers/interspeech_2011/i11_0505.pdf

Villalba, J., Brümmer, N., Dehak, N., 2018. End-to-end versus embedding neural networks for language recognition in mismatched conditions. In: Proceedings of Odyssey 2018 - The Speaker and Language Recognition Workshop. https://www.isca-speech.org/archive/Odyssey_2018/pdfs/71.pdf.Les Sables d'Olonne, France

Villalba, J., Lleida, E., 2012. Bayesian adaptation of PLDA based speaker recognition to domains with scarce development data. In: Proceedings of Odyssey 2012 - The Speaker and Language Recognition Workshop. COLIPS, Singapore. https://www.isca-speech.org/archive/odyssey_2012/papers/od12_047.pdf

Villalba, J., Lleida, E., 2014. Unsupervised adaptation of PLDA by using variational bayes methods. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014. IEEE, Florence, Italy, pp. 744–748. https://doi.org/10.1109/ICASSP.2014.6853695.

Villalba, J., Lleida, E., Ortega, A., Miguel, A., 2013. The I3A speaker recognition system for NIST SRE12: post-evaluation analysis. In: Proceedings of the 14th Annual Conference of the International Speech Communication Association, Interspeech 2013. ISCA, Lyon, France, pp. 3679–3683. https://www.isca-speech.org/archive/archive_papers/interspeech_2013/i13_3679.pdf

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., Lang, K., 1989. Phoneme recognition using time-delay neural networks. IEEE Trans. Acoust., Speech, Signal Process. 37 (3), 328–339. https://doi.org/10.1109/29.21701.

Zhang, C., Koishida, K., 2017. End-to-end text-independent speaker verification with triplet loss on short utterances. In: Proceedings of the 18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017. ISCA, Stockholm, Sweden, pp. 1487–1491. https://doi.org/10.21437/Interspeech.2017-1608.

Zhang, S.X., Chen, Z., Zhao, Y., Li, J., Gong, Y., 2017. End-to-end attention based text-dependent speaker verification. 2016 IEEE Workshop on Spoken Language Technology, SLT 2016 - Proceedings. IEEE, San Diego, CA, USA, pp. 171–178. https://doi.org/10.1109/SLT.2016.7846261. 1701.00562.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C.L.Y., Courville, A., 2017. Towards end-to-end speech recognition with deep convolutional neural networks. https://doi.org/10.21437/Interspeech.2016-1446.http://arxiv.org/abs/1701.02720