Odyssey 2018 The Speaker and Language Recognition Workshop
26-29 June 2018, Les Sables d'Olonne, France

On the use of X-vectors for Robust Speaker Recognition

Ondřej Novotný, Oldřich Plchot, Pavel Matějka, Ladislav Mošner, and Ondřej Glembek

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Czechia
inovoton@fit.vut.cz

Abstract

Text-independent speaker verification (SV) is currently in the process of embracing DNN modeling in every stage of SV system. Slowly, the DNN-based approaches such as end-to-end modelling and systems based on DNN embeddings start to be competitive even in challenging and diverse channel conditions of recent NIST SREs. Domain adaptation and the need for a large amount of training data are still a challenge for current discriminative systems and (unlike with generative models), we see significant gains from data augmentation, simulation and other techniques designed to overcome lack of training data. We present an analysis of a SV system based on DNN embeddings (x-vectors) and focus on robustness across diverse data domains such as standard telephone and microphone conversations, both in clean, noisy and reverberant environments. We also evaluate the system on challenging far-field data created by re-transmitting a subset of NIST SRE 2008 and 2010 microphone interviews. We compare our results with the state-of-the-art i-vector system. In general, we were able to achieve better performance with the DNN-based systems, but most importantly, we have confirmed the robustness of such systems across multiple data domains.

Index Terms: Speaker Recognition, Embedding, X-vectors, DNN

1. Introduction

In recent years, there have been many attempts to take advantage of neural networks (NNs) in speaker verification. They slowly found their way into the state-of-the-art systems that are based on modeling the fixed-length utterance representations, such as i-vectors [1], by Probabilistic Linear Discriminant Analysis (PLDA) [2].

Most of the efforts to integrate the NNs into the SV pipeline involved replacing or improving one or more of the components of an i-vector + PLDA system (feature extraction, calculation of sufficient statistics, i-vector extraction or PLDA classifier) with a neural network. On the front-end level, let us mention for example using NN bottleneck features (BNF) instead of conventional MFCC features [3] or simply concatenating BNF and MFCCs [4]. Later in the modeling stage, NN acoustic models can be used instead of Gaussian Mixture Models (GMM) for extraction of sufficient statistics [5] or for either complementing PLDA [6, 7] or replacing it [8].

The work was supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK", Google Faculty Research Award program, Czech Science Foundation under project No. GJ17-23870Y, and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

This line of work has logically resulted in attempts to train a larger DNN directly for the speaker verification task, i.e., binary classification of two utterances as a *target* or a *non-target* trial [9, 10, 11, 12]. Such systems are known as *end-to-end* systems and have been proven competitive for text-dependent tasks [9, 10] as well as text-independent tasks with short test utterances and an abundance of training data [11]. On text-independent tasks with longer utterances and moderate amount of training data, the i-vector inspired end-to-end system [12] already outperforms generative baselines, but at the cost of high complexity in memory and computational costs during training.

While the fully end-to-end SV systems have been struggling with large requirements on the amount of training data (often not available to the researchers) and high computational costs, focus on speaker recognition has shifted back to generative modeling, but now with utterance representations obtained from a single NN. Such NN takes the frame level features of an utterance as an input and directly produces an utterance level representation, usually referred to as an *embedding* [13, 9, 10, 14, 15]. The embedding is obtained by the means of a *pooling mechanism* (for example taking the mean) over the frame-wise outputs of one or more layers in the NN [13], or by the use of a recurrent NN [9]. One effective approach is to train the NN for classifying a set of training speakers, i.e., using multiclass training [13, 14, 15]. In order to do speaker verification, the embeddings are extracted and used in a standard backend, e.g., PLDA. Such systems have recently been proven competitive for both short and long utterance durations in text-independent speaker verification [14, 15].

In this work, we use the model proposed by David Snyder [15] and extend the analysis in [16] which already presents the *x-vector* (the embedding) as a robust feature for PLDA modeling, and provides state-of-the-art results on NIST SRE 2016 and Speakers In The Wild (SITW) challenge. We build on top of the available Kaldi recipe [17], and we modify the training data set; this allows for testing also on benchmarks like NIST SRE 2010 (English telephone and microphone data), PRISM [18] (to analyze noise and reverberation robustness), and our own far-field dataset based on NIST SRE10. We also experiment with training data, analyzing separately the effect of augmentation and amount of training speakers. We decided to omit the techniques necessary to obtain state-of-the-art results on SRE16, such as adaptive score normalization or unsupervised adaptation of PLDA, as we want to explore general robustness across many channels, and provide baselines for further research. We also provide a detailed description of additional data augmentations that we used on top of the original Kaldi recipe.

10.21437/Odyssey.2018-24

2. Speaker Recognition Systems

Our goal is to provide a comprehensive analysis of systems based on x-vectors. We will therefore present several systems that differ in training of the x-vector extraction DNN, and compare them with the state-of-the-art i-vector system based on stacked bottleneck features and MFCCs.

2.1. Baseline i-vector System

For our the i-vector system, we use the simple and effective recipe with MFCCs concatenated with stacked bottleneck features [4].

19 MFCCs, from 24 Mel-filter banks, together with log-energy were extracted using a 25 ms Hamming window over 10 ms frames. Bandwidth was limited to 120-3800 Hz and MFCCs were augmented with their delta and double delta coefficients calculated using a 5 frame window. Resulting 60-dimensional vectors are subjected to feature warping using a 3 s sliding window before removing the silence.

Bottleneck Neural-Network (BN-NN) refers to such topology of a NN, where one of the hidden layers has significantly lower dimensionality than the surrounding layers. A bottleneck feature vector is generally understood as a by-product of forwarding a primary input feature vector through the BN-NN and reading off the vector of values at the bottleneck layer. We have used a cascade of two such NNs for our experiments. The output of the first network is *stacked* in time, defining context-dependent input features for the second NN, hence the term Stacked Bottleneck features.

The NN input features are 24 log Mel-scale filter bank outputs augmented with fundamental frequency features from 4 different f_0 estimators (Kaldi, Snack[1], and other two according to [19] and [20]). Together, we have 13 f_0 related features, see [21] for more details. Conversation-side based mean subtraction is applied on the whole feature vector, then 11 frames of log filter bank outputs and fundamental frequency features are stacked. Hamming window and DCT projection (0^{th} to 5^{th} DCT base) are applied on the time trajectory of each parameter resulting in $(24 + 13) \times 6 = 222$ coefficients on the first stage NN input.

The configuration of the first NN is $222 \times D_H \times D_H \times D_{BN} \times D_H \times K$, where $K = 9824$ is the number of target triphones. The dimensionality of the bottleneck layer, D_{BN} was set to 80 (this was shown as optimal in [22]). The dimensionality of other hidden layers was set to 1500. The bottleneck outputs from the first NN are sampled at times $t-10$, $t-5$, t, $t+5$ and $t+10$, where t is the index of the current frame. The resulting 400-dimensional features are inputs to the second stage NN with the same topology as the first stage. The network was trained on Fisher English corpus, and data were augmented with two noisy copies.

Finally, the 80-dimensional bottleneck outputs from the second NN (referred as SBN) are concatenated with MFCCs and taken as features for the conventional GMM/UBM i-vector system, with 2048 components in UBM and 600-dimensional i-vectors.

Voice activity detection (VAD) was performed by the BUT Czech phoneme recognizer [23], dropping all frames that are labeled as silence or noise. The recognizer was trained on the Czech CTS data, but we have added noise with varying SNR to 30% of the database.

[1] http://kaldi.sourceforge.net, www.speech.kth.se/snack/

2.2. The Embedding System

We experiment with a DNN architecture for embeddings described in [15] and [16]. Specifically, we use the Kaldi recipe [17] from David Snyder and 512 dimensional embeddings extracted from the first layer after the pooling layer (embedding-a, also referred to as the x-vector), which is consistent with [16].

Input features to the DNN were MFCCs, extracted using a 25 ms Hamming window. We used 23 Mel-filter banks and we limited the bandwidth to 20–3700 Hz range. 20 MFCCs were calculated every 10 ms. This 20-dimensional feature vector was subjected to short time mean- and variance-normalization using a 3 s sliding window.

For training, we used default energy based VAD from the Kaldi recipe. For embedding extraction, we applied the same VAD as for the i-vector system.

The embedding DNN [16], can be divided into three parts. The first part operates on the frame level and begins with 5 layers of time-delay architecture [24]. The first four layers contain each 512 neurons, the last layer before statistic pooling has 1500 neurons. The consequent pooling layer gathers mean and standard deviation statistics from all frame-level inputs. The single vector of concatenated means and standard deviations is propagated through the rest of the network. The part of the network where embeddings are extracted consists of two hidden layers each with 512 neurons and the final output layer has a dimensionality corresponding to the number of speakers. The DNN uses ReLus as non-linearities in hidden layers, soft-max on the output layer and is trained by optimizing multi-class cross entropy.

3. Experimental Setup

We used the PRISM [18] training dataset definition without added noise or reverberation to train UBM and i-vector transformation. The set comprises Fisher 1 and 2, Switchboard phase 2 and 3 and Switchboard cellphone phases 1 and 2, along with a set of Mixer speakers. This includes the 66 held out speakers from SRE10 (see Section III-B5 of [18]), and 965, 980, 485 and 310 speakers from SRE08, SRE06, SRE05 and SRE04, respectively. A total of 13,916 speakers are available in Fisher data and 1,991 in Switchboard data.

Five variants of gender-independent PLDA models were trained: one only on the clean training data, the rest included also artificially added different mixes of noises and reverberation. Artificially added noise and reverb segments totaled approximately 24000 segments or 30% of total number of clean segments for PLDA training, see details in Sec. 3.2.

We evaluated our systems on the *female* portions of the following conditions in NIST SRE 2010 [25] and PRISM [18]:

- *tel-tel*: SRE 2010 extended telephone condition involving normal vocal effort conversational telephone speech in enrollment and test (known as "condition 5").
- *int-int*: SRE 2010 extended interview condition involving interview speech from different microphones in enrollment and test (known as "condition 2").
- *int-mic*: SRE 2010 extended interview-microphone condition involving interview enrollment speech and normal vocal effort conversational telephone test speech recorded over a room microphone channel (known as "condition 4").
- *prism,noi*: Clean and artificially noised waveforms from both interview and telephone conversations recorded over

lavalier microphones. Noise was added at different SNR levels and recordings tested against each other.

- *prism,rev*: Clean and artificially reverberated waveforms from both interview and telephone conversations recorded over lavalier microphones. Reverberation was added with different RTs and recordings are tested against each other.
- *prism,chn*: English telephone conversation with normal vocal effort recorded over different microphones from both SRE2008 and 2010 are tested against each other.

Additionally, we used the *Core-Core* condition from the SITW challenge – *sitw-core-core*. SITW [26] dataset is a large collection of real-world data exhibiting speech from individuals across a wide array of challenging acoustic and environmental conditions. These audio recordings do not contain any artificially added noise, reverberation or other artifacts. This database was collected from open-source media. The *sitw-core-core* condition comprises audio files each containing a continuous speech segment from a single speaker. Enrollment and test segments contain between 6-180 seconds of speech. We scored all trials (both genders).

We also test on NIST SRE 2016 [27], but we split the trial set by language into Tagalog (*sre16-tgl-f*) and Cantonese (*sre16-yue-f*). We use only female trials (both single- and multi-session). We did not use SRE'16 unlabeled development set in any way.

The speaker verification performance is evaluated in terms of the equal error rate (EER).

3.1. NIST Retransmitted Set (BUT-RET)

To evaluate the impact of room acoustics on the accuracy of speaker recognition, a proper dataset of reverberant audio is needed. An alternative that fills a qualitative gap between unsatisfying simulation (despite the improvement of realism [28]) and costly and demanding real speaker recording is retransmission. We can also advantageously use the fact that a *known* dataset can be retransmitted so that the performances are readily comparable with known benchmarks. Hence, this was the method to obtain a new dataset.

The retransmission took place in a room whose floor plan is displayed in Figure 1. The outcome of retransmission is supposed to be used for many tasks in the future, hence the layout of microphones is intentional. The loudspeaker-microphone distance rises steadily for microphones 1…6 to study deterioration as a function of distance. Microphones 7…12 form a large microphone array mainly focused to explore beamforming. Here, we use them as single microphones in different positions with respect to the speaker.

For this work, a subset of NIST SRE 2010 data was retransmitted. The dataset consists of 459 female recordings with durations of three and eight minutes. The total number of female speakers is 150. The files were played in sequence and recorded simultaneously by a multi-channel acquisition card that ensured sample precision synchronization.

We denote the retransmitted data as condition *BUT-RET-*∗, where BUT-RET-*orig*, represents original (not retransmitted) data and BUT-RET-*avg* represents the average of results from all fourteen microphones.

3.2. PLDA Augmentation Sets

For extension of the PLDA training set, we created new artificially corrupted training sets from the PRISM training set.

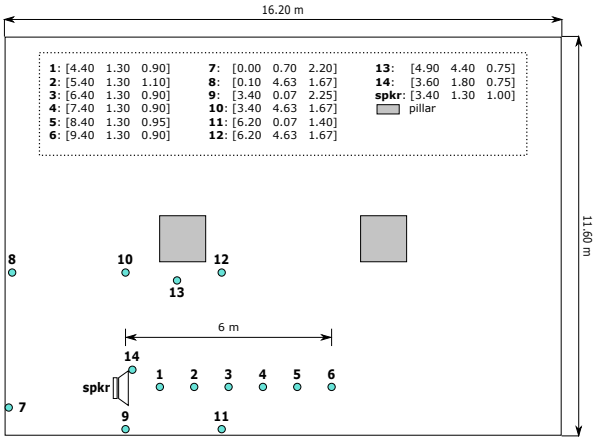

Figure 1: Floor plan of the room in which the retransmission took place. Coordinates are in meters and lower left corner is the origin.

3.2.1. Adding Noise

We prepared a noise dataset from three sources of different types of noise:

- 272 samples (4 minutes long) taken from the Freesound library [2] (real fan, HVAC, street, city, shop, crowd, library, office and workshop).
- 7 samples (4 minutes long) of artificially generated noises: various spectral modifications of white noise + 50 and 100 Hz hum.
- 25 samples (4 minutes long) of babbling noises by merging speech from 100 random speakers from Fisher database using speech activity detector.

Noises were divided into three disjoint groups for training (223 files), development (40 files) and test (41 files). Development and test subsets are not used in this work.

3.2.2. Reverberation

We prepared two sets with room impulse responses (RIRs). The first set consists of real room impulse responses from several databases: AIR [29], C4DM [30, 31], MARDY [32], OPENAIR [33], RVB 2014 [34], and RWCP [35]. Together, they form a set with all types of rooms (small rooms, big rooms, lecture room, restrooms, halls, stairs etc.). All room models have more than one impulse response per room (different RIR was used for source of the signal and source of the noise to simulate different locations of their sources). Rooms were split into two disjoint sets, with 396 rooms for training, 40 rooms for test.

The second set consists of artificially generated room impulse responses using "Room Impulse Response Generator" tool from E. Habets [36]. The tool can model the size of room (3 dimensions), reflectivity of each wall, type of microphone, position of source and microphone, orientation of microphone towards the audio source, and number of bounces (reflections) of the signal. We generated a pair of RIRs for each room model (one used for source of the sound, one for source of the noise). Again we generated two disjoint sets, with 1594 RIRs for training and 250 RIRs for test. The test subset is not used in this work.

[2] http://www.freesound.org

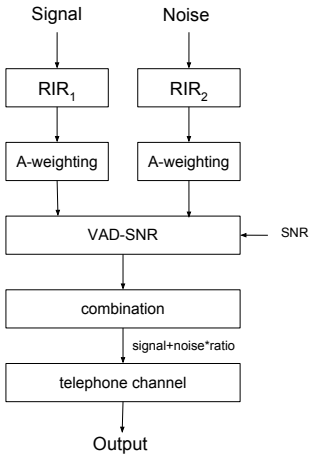

Figure 2: The process of data preparation (corruption) for new SRE condition design.

3.2.3. Composition of the Augmented Training Set

To mix the reverberation, noise and signal at given SNR, we followed the procedure outlined in Figure 2. The pipeline begins with two branches, when speech and noise are reverberated separately. Different RIRs from the same room are used for signal and noise, to simulate different positions of sources.

The next step is A-weighting. A-weighting is applied to simulate the perception of the human ear to added noise [37]. With this filtering, the listener would be able to better perceive the SNR, because most of the noise energy is coming from frequencies that the human ear is sensitive to.

In the following step, we set a ratio of noise and signal energies to obtain the required SNR. Energies of the signal and noise are computed from frames given by original signal's voice activity detection (VAD). It means the computed SNR is really present in speech frames which are important for our recognition (frames without voice activity are removed during processing).

After the combination, where signal and noise are summed together at desired SNR, we filter the resulting signal with telephone channel. In case we want to add only noise or reverberation, the appropriate part of the algorithm is used.

3.3. Embedding Augmentation Sets

We experimented with four sets of DNN training data.We kept most of the parameters from the original recipe. Every speaker has to have at least 6 utterances (set to 8 in the original recipe) and every utterance has to be at least 500 frames long. The consequence of this constraint is having fewer speakers, especially in training on clean data, because there, utterances were not duplicated by the augmentation. It is worth noting that increasing the number of training speakers also increases the model size due to the larger output layer.

The statistics of the training data for all four models are listed in Table 1. Our first model was trained only on "clean" original data without any augmentation. The second model (Aug I.) was trained on augmented data, but the number of speakers was limited to be the same as in the first model. The third model (Aug II.) was similar to the second but without any limitation.

In the original Kaldi recipe, training data were augmented with reverberation, noise, music, and babble and combined with original clean data. The package of all noises and room impulse responses can be downloaded from OpenSLR[3] [38], and includes MUSAN noise corpus (843 noises).

For data *augmentation with reverberation*, the total amount of RIRs is divided into two lists for medium and small rooms. A probability of selecting a small or medium room is the same. We add the reverberation to obtain a single replica of original training data.

For *augmentation with noise*, we created three replicas of the original data. The first replica was augmented by adding MUSAN noises at SNR levels in the range of 0-15 dB. In this case, the noise was added as a foreground noise (that means several non-overlapping noises can be added to the input audio). The second replica was augmented by music at SNRs from 5 to 15dB as background noise (one noise per audio with the given SNR). The last noisy replica of training data was created by augmentation with babble noise. SNR levels were at 13-20 dB and we used 3-7 noises per audio. All augmented data were pooled and a random subset of 128k audios was selected and combined with clean data. The process of data augmentation is also described in [16].

For the last model (Aug III.), we add our augmentation: real room impulse responses and stationary noises described in Sec. 3.2. The original RIR list was extended by our list of real RIRs and we kept one reverberated replica. Our stationary noises were used to create another replica of data with SNR levels in range 0-20 dB. We combined all replicas and selected a subset of 150k files. As a result, we obtained 11383 speakers after filtering.

Table 1: Numbers of speakers, utterances and speech length used for training the embedding DNN.

Parameter	[E]mbedding -clean	E-Aug I.	E-Aug II.	E-Aug III.
speakers	3359	3359	9544	11383
utterances	58965	72371	211906	268219
speech duration [h]	2488	3494	10289	13288

4. Experiments and Discussion

We conducted a set of experiments with embeddings to analyze their robustness in different data domains. We also perform analysis with embedding DNNs aimed at answering the following two questions: (i) How do embeddings compare to the traditional state-of-art system with multi-condition training? (ii) How does the embeddings' performance depend on the amount and type of the training data for the embedding DNN?

In our experiments with i-vectors (600-dimensional vectors) and x-vectors (512-dimensional vectors), we used identical preprocessing. First, we reduced the dimensionality by LDA to 200 dimensions and then we subjected the reduced vectors to the global mean- and length- normalization [1, 39]. Speaker verification is performed by means of PLDA [40]. We do not include any adaptive score normalization, unsupervised PLDA adaptation or other tricks that are necessary to achieve the best possible performance on SRE16.

[3] `http://www.openslr.org/resources/28/rirs_noises.zip`

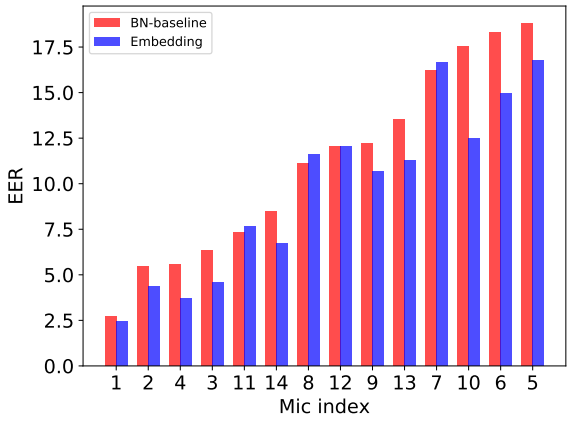

Figure 3: Results on retransmitted NIST data for all microphone from Fig. 1. Ordered by EER from the baseline system. Red bars are for the baseline, blue bars for the Embedding-Aug III.

As a reference and cross-check, we include results with embeddings from the original Kaldi [16] recipe in Table 2. Our implementation of PLDA was trained on the same list as in the recipe and, this time only, we used a similar form of adaptive score normalization. We did not perform unsupervised PLDA adaptation, so these results are comparable to the *unadapted* ones as they are printed by the recipe.

Table 2: Results (EER [%]) achieved with original Kaldi recipe from [16, 17]. Results are presented on gender-independent conditions.

Condition	BN-baseline	Embedding-Kaldi
sre16-tgl	19.5	19.8
sre16-yue	9.10	6.90
sitw-core-core	7.68	7.14

4.1. Comparison with the Baseline

At first, we will focus on the comparison of the baseline system with our best embedding system (Embedding - Aug III) in Table 3. Results are separated into two main blocks: for the baseline system and for the embedding system. Columns of each block represent different scenarios in PLDA training. In the first column, the results correspond to the system where the PLDA was trained only on the clean data without any augmentation. In the next five columns, we list the results for multi-condition training. We trained five different PLDAs, every time using a different mix of corrupted data added to the training list: N - PLDA extended by the noised data, AR - PLDA extended by the data corrupted with artificial generated RIRs, RR - PLDA extended by the data corrupted with the real RIRs. (A/R)R+N - PLDA extended by the data with both types of distortion (noise and reverberation).

The table is also divided into blocks according to the test data domain: telephone channel (conditions *tel-tel, sre16-tgl-f, sre16-yue-f*), various microphones (conditions *int-int, int-mic, prism,chn, sitw-core-core*), artificially corrupted data with additive noise *prism,noi* and reverberation *prism,rev*, far-field microphone data (retransmitted) *BUT-RET-avg*.

When analyzing the telephone block, we can observe a significant drop in performance on relatively clean English data *tel-tel*. This can be perhaps explained by a very good match in the training data for i-vectors w.r.t. clean telephone conversations. On more challenging SRE16, where we deal with new and noisier telephone channels (landline and cellular telephone networks from China and Philippines), we observe a clear improvement on Cantonese. Results on Tagalog are almost the same and the reason for a low performance w.r.t. Cantonese is still unexplained.

In the domain which involves microphone data, we see some mixed results on easier conditions like *in-mic* and *prism,chn*. We observe that the *prism,chn* has been much better with the i-vector system and conversely *int-mic* is much better with the embedding one. On more difficult conditions like *in-int* and *sitw-core-core*, we can already see improvements with the embedding system, which can be further improved by multi-condition training in PLDA.

In the artificially created noisy and reverberant condition, we can see better performance from the i-vector baseline when training the PLDA without augmentation. When augmenting the PLDA training list, the difference in performance is lowered, but the baseline system still appears to handle reverberation better.

We have to admit that we are seeing inconsistent (and usually worse than the baseline) results with this embedding system on our other internal artificially created noisy and reverberant conditions. Luckily, we are continuously working on retransmitting the SRE data in real rooms, mostly for the purposes of our research in speaker recognition with microphone arrays and far-field data [41]. Experiments with these retransmitted data show consistently better or similar performance of the embedding system and a positive effect of multi-condition training of PLDA. We can see the performance on data recorded over individual microphones in Figure 3. Placement of microphones and room dimensions are depicted in Figure 1. The embedding system has always outperformed or matched the i-vector baseline. In case of three microphones (11, 8 and 7), the performance of embeddings was only marginally worse.

4.2. Analysis of Embedding DNN Training

Now we will focus on the analysis with training of the embedding system. We will vary the amount of training data (and also training speakers) by the means of augmentation and look closely at the results. Results from all four embedding networks are listed in Table 4. The first block represents the system trained only on original ("clean") data, without any augmentation. Every next block represent the system with increasing number of training utterances, speakers and types of noise in the augmentation (see Section 3.3 and Table 1).

Trends in the multi-condition PLDA training for all four systems are the same as in Table 3, so for the sake of clarity, we report only cases with real impulse responses and noise.

The first two blocks represent the same network. The difference is in augmentation: while *Embedding-clean* was trained only on clean, *Embedding-Aug I.*, is trained with augmented data, but we kept the number of speakers identical for both. We can clearly see that just adding additional hours of training data consistently improved the performance and also that the trend of contribution of the multi-condition PLDA training to the performance is the same.

The second and the third blocks represent the comparison of different numbers of speakers in training and therefore also change in the model size (size of the output layer of the network). *Embedding-Aug I.* has 3359 speakers as the output, while

Table 3: Comparison of bottle-neck i-vector baseline with neural network embedding in various data domains. Both blocks are divided into columns corresponding to the systems trained in multi-condition fashion (with noised and reverberated data in PLDA). Results (EER [%]) in each column correspond to the different PLDA multi-condition training set: N - noise, (A/R)R- artificial/real reverberation, or both (+).

	BN Baseline						Embedding-Aug III.					
	PLDA clean	PLDA extension data					PLDA clean	PLDA extension data				
Condition	-	N	AR	RR	AR+N	RR+N	-	N	AR	RR	AR+N	RR+N
tel-tel	0.94	1.04	**0.92**	0.93	**0.92**	0.93	1.3	1.43	1.27	1.27	1.28	1.29
sre16-tgl-f	21.88	**21.24**	21.76	21.82	21.92	21.93	22.73	22.52	22.69	22.87	22.53	22.56
sre16-yue-f	13.45	13.02	13.35	13.45	13.39	13.44	10.36	**9.61**	10.37	10.45	10.46	10.61
int-int	3.88	4.07	3.75	3.77	3.76	3.73	3.36	3.72	3.25	3.29	3.24	**3.22**
int-mic	1.85	1.69	1.75	1.76	1.78	1.78	1.33	1.43	1.3	1.3	1.24	**1.22**
prism,chn	0.40	0.46	0.39	0.39	0.37	**0.36**	0.62	0.81	0.6	0.61	0.61	0.61
sitw-core-core	8.09	7.85	8.11	8.02	8.04	8.03	7.87	**7.3**	7.84	7.72	7.32	7.41
prism,noi	2.43	1.98	2.45	2.45	2.16	2.2	2.76	**1.9**	2.72	2.63	2.09	2.11
prism,rev	1.42	1.39	1.38	**1.30**	1.36	1.31	2.08	2.02	1.91	1.79	1.69	1.6
BUT-RET-orig	1.45	1.58	1.46	1.47	1.48	**1.43**	1.73	1.73	1.65	1.69	1.69	1.63
BUT-RET-avg	11.64	11.48	11.49	11.2	11.59	11.12	11.51	10.78	10.86	10.21	10.4	**9.71**

Table 4: Results (EER [%]) obtained in four scenarios. Each block corresponds to the system trained on different data (see Table 1). Blocks are divided into columns corresponding to the systems trained in multi-condition fashion (with noised and reverberated data in PLDA). Each column correspond to the different PLDA multi-condition training set: N - noise, RR - real reverberation, or both (+).

	Embedding-clean				Embedding-Aug I.				Embedding-Aug II.				Embedding-Aug III.			
	PLDA clean	PLDA extension data			PLDA clean	PLDA extension data			PLDA clean	PLDA extension data			PLDA clean	PLDA extension data		
Condition	-	N	RR	RR+N	-	N	RR	RR+N	-	N	RR	RR+N	-	N	RR	RR+N
tel-tel	2.56	2.94	2.55	2.65	2.05	2.45	1.94	1.9	1.38	1.61	1.34	1.32	1.3	1.43	1.27	**1.29**
sre16-tgl-f	27.25	26.99	27.26	27.1	25.02	24.94	24.86	24.89	22.69	**22.14**	22.71	22.37	22.73	22.52	22.87	22.56
sre16-yue-f	15.51	15.78	15.45	15.22	14.12	13.31	14.05	13.89	10.27	9.9	10.43	10.45	10.36	**9.61**	10.45	10.61
int-int	5.01	5.57	5.0	4.84	4.49	4.84	4.47	4.28	3.45	3.87	3.45	3.39	3.36	3.72	3.29	**3.22**
int-mic	2.15	2.44	2.11	2.02	1.86	2.33	1.84	1.81	1.33	1.29	1.34	1.27	1.33	1.43	1.3	**1.22**
sitw-core-core	10.9	11.02	10.78	10.44	9.75	9.54	9.58	9.21	7.86	7.42	7.62	7.42	7.87	**7.3**	7.72	7.41

Embedding-Aug II. has seen 9544 speakers. Again, we can see the same trend as before and additional improvements in performance. The exception is the Tagalog condition of SRE16, which seems to be different and possibly too much out-of-domain.

The last block represents the largest network (*Embedding-Aug III.*). We extended the number of speakers and we also added more augmented data to the training. We added also the augmentation data for the PLDA multi-condition training (see Section 3.2), we added real room impulse responses and additional set of stationary noises. This brought small improvements on the *tel-tel* condition. On the *sre16-yue-f* and *sitw-core-core*, we can see similar performance, with the exception of small improvement achieved with PLDA+N. On the remaining conditions, the largest network has kept its robustness and similar performance.

5. Conclusion

We can conclude that the analyzed embedding architecture shows a robust performance under various conditions. We have presented new results on well-known benchmarks and we have performed an analysis with far-field data. We have verified that this architecture is indeed data-hungry by extending the original recipe with our collection of augmentation data and at the same time further improving the performance.

6. References

[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] S. J. D. Prince, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. International Conference on Computer Vision (ICCV)*, Rio de Janeiro, Brazil, 2007.

[3] A. Lozano-Diez, A. Silnova, P. Matějka, O. Glembek, O. Plchot, J. Pešán, L. Burget, and J. Gonzalez-Rodriguez, "Analysis and Optimization of Bottleneck Features for Speaker Recognition," in *Proceedings of Odyssey 2016*. 2016, vol. 2016, pp. 352–357, International Speech Communication Association.

[4] Pavel Matějka, Ondřej Glembek, Ondřej Novotný, Oldřich Plchot, František Grézl, Lukáš Burget, and Jan Černocký, "Analysis Of DNN Approaches To Speaker Identification," in *Proceedings of the 41th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016), 2016*. 2016, pp. 5100–5104, IEEE Signal Processing Society.

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

[6] S. Novoselov, T. Pekhovsky, O. Kudashev, V. S. Mendelev, and A. Prudnikov, "Non-linear PLDA for i-vector speaker verification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Sept 2015, pp. 214–218.

[7] G. Bhattacharya, J. Alam, P. Kenny, and V. Gupta, "Modelling speaker and channel variability using deep neural networks for robust speaker verification," in *2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16*, 2016.

[8] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1700–1704.

[9] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5115–5119.

[10] S. X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-End attention based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 171–178.

[11] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, Dec 2016, pp. 165–170.

[12] Johan Rohdin, Anna Silnova, Mireia Diez, Oldřich Plchot, Pavel Matějka, and Lukáš Burget, "End-to-end DNN based speaker recognition inspired by i-vector and PLDA," in *Proceedings of ICASSP*. 2018, IEEE Signal Processing Society.

[13] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4052–4056.

[14] G. Bhattacharya, J. Alam, and P. Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification," in *Interspeech 2017*, 08 2017, pp. 1517–1521.

[15] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[16] David Snyder, Daniel Garcia-Romero, Greg Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in *Proceedings of ICASSP*, 2018.

[17] David Snyder, "NIST SRE 2016 Xvector Recipe," `https://david-ryan-snyder.github.io/2017/10/04/model_sre16_v2.html`, 2017.

[18] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer, "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," in *Proceedings of SRE11 analysis workshop*, Atlanta, Dec. 2011.

[19] Kornel Laskowski and Jens Edlund, "A Snack implementation and Tcl/Tk Interface to the Fundamental Frequency Variation Spectrum Algorithm," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010.

[20] David Talkin, "A Robust Algorithm for Pitch Tracking (RAPT)," in *Speech Coding and Synthesis*, W. B. Kleijn and K. Paliwal, Eds., New York, 1995, Elsevier.

[21] Martin Karafiát, František Grézl, Karel Veselý, Mirko Hannemann, Igor Szőke, and Jan Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Interspeech 2014*, 2014, pp. 3002–3006.

[22] Pavel Matějka et al., "Neural network bottleneck features for language identification," in *IEEE Odyssey: The Speaker and Language Recognition Workshop*, Joensu, Finland, 2014.

[23] Pavel Matějka, Lukáš Burget, Petr Schwarz, and Jan Černocký, "Brno University of Technology System for NIST 2005 Language Recognition Evaluation," in *Proceedings of Odyssey 2006*, San Juan, PR, 2006.

[24] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, 2015, pp. 3214–3218.

[25] "National Institute of Standards and Technology," http://www.nist.gov/speech/tests/spk/index.htm.

[26] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson, "The Speakers in the Wild (SITW) Speaker Recognition Database," in *Interspeech 2016*, 2016, pp. 818–822.

[27] "The NIST year 2016 Speaker Recognition Evaluation Plan," `https://www.nist.gov/sites/default/files/documents/2016/10/\\07/sre16\_eval\_plan\_v1.3.pdf`, 2016.

[28] Mirco Ravanelli, Piergiorgio Svaizer, and Maurizio Omologo, "Realistic Multi-Microphone Data Simulation for Distant Speech Recognition," in *Interspeech 2016*, 2016, pp. 2786–2790.

[29] "Aachen Impulse Response Database," http://www.iks.rwth-aachen.de/en/research/tools-downloads/databases/aachen-impulse-response-database/.

[30] "C4DM (Center for Digital Music) RIR database," http://isophonics.net/content/room-impulse-response-data-set.

[31] R. Stewart and M. Sandler, "Database of omnidirectional and B-format room impulse responses," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 165–168.

[32] "Multichannel Acoustic Reverberation Database at York," http://www.commsp.ee.ic.ac.uk/ sap/resources/mardy-multichannel-acoustic-reverberation-database-at-york-database/.

[33] "OpenAir Impulse Response Database," http://www.openairlib.net/auralizationdb.

[34] "Reverb Challenge ," http://reverb2014.dereverberation.com/index.html.

[35] "RWCP Sound Scene Database," http://www.openslr.org/13/.

[36] Emanuël A.P. Habets, "Room Impulse Response Generator," https://www.audiolabs-erlangen.de/content/05-fau/professor/00-habets/05-software/01-rir-generator/rir_generator.pdf.

[37] Ronald M. Aarts, "A comparison of Some Loudness Measures for Loudspeaker Listening Tests," *J. Audio Eng. Soc*, vol. 40, no. 3, pp. 142–146, 1992, `http://www.extra.research.philips.com/hera/people/aarts/RMA_papers/aar92a.pdf`.

[38] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5220–5224.

[39] Daniel Garcia-Romero and Carol Y. Espy-Wilson, "Analysis of i-vector length normalization in Gaussian-PLDA speaker recognition systems," in *Proc. Interspeech*, 2011.

[40] P. Kenny, "Bayesian speaker verification with Heavy–Tailed Priors," keynote presentation, Proc. of Odyssey 2010, June 2010.

[41] Ladislav Mošner, Pavel Matějka, Ondřej Novotný, and Jan Černocký "Dereverberation and beamforming in far-field speaker recognition," in *Proceedings of ICASSP*, 2018.