# DEEP SPEAKER EMBEDDING LEARNING WITH MULTI-LEVEL POOLING FOR TEXT-INDEPENDENT SPEAKER VERIFICATION

*Yun Tang, Guohong Ding, Jing Huang, Xiaodong He, Bowen Zhou*

JD AI Research
675 East Middlefield Road
Mountain View, CA 94043, USA

## ABSTRACT

This paper aims to improve the widely used deep speaker embedding x-vector model. We propose the following improvements: (1) a hybrid neural network structure using both time delay neural network (TDNN) and long short-term memory neural networks (LSTM) to generate complementary speaker information at different levels; (2) a multi-level pooling strategy to collect speaker information from both TDNN and LSTM layers; (3) a regularization scheme on the speaker embedding extraction layer to make the extracted embeddings suitable for the following fusion step. The synergy of these improvements are shown on the NIST SRE 2016 eval test (with a 19% EER reduction) and SRE 2018 dev test (with a 9% EER reduction), as well as more than 10% DCF scores reduction on these two test sets over the x-vector baseline.

***Index Terms***— Speaker recognition, x-vector, multi-level pooling

## 1. INTRODUCTION

Speaker verification (SV)[1] is one of the key components for human machine interface and initially was widely used in person identification for security purposes. Nowadays with intelligent speech assistants such as Alexa, Google Home, Siri and Cortana being used in home environments as well as on smartphones, the demand for SV technology is rising. This is especially true for robust speaker verification in challenging acoustic conditions and different population of speakers. The speaker verification problem usually falls into two categories: text-dependent (TD) SV and text-independent (TI) SV. In the TD SV system, the transcriptions for the test utterances and enrollment utterances are the same and usually limited to a small set. In the TI SV system, there is no constraint on transcriptions for both test and enrollment utterances. Hence the TI case is more difficult than the TD case due to larger variations introduced by different utterance transcriptions and duration. In this study, we will focus on the more challenging TI speaker verification system.

Recently, more attention has been drawn to the use of deep neural networks (DNN) to generate speaker embedding representations [2] [3] [4]. These deep speaker embedding systems are shown to have large improvements over the i-vector [5] based methods, especially the recently proposed x-vector system with data augmentation [4]. The DNN based speaker embedding extraction system usually consists of three components: frame level feature processing, utterance (speaker) level feature processing and training loss. Frame level processing deals with local short span of acoustic features. It can be done via recurrent neural networks [2] or convolutional neural networks [3][6]. Utterance level processing forms speaker representation based on the frame level output. A pooling layer is used to gather frame level information to form utterance level representation. Methods such as statistical pooling [3], max pooling[7], attentive statistical pooling [8], multi-headed attentive statistical pooling [9] are popular choices. Cross entropy and triplet loss are two widely used training losses. Cross entropy based methods are focused on reducing the confusion for all speakers in the training data [3], while triplet loss based methods [10][11][12][13] are focused on increasing the margin between similar speakers.

In this paper, we propose a novel deep speaker embedding learning framework to improve upon the x-vector system. We first add LSTM layers to the x-vector's TDNN structure, because sequential modeling of an utterance would generate different speaker information from that of TDNN. In order to aggregate these different information, we propose a multi-level pooling strategy to fuse different information from the different frame level models, one from TDNN and one from LSTM. We also add a regularization term on the speaker embedding extraction layer in order to make the system output more suitable for the back-end processing. Overall, the proposed new speaker embedding system improves the equal error rate (EER) by 19% and the detection cost function(DCF)[1] score by 12%, compared to the previous x-vector baseline on the NIST SRE 2016 eval test. This is to our knowledge the lowest EERs (9.2% on Tagalog and 3.1% on Cantonese) on SRE 2016 eval test in publications so far.

The rest of this paper is organized as follows. Section 2 briefly describes prior work, including the baseline x-vector system and related work. The proposed new system is intro-

duced in Section 3 on gathering speaker information from different modeling levels. The experimental set up, results and analysis are described in Section 4. Finally, the conclusion is given at Section 5.

## 2. PRIOR WORK

### 2.1. Baseline x-vector System

Figure 1 depicts the deep neural network configuration for x-vector system [3][4]. The blue, yellow and green blocks represent the frame level, utterance level and training loss used in the system training. Black arrows indicate a ReLU activation function. Batch normalization layers are added in two adjacent layers, while the red arrows indicate there is no nonlinear mapping added between layers.

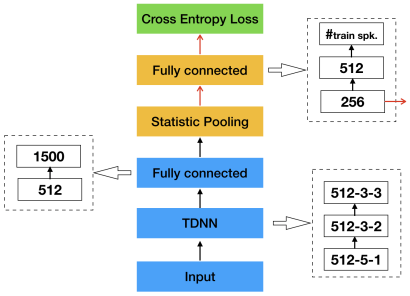


**Fig. 1**. Model structure of baseline x-vector system.

The frame level model consists of three TDNN layers to extract speaker information at the frame level. X-Y-Z in the block represents the number of filters, filter window size and dilation number at this layer. At the utterance level, high order stats pooling methods such statistical pooling or attentive statistical pooling can be used. The output (concatenation of the mean and standard deviation vectors) is fed into another three fully connected layers. Speaker embeddings are then extracted from the output of the first linear projection layer.

Cross entropy loss is used to train the system and reduce the confusion between all speakers in the training set. Once speaker embeddings are extracted, LDA and PLDA [14] are used as back-end scoring, as in the standard Kaldi x-vector recipe.

### 2.2. Related work

Combining CNN or TDNN with LSTM is proved effective in automatic speech recognition (ASR) tasks. Sainath et al. [15] proposed to stack CNN, LSTM and DNN sequentially for speech recognition and it shows superior results than using CNN, LSTM or DNN alone. Peddinti et al. [16] conducted experiments to compare the stack order of TDNN and LSTM and found interleaving of TDNN layers with LSTM layers could be more effective than simple stacking strategy.

Pooling module is the key component to bridge frame level features to speaker representation. High order statistical pooling [3] shows better performance than simple pooling method like average or maximum pooling. Okabe et al. [8] proposed parametric based attentive pooling to give different importance for each frame feature. Zhu et al. [9] further extended it to multiple heads to generate utterance representation from different views. The multiple-level pooling proposed in this work is focused on the way to gather pooling features instead of pooling module itself.

## 3. DEEP SPEAKER EMBEDDING WITH MULTI-LEVEL POOLING

Figure 2 shows the proposed new deep speaker embedding training used in this study. In order to generate complimentary information, a hybrid structure with both TDNN and LSTM layers is proposed in this framework. The TDNN layer would focus on the local feature representation, while the LSTM layer would consider global and sequential information from the whole utterance. The multi-level pooling from the TDNN and LSTM layers generate different representations on the utterance level. This would help to gather speaker information from different spaces and generate a comprehensive speaker representation. This would also help to pass error signals back to earlier layers which, in turn, helps alleviate the vanishing gradient problem. The outputs of different pooling layers are concatenated and fused into the following fully connected layers. Even though residual networks with very deep neural network structures [17][6] are popular for computer vision tasks, we haven't seen good improvements from very deep structures, as shown in [7]. Therefore we use the same number of TDNN layers as in x-vector model.

The x-vector system is not an end-to-end SV system. The extracted speaker embeddings are passed to the back-end classifiers of LDA and PLDA, which are both linear transforms trained with different objective functions. Thus there is a mismatch of training loss and LDA/PLDA training objectives, as noticed in [18]. In order to make the embedding output suitable for the backend, a regularization term is introduced by applying a constraint on the norm of the embedding layer output.

$$\mathcal{L} = -\sum_{i=1}^{M} log \frac{\exp^{w_{c_i}^T x_i + b_{c_i}}}{\sum_j^N \exp^{w_j^T x_i + b_j}} + \lambda ||z_i||_2 \qquad (1)$$

In the above equation, $c_i$ is the speaker index from training sample $i$, $w_j$ corresponds to j-th column of $W \in \mathcal{R}^{dXN}$ the last linear projection layer before softmax, $b_j$ is the bias term, $N$ and $M$ are the number of speakers and samples in the training data respectively. $x_i$ is the output tensor for the 2nd fully connected layer in the utterance level model, $z_i$ the output tensor of the 1st fully connected layer without ReLU non-

linearity and batch normalization. Note, the regularization is applied to the output of the speaker embedding extraction layer directly instead of the weights in the neural network.

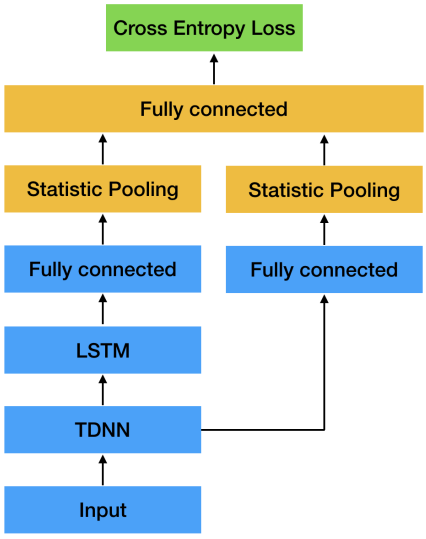


**Fig. 2**. Deep speaker embedding with multi-level pooling (Model $MP$ in Table 1)

## 4. EXPERIMENTS AND RESULTS

### 4.1. Data set

Both the baseline x-vector and our proposed deep speaker embedding systems are evaluated on the NIST SRE16 eval test and SRE18 dev test. The SRE16 test set includes Tagalog and Cantonese telephone speech. For the SRE18 dataset, only Call My Net2 (CMN2) portion of development data set is used in our test. The CMN2 data is composed of both PSTN and VOIP data collected outside North America, spoken in Tunisian Arabic. The amount of enrollment data is around 60 seconds per speaker and the duration test data is ranging approximately from 10 seconds to 60 seconds. In both SRE16 and SRE18, the tasks aim to deal with domain mismatch between the development and test data, including channel, noise condition and language mismatches.

In order to handle multiple domain data, a large training dataset is formed from SRE18 allowed, publicly available, data sources: SRE data from years 2004 to 2006, 2008, 2010; all Switchboard data; all Fisher data and all Voxceleb data, with a total amount of $13,564$ hours. This dataset consists of both telephone and microphone speech from total $20,803$ speakers. The model is trained on the 8kHz data. The features include 40 dimensional filterbanks and 3 dimensional pitch features. The feature has frame-lengths of 25ms, mean-normalized over a sliding window of up to 3 seconds. The Kaldi energy based SAD is used to filter out non-speech frames.

We follow the same data augmentation procedure conducted in [4] on the training data to alleviate the noise condition mismatch. The data augmentation strategy includes both adding additive noises and reverberation data.

In both SRE16 and SRE18 NIST provided tens of hours unlabelled but matched data for development. For back-end scoring, an LDA projects speaker embeddings from 256 to 150, estimated from SRE training data. The same SRE training data is also used to estimate the PLDA classifier. An extra PLDA adaptation [19] is conducted on the unlabelled development data to reduce training/testing mismatch.

### 4.2. Experimental Setup

The configuration of the baseline x-vector system is described in Section 2.1. One thing to mention is that a smaller neuron number is used in the speaker embedding extraction layer, say 256 instead of 512, as shown in Figure 1. This benefits the following backend processing step such as LDA and PLDA estimation to give better evaluation results.

The configurations of the three systems to be examined as well as the baseline system are compared in Table 1:

**Table 1**. Experimental model configures

| Model Name | Model Configurations |
|---|---|
| x-vector | TDNN1-TDNN2-TDNN3-P |
| A | TDNN1-P-TDNN2-P-TDNN3-P |
| B | TDNN1-TDNN2-TDNN3-LSTM-P |
| MP | TDNN1-TDNN2-TDNN3-P-LSTM-P |

In the above table, TDNN$i$ represents the $ith$ TDNN layer in the frame level model, $P$ represents a pooling operation applied to the output of the previous layer. Each pooling operation includes two fully connected layers with statistical pooling afterwards. Model $MP$ (Fig. 2) is the proposed system and x-vector is the baseline system in this study. Models $A$ and $B$ are model configurations between $MP$ and the x-vector baseline. These two models are used to understand the differences between regular pooling and our multi-level pooling.

Compared with x-vector and model $A$, model $B$ and $MP$ each have one bidirectional LSTM layer after TDNN blocks in the frame level model. The number of neurons is 512. x-vector and model $B$ are single pooling based systems while model $A$ and $MP$ are multiple pooling based systems. In the multiple pooling based systems ($A$ and $MP$), different statistical pooling outputs will be concatenated together before sending to the following fully connected layers in the utterance level model. In order to keep the size of pooling output across different systems the same, the number of neurons in

the 2nd fully connected layer (before pooling module) will be changed from 1500 to 500 and 750 for model $B$ and $MP$ respectively.

The system is implemented with PyTorch and optimized with stochastic gradient descent (SGD) using momentum (0.9), with a mini-batch size of 256.

### 4.3. Experimental Results

We report results in terms of EER and the minimum DCF at $P_{target} = 0.01$ and $P_{target} = 0.005$. No score normalization, e.g., t-norm or s-norm [20], is applied in this study.

**Table 2**. Results on SRE16 eval test.

| Model | $\lambda = 0$ | | | $\lambda = 0.001$ | | |
|---|---|---|---|---|---|---|
| | Pooled | Tag. | Can. | Pooled | Tag. | Can. |
| x-vector | 7.61 | 10.98 | 3.95 | 6.90 | 10.20 | 3.50 |
| A | 8.17 | 11.78 | 4.45 | 7.51 | 10.93 | 4.06 |
| B | **6.64** | 9.80 | **3.40** | 6.84 | 9.83 | 3.82 |
| MP | 6.68 | **9.74** | 3.51 | **6.13** | **9.18** | **3.13** |

The EERs on the SRE16 test set are shown in Table 2. The left side of the table are results without regularization and the right side with regularization $\lambda = 0.001$.

We first examine our results when no regularization is applied. Our results from the left side of Table 1 shows that with no regularization, there is no gain with pooling comparing Model $A$ to the baseline x-vector. Adding a bidirectional LSTM in Model $B$ helps reducing the EER by 12.7% relative to the baseline.

Now we move on to our results with regularization (the right side of Table 2). Compared to the results on the left side, most models achieve significant EER reduction. Our proposed model ($MP$) achieves the best results in this test set. These results show that multiple pooling operations is only beneficial if the pooling information comes from different sources. In our proposed model $MP$, one pooling operation is conducted after the TDNN layers, which focus on local information, and the 2nd pooling operation is conducted after the LSTM layer, which extracts sequential information.

Adding a regularization term on the norm of the embedding output is useful for most models, especially for models with multiple pooling. This reduces the range of the norm of the output embedding vectors, from 500 to 10. This makes the speaker embeddings more numerically stable for the backend scoring. Reducing the norm of the output embeddings also may condense features from different levels of the model, into the same numerical range, which helps them to be fused properly.

Overall, comparing the result from our proposed model $MP$ with regularization, against the baseline x-vector model, we observe a relative 19% EER reduction. The corresponding DCF scores are reported in Table 3, for different $P_{target}$ values. There is a 14.6% and 12.3% DCF score reduction compared with the baseline for $P_{target}$ = 0.01 and 0.005 respectively.

**Table 3**. DCF scores for SRE16 (pooled) test set

| model | $\lambda = 0$ | | $\lambda = 0.001$ | |
|---|---|---|---|---|
| | $p = 0.01$ | $p = 0.005$ | $p = 0.01$ | $p = 0.005$ |
| x-vector | 0.593 | 0.651 | 0.594 | 0.673 |
| B | 0.581 | 0.656 | 0.525 | 0.586 |
| MP | 0.567 | 0.632 | **0.506** | **0.571** |

**Table 4**. Evaluation results on the SRE18 (CMN2) dev set

| $\lambda$ | model | EER | $DCF(0.01)$ | $DCF(0.005)$ |
|---|---|---|---|---|
| 0.0 | x-vector | 7.29 | 0.593 | 0.651 |
| | B | 7.90 | 0.581 | 0.656 |
| | MP | 7.16 | 0.567 | 0.632 |
| 0.001 | x-vector | 7.46 | 0.594 | 0.673 |
| | B | 7.77 | 0.525 | 0.586 |
| | MP | **6.61** | **0.506** | **0.571** |

In Table 4, results on the SRE18 CMN2 development set are presented. Similar to SRE16 results, model $MP$ achieves the best result: 9.3% EER reduction and 12% to 14% DCF score reduction are achieved when regularization is applied.

## 5. CONCLUSIONS

In this study, we propose a novel pooling strategy for gathering speaker information from different levels of the model. Our three main contributions are as follows: (1) A hybrid model structure including both TDNN and LSTM is employed to generate complimentary information. Output from TDNN blocks focuses on local information and LSTM layer emphasizes on global and sequential information generated from the whole utterance. (2) Different representation extract from the hybrid model are pooled at multiple levels and combined to obtain a robust speaker representation. (3) A regularization is applied to the output of the embedding extraction layer, which significantly reduces the norm of extracted embedding and keeps it in a reasonable value range as well as keeping the embedding more numerically stable for the following backend scoring. The synergy of these improvements are shown on the NIST SRE 2016 eval test (with a 19% EER reduction) and SRE 2018 dev test (with a 9% EER reduction), as well as more than 10% DCF scores reduction on these two test sets over the x-vector baseline.

## 6. REFERENCES


[1] John HL Hansen and Taufiq Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal processing magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[2] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[3] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[4] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP*, 2018.

[5] Najim Dehak, Patrick Kenny, Reda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.

[6] Na Li, Deyi Tuo, Dan Su, Zhifeng Li, and Dong Yu, "Deep discriminative embeddings for duration robust speaker verification," *Proc. Interspeech 2018*, pp. 2262–2266, 2018.

[7] Sergey Novoselov, Andrey Shulipa, Ivan Kremnev, Alexandr Kozlov, and Vadim Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," in *Odyssey*, 2018.

[8] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[9] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3573–3577, 2018.

[10] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[11] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez-Moreno, "Generalized end-to-end loss for speaker verification," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883, 2018.

[12] Sergey Novoselov, Vadim Shchemelinin, Andrey Shulipa, Alexandr Kozlov, and Ivan Kremnev, "Triplet loss based cosine similarity metric learning for text-independent speaker recognition," *Proc. Interspeech 2018*, pp. 2242–2246, 2018.

[13] Zili Huang, Shuai Wang, and Kai Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3623–3627, 2018.

[14] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

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

[16] Vijayaditya Peddinti, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[18] Lantian Li, Zhiyuan Tang, Dong Wang, and Thomas Fang Zheng, "Full-info training for deep speaker feature learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5369–5373.

[19] Md Jahangir Alam, Gautam Bhattacharya, and Patrick Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 176–180.

[20] Najim Dehak, Reda Dehak, James R Glass, Douglas A Reynolds, and Patrick Kenny, "Cosine similarity scoring without score normalization techniques.," in *Odyssey*, 2010, p. 15.