Attentional triplet neural networks for text-dependent speaker verification

Yiyang Wang and Wenjun Zhang

1 Department of microelectronics, Tsinghua University, Beijing, 100084, China

wyiyang1259@163.com

Abstract. Deep Neural Networks (DNNs) have been widely used in speech processing and show great performance on a range of tasks, such as speech recognition, machine translation, and speaker verification. In this paper we propose a new type of DNN model for text-dependent speaker verification. The frame-level features, being extracted by DNN, are usually equally weighted and aggregated (or averaged) to compute an utterance-level speaker representation. We combine Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to extract frame-level features, then provide every frame with different weights produced by attention mechanism, so that utterance-level speaker representation can be generated by weighted averaging frame-level features. We explore different generating methods on the attention weights. Besides, attention mechanism can also be used to align temporal information between enrollment and evaluation utterance. Triplet loss function is used to optimize our models, requiring inputs group in triplet style. Ultimately, results of experiment on RSR2015 database show that our attention-based model outperforms various baseline models.

1. Introduction

Speaker verification is the process of verifying whether an evaluation utterance belongs to the specific speaker who has been claimed. Based on the text content, speaker verification can be categorized into text-dependent and text-independent tasks [1]. In text-dependent systems, the lexicon of the utterance is constrained to specific phrase and text-independent system is on the contrary. Compared with text-independent task, text-dependent speaker verification usually achieves more reliable results because of the constraint of the phonetic variability. Due to different shape of vocal tract, each person has unique acoustic feature called “voiceprint”, which is also the biological basic of speaker verification. Similar to fingerprint, voiceprint can also be used to lock personal equipment or mobile applications. Voiceprint lock is more convenient than fingerprint lock in that we don’t need to press a button when we want to wake device up. Besides, the security should be guaranteed. For example, smart devices would be awoken only by specific utterance of certain person, where text-dependent speaker verification is necessary for preventing unauthorized usage.

Typical models built for speaker verification include GMM-SVM [2], GMM-UBM [3], and i-vector/PLDA. Based on these models, the best feature’s form and robust way of combining them have also been investigated [4]. Recently, with great success of applying Deep Neural Networks (DNNs) in speech signal processing, more efforts have been focusing on text-dependent speaker verification based on deep learning [5]. In [6], a speaker discriminative DNN comprises of fully connected (FC) layers is trained to extract frame-level features from the last hidden layer. These features are aggregated with equal weights into utterance-level feature which is called d-vector. Since Recurrent Neural Networks
(RNNs) are proved to have advantages in processing speech signal over DNNs based on FC layers, they have been widely used in speaker verification models [7]. Although dedicated to image processing originally, Convolutional Neural Networks (CNNs) are also applied in speaker verification tasks since they are able to learn speaker characteristics from different aspects of the input sequence [8] [9]. As a branch of speech recognition, speaker recognition can also adopt some techniques of it [10] [11] [12]. Recently attention mechanism has gained popularity in training neural networks as a result of the ability to emphasize the most relevant elements of the input sequence [13] [14]. In other words, frame-level features extracted by DNNs are smartly combined with different weights achieved from an attention mechanism model to generate an utterance-level speaker representation, instead of just equally weighting and averaging all the frames [15] [16].

Several models have been proposed for text-dependent speaker verification. However, their model size is not appropriate when applied to mobile devices. In this paper, we propose a novel small footprint triplet style text-dependent speaker verification neural network model using attention mechanism in special forms. Specifically, alignment algorithm based on attention mechanism performs well without increasing too many parameters. Considering cosine similarity between evaluation and enrollment utterance, we use the triplet loss function like in FaceNet [17].

This paper is organized as follows. Section 2 provides a brief overview of speaker verification in general. Section 3 presents the proposed methods for speaker verification, which include our basic model, the attention algorithm and the objective function. Then, in Section 4, we show the experimental results on the RSR2015 database. Finally, Section 5 concludes this paper and suggests directions for the future work.

2. Speaker Verification Protocol

Speaker verification has relatively fixed procedures. Its protocol can be summarized as three steps: training, enrollment, and evaluation.

Training At first, we build and train a model to represent speaker’s identity. In most cases, especially when the model is based on DNNs, it is used to extract specific features. So, the destination of training step is improving the model’s ability to extract speaker features more precisely. Due to superior ability of learning, DNNs usually achieve better results when trained with adequate data.

Enrollment In this stage, a certain speaker provides a few utterances which are used to estimate speaker representations. Taking speech rate and environmental noise into consideration, these representations are averaged to generate final speaker model.

Evaluation We evaluate trained model (feature extractor) in evaluation stage. A scoring function is used to achieve scores, estimating degree of relevance between the evaluation utterance and that enrollment speaker. We accept that the utterance comes from the certain speaker if the score exceeds the pre-defined threshold, and reject otherwise. Two types of error can occur when we make the decision: false accept and false reject. The false accept rate (FAR) and false reject rate (FRR) depend on the threshold. To take both convenience and security into account, we adjust the threshold to make the two rates equal. This rate value is called equal error rate (EER), the common index evaluating speaker verification systems.

In this paper, we use cosine similarity between the speaker representation \( f(X) \) of an evaluation utterance \( X \) and the speaker model \( m_{spk} \) as scoring function:

\[
S(X, spk) = \frac{f(X) \cdot m_{spk}}{||f(X)|| \cdot ||m_{spk}||}
\]

3. Proposed methods

3.1. Preprocessing

The enrollment and evaluation utterances are sampled at 16 kHz and recorded for shorter than 3 seconds. Each utterance is zero-padded in the end to maintain 3 seconds long. Then, 40-dimension log-mel filter-
bank using 25 ms analysis window with 10 ms overlap are extracted. These end up as log-mel filter-bank of size $300 \times 40$ for a 3-second utterance.

3.2. Basic architecture
Recently, CNNs have been widely used in speech processing tasks due to the ability of capturing energy modulation patterns across time and frequency when applied to spectrogram-like input [8]. Besides, RNNs are acknowledged to have advantages in processing sequence because they can memorize previous information and take them into current calculation. We use combination of CNN and long short-term memory (LSTM) RNN as primary architecture and try to improve it by adopting various methods introduced in the following parts. Average Pooling layer is used to transform frame-level features to utterance-level representation by simply averaging frame-level features. The structure is shown in figure 1(a). The speaker representation will be fed into scoring function to calculate similarity score.

3.3. Triplet loss
Based on cosine similarity scoring function in equation (1), we use triplet loss in training stage. As shown in figure 2, triplet loss takes in three samples as inputs: an anchor (a certain utterance from a specific speaker), a positive example (another utterance with the same phrase as the anchor from the same speaker), a negative example (another utterance with different phrase or from different speaker). In other words, a
sample will be decided as “negative example” on condition that either different phrase content or different speaker, which is determined by intrinsic property of text-dependent speaker verification task. In training stage, we seek to update model parameters so that the cosine similarity between the anchor and the positive example is larger than the cosine similarity between the anchor and the negative example.

![Diagram](Image)

**Figure 2.** The Triplet loss based on cosine similarity.

In brief, they are constrained by this formula [18]:

$$s^p - s^n > m$$

where $s^p$ is the cosine similarity between the anchor $a$ and the positive example $p$, and $s^n$ is the cosine similarity between the anchor $a$ and the negative example $n$. We impose a minimum margin $m$ between those similarities. The cost function for $N$ triplets can be written as

$$L_{triplet} = \sum_{i=1}^{N} \left[ s^p - s^n + m \right]_+$$

where the operator $[x]_+ = \max(x, 0)$.

### 3.4. Attention mechanism

The attention mechanism has lately become popular because it has been successfully applied in several tasks, including speech recognition, machine translation, and image captioning due to their ability to emphasize the most relevant elements of the input sequence. In this paper, we utilize attention mechanism in two ways.

#### 3.4.1. CNN-based attention layer

In our basic model, we directly average LSTM outputs at all frames. Alternatively, we could learn scalar scores as weights for every frame:

$$e_t = f(d_t), \quad t = 1, L, T.$$  \hspace{1cm} (4)

Then we can use $e_t$ to compute the normalized weights $\alpha_t$:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)}$$

such that $\sum_{t=1}^{T} \alpha_t = 1$. Finally, we achieve utterance-level representation by weighted averaging frame-level features exported from LSTM layer:

$$c_t = \sum_{t=1}^{T} \alpha_t h_t$$

We choose FBank features as $d_t$ in equation (4) and LSTM outputs as $h_t$ in equation (6), with combination of CNN and fully connected (FC) layer as function in equation (4). Figure 1(b) shows the self-attention layer.
3.4.2. Alignment algorithm. The position of critical information of the evaluation is different from that of the enrollment in most cases, which is not friendly to cosine similarity scoring function. We adopt attention mechanism to align evaluation and corresponding enrollment utterance [19]. Figure 1(c) shows the proposed alignment method based on the attention mechanism. $h_s$ and $h_t$ are the evaluation and enrollment frame-level feature sequences exported from LSTM layer in Figure 1(a). We can find the frame-level relevance between the two sequences by defining an alignment score between the $s$-th frame of the evaluation and the $t$-th frame of the enrollment:

$$\text{score}(h_s, h_t) = h_s^i \cdot h_t^i$$

(7)

Then the normalized weights can be generated by a softmax function as:

$$a_t(s) = \frac{\exp(\text{score}(h_s, h_t))}{\sum_t \exp(\text{score}(h_s, h_t))}$$

(8)

where the size of $a_t(s)$ equals the number of time steps. With the alignment vector as weights, we can derive a context vector sequence $c_t$ that captures evaluation side information that is the most enrollment-relevant by computing the weighted average over all the evaluation frame-level speaker vectors:

$$c_t = \sum_s a_t(s) \cdot h_s$$

(9)

Finally, we average the vector $c_t$ across all time steps using average pooling layer in figure 1(a).

For example, the first frame of the enrollment is used to calculate dot product with every frame of the evaluation. The results are normalized with softmax function so we can achieve weights $a_t(s) = [a_t(1), a_t(2), \ldots, a_t(T)]^T$ and the first frame of $c_t$ through equation (9). The rest frames of $c_t$ can be achieved using similar procedures.

We also use alignment mechanism in training stage, where the enrollment can be replaced by the anchor $a$ and the evaluation by the positive example $p$ or negative example $n$.

3.5. Pruning
Not every frame of the evaluation is enrollment-relevant. Some frames may even make adverse contributions to estimating relevance between the evaluation and the enrollment. As a result, it is essential to prune the small weights $a_t(s)$ in (8): all weights below a threshold are removed (set to zero) [20]. This method can also reduce operations.

4. Experiments
4.1. Database
We use the RSR2015 database, a text-dependent speaker verification corpus, to evaluate the effectiveness of the proposed systems. RSR2015 involved 300 speakers (157 male, 143 female) and for each speaker, there were 3 enrollment sessions of 73 utterances each and 6 verification sessions of 73 utterances each, for a total of 657 utterances in 9 sessions per speaker. All sessions were recorded using portable devices (handphones and tablets) [21]. Part 1, 2, and 3 of RSR2015 are used for training and evaluation. We organize training data as triplet groups described in Section 3. To prevent over-fitting, we also use dropout strategy. We experiment as procedures described in section 2, using Keras to build our models. After evaluation, we make the detection error trade-offs (DET) curve to find the equal error rate (EER).

4.2. Methods for comparison
We compare the following methods to evaluate the effectiveness of the proposed approach. The d-vector model and LSTM-based end2end model are used as baseline.

- **D-vector model** [6]: In this method, we trained a DNN with four FC layers and 256 nodes per layer. The first two layers do not use dropout while the last two layers drop 50 percent of
activations. Average pooling layer is next to the fourth FC layer and the softmax function is used as activation in the output layer. In the enrollment and evaluation stage, the output layer is removed. We use the output from the average pooling layer to compute the cosine similarity.

- **LSTM-based end2end model [7]**: A LSTM layer with 256 nodes is used to extract features in this approach. Cosine similarity function is embedded in this model, instead of just scoring in evaluation stage. After computed with output from LSTM, cosine similarity $S(X, spk)$ is fed to a logistic regression including a linear layer with a bias. The architecture is optimized using the binary cross-entropy loss.

- **Basic triplet model** proposed in Section 3 as figure 1(a).
- **CNN-based attention model**.
- **Basic triplet model with alignment mechanism**.
- **Basic triplet model with alignment mechanism and pruning method**.

### 4.3. Experimental results and analysis

For the last three models, we first initialized all the parameters stochastically for training. However, it seems that the triplet-CNN attention model did not converge and the other two models showed worse results than basic triplet model. Then we initialized parameters with trained basic triplet model and tried another two methods for training: freezing parameters of the first two layers before average pooling layer, and without freezing any parameter. Table 1 shows these results using EER(equal error rate) as evaluation metric.

| Configuration                        | (a)  | (b)  | (c)  |
|--------------------------------------|------|------|------|
| basic triplet model                  | 1.54%| -    | -    |
| triplet-CNN attention                | -    | 1.25%| 1.35%|
| triplet-alignment                    | 2.12%| 1.38%| 1.22%|
| triplet-alignment-pruning            | 2.04%| 1.46%| 1.24%|

It is observed that loading parameters of the trained basic triplet model is necessary for these three models. The output of CNN attention layer is multiplied with features extracted from LSTM layer, so it is difficult to optimize them at the same time using stochastic gradient descent algorithm. Maybe the trained model can provide a relatively explicit training route for them. Besides, the triplet-CNN model benefits more from freezing the first two layers which are the critical parts for extracting features. When all the parameters are trained jointly, CNN-based attention layer is hard to coordinate with the first two layers if they are not frozen. Due to these reasons, models trained with strategy (a) converge slowly or to nonoptimal directions.
### Table 2. EER and model size comparisons.

| Configuration                | Model Size | EER  |
|-----------------------------|------------|------|
| d-vector                    | 164.7k     | 4.68%|
| LSTM-based model            | 369.9k     | 4.20%|
| basic triplet model         | 451.8k     | 1.54%|
| triplet-CNN attention       | 622.1k     | 1.25%|
| triplet-alignment           | 451.8k     | 1.22%|
| triplet-alignment-pruning   | 451.8k     | 1.24%|

The best results and model size comparisons among the proposed and baseline models are shown in table 2. Our proposed methods are italic. For the consistency of conditions, we set number of enrollments as one for all the models. Our basic triplet model shows better results than d-vector model and LSTM-based model, suggesting that RNNs can achieve better results in speech processing tasks when they are combined with CNNs. Increasing the number of enrollments may narrow the gap between them. This indicates the performance of d-vector model is limited by the relatively smaller training set, while the triplet model learns speaker features in triplet groups which are not directly related to absolute speaker identities.

The EER decreases from 1.54% to 1.25% (18.8% relative reduction) after the attention module is inserted into the triplet model. This suggests that our attention module variant is helpful despite increasing parameters to some extent. One interesting finding is that the triplet-alignment model performs slightly better than the triplet-CNN attention model with fewer parameters, just at the expense of increasing a few multiply and add operations. This indicates the effectiveness of the alignment mechanism based on attention theory.

Pruning on the triplet-alignment model is based on comparing weights with a specific threshold. As a result, it may prune several contributory relevant weights for alignment. This is also the reason why the EER increases slightly (by 0.02%) after pruning, although it decreases arithmetical operations as expected.

### 5. Conclusion

In this paper, we propose a triplet model for text-dependent speaker verification and introduce attention mechanism to improve it. Experimental results show that CNNs are important for text-dependent speaker verification task. The CNN-based attention model and triplet-alignment model achieve similar results while the former needs more parameters in a large scale. Compared with various baselines, the triplet-alignment model shows significant improvement. This suggests that evaluation-enrollment temporal alignment mechanism really works as expected. Pruning decreases operations at the expense of slightly increasing EER. For future work, we will apply pruning to the whole neural networks for the purpose of compression.

### 6. References

[1] J. P. Campbell, “Speaker recognition: a tutorial,” in Proc. of the IEEE, vol. 85, no. 9, pp. 1437–62, Sept. 1997.

[2] W. M. Campbell, D. E. Sturim and D. A. Reynolds, “Support vector machines using GMM supervectors for speaker verification,” in IEEE Signal Processing Letters, vol. 13, no. 5, pp. 308–11, May 2006.

[3] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, “Speaker verification using adapted gaussian mixture models,” Digital signal process., vol. 10, no. 1-3, pp. 19–41, 2000.

[4] Smarajit Bose, Amita Pal, Anish Mukherjee, and Debasmita Das, “Robust Speaker Identification Using Fusion of Features and Classifiers,” International Journal of Machine Learning and Computing, vol. 7, no. 5, pp. 133–138, 2017.
[5] Amna Irum and Ahmad Salman, “Speaker Verification Using Deep Neural Networks: A Review,” *International Journal of Machine Learning and Computing*, vol. 9, no. 1, pp. 20–25, 2019.

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” *2014 IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, 2014, pp. 4052–6.

[7] G. Heigold, I. Moreno, S. Bengio and N. Shazeer, “End-to-end text-dependent speaker verification,” *2016 IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, 2016, pp. 5115–9.

[8] N. N. An, N. Q. Thanh and Y. Liu, “Deep CNNs With Self-Attention for Speaker Identification,” in *IEEE Access*, vol. 7, pp. 85327–37, 2019.

[9] Supaporn Bunrit, Thuttaphol Inkian, Nittaya Kerdpasop, and Kittisak Kerdpasop, “Text Independent Speaker Identification Using Deep Learning Model of Convolution Neural Network,” *International Journal of Machine Learning and Computing*, vol. 9, no. 2, pp. 143–148, 2019.

[10] Thum Wei Seong, M. Z. Ibrahim, and D. J. Mulvaney, “WADA-W: A Modified WADA SNR Estimator for Audio-Visual Speech Recognition,” *International Journal of Machine Learning and Computing*, vol. 9, no. 4, pp. 446–451, 2019.

[11] Shaun V. Ault, Rene J. Perez, Chloe A. Kimble, and Jin Wang, “On Speech Recognition Algorithms,” *International Journal of Machine Learning and Computing*, vol. 8, no. 6, pp. 518–523, 2018.

[12] Ferda Ernawan and Nur Azman Abu, “Efficient Discrete Tchebichef on Spectrum Analysis of Speech Recognition”, *International Journal of Machine Learning and Computing*, vol. 1, no. 1, pp. 1–6, 2011.

[13] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, “Attention-based models for speech recognition,” in *Advances in Neural Information Processing Systems*, 2015, pp. 577–85.

[14] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.

[15] F. A. Rezaur rahman Chowdhury, Q. Wang, I. L. Moreno and L. Wan, “Attention-Based Models for Text-Dependent Speaker Verification,” *2018 IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, 2018, pp. 5359–63.

[16] S. Zhang, Z. Chen, Y. Zhao, J. Li and Y. Gong, “End-to-End attention-based text-dependent speaker verification,” *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, 2016, pp. 171–8.

[17] F. Schroff, D. Kalenichenko and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 815–23.

[18] 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.

[19] Y. Zhang, M. Yu, N. Li, C. Yu, J. Cui and D. Yu, “Seq2Seq Attentional Siamese Neural Networks for Text-dependent Speaker Verification,” *2019 IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom, 2019, pp. 6131–35.

[20] Song Han, Huizi Mao, and William J Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding.” *International Conf. on Learning Representations*, 2016.

[21] Anthony Larcher, Kong Aik Lee, Bin Ma, Haizhou Li, “Text-dependent speaker verification: Classifiers, databases and RSR2015”. *Speech Communication*, vol. 60, pp. 56–77, 2014.