Triplet Based Embedding Distance and Similarity Learning for Text-independent Speaker Verification

Zongze Ren, Zhiyong Chen and Shugong Xu
Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai, China
E-mail: {zongzeren, bicbrv, shugong}@shu.edu.cn

***Abstract*—Speaker embeddings become growing popular in the text-independent speaker verification task. In this paper, we propose two improvements during the training stage. The improvements are both based on triplet because the training stage and the evaluation stage of the baseline x-vector system focus on different aims. Firstly, we introduce triplet loss for optimizing the Euclidean distances between embeddings while minimizing the multi-class cross entropy loss. Secondly, we design an embedding similarity measurement network for controlling the similarity between the two selected embeddings. We further jointly train the two new methods with the original network and achieve state-of-the-art. The multi-task training synergies are shown with a 9% reduction equal error rate (EER) and detected cost function (DCF) on the 2016 NIST Speaker Recognition Evaluation (SRE) Test Set.**

Index Terms: speaker verification, deep neural network, similarity learning

I. Introduction

Speaker verification (SV) is a problem to give a decision that whether two utterances said by one person and can be defined as variable-length sequence classification task at utterance-level. The text-independent speaker verification (TI-SV) task is more challenging because it does not have any lexicon or pronunciation constraints, the corpora also do not have transcript labels for training but only speaker information.

Previously, statistical models are extensively used for solving TI-SV problem, i-vector [1] based system is a representative of this type of approach and has achieved significant success in modelling speaker identity and channel variability in its space. Deep neural network (DNN) attracts more attention for TI-SV task recently. Frame acoustic sequences are extracted from raw audio signals and then stacked as the input of DNN. [2] uses several fully connected layers and the output number of the last layer corresponds to the number of speakers in the training processing. Because predictions of frame-level [2] are not accurate enough, utterance-level representatives are considered by introducing several pooling operations. [3] uses statistical pooling of concatenating the average and standard deviation, called x-vector, which is recognized as a state-of-the-art solution. Other pooling methods such as self-attentive pooling [4], attentive statistical pooling [5] and high-order statistical pooling [6] also show their advantages.

On the other hand, part of the knowledge involved in the training data is used to learn a classifier that will be ultimately thrown away in the verification tasks. Triplet is a common way, FaceNet [7] firstly uses triplet loss in face-recognition task, the loss function minimizes the distance between an anchor and a positive while maximizes the distance between the anchor and a negative. [8], [9], [10], [11] use triplet loss in TI-SV task directly. However, systems only using single triplet loss are hard to converge in engineering. [12] uses triplet loss to finetune the softmax pre-trained network, making it easy to train in reality. After selecting a triplet, cosine similarity metric learning (CSML) [13] is proposed to train a metric for back-end scoring to replace traditional LDA-PLDA. Furthermore, [14] proposes a generalized end-to-end (GE2E) loss, which constructs a special batch to train speaker verification models more efficiently. And [15] trains the network architecture under the joint supervision of softmax loss and center loss. Similarity task is a typical problem in natural language processing (NLP), there is no inherent ordering of the two sentences being compared. [16], [17] concatenate the two utterance embeddings to contain both possible sentence orderings as the input of DNN.

Inspired by all these works, we put forward two methods based on triplet training while optimizing the multi-class classification target and test these two methods on Speaker Recognition Evaluation 2016 (SRE16) test set. Overall, proposed systems outperform the baseline system and here are our main contributions: (1) We choose a popular speaker verification network, x-vector as the baseline and implement Euclidean distances of a triplet, meaning that we minimize the distances between the anchor embedding and positive embedding while maximizes the distance between the anchor embedding and negative embedding, which reduces EER a lot. (2) We concatenate two embeddings of a triplet and feed them to a second DNN, named embedding similarity measurement network to measure the similarity between them, which makes a significant reduction of DCF by traditional LDA-PLDA backend scoring. (3) We jointly train the two methods with the original softmax-loss and achieve better performance. This idea of adding more constraints can extend to other fields.

The remainder of this paper is organized as follows. Section II reviews the x-vector structure of TI-SV tasks. In section III, we describe some details of the proposed improvements. The results of experiments and analysis are shown in Section IV. Lastly, we give the conclusion in Section V.

II. Baseline x-vector system

In our work, speaker embedding is extracted by the x-vector architecture [3], Table I shows the detail parameters. The

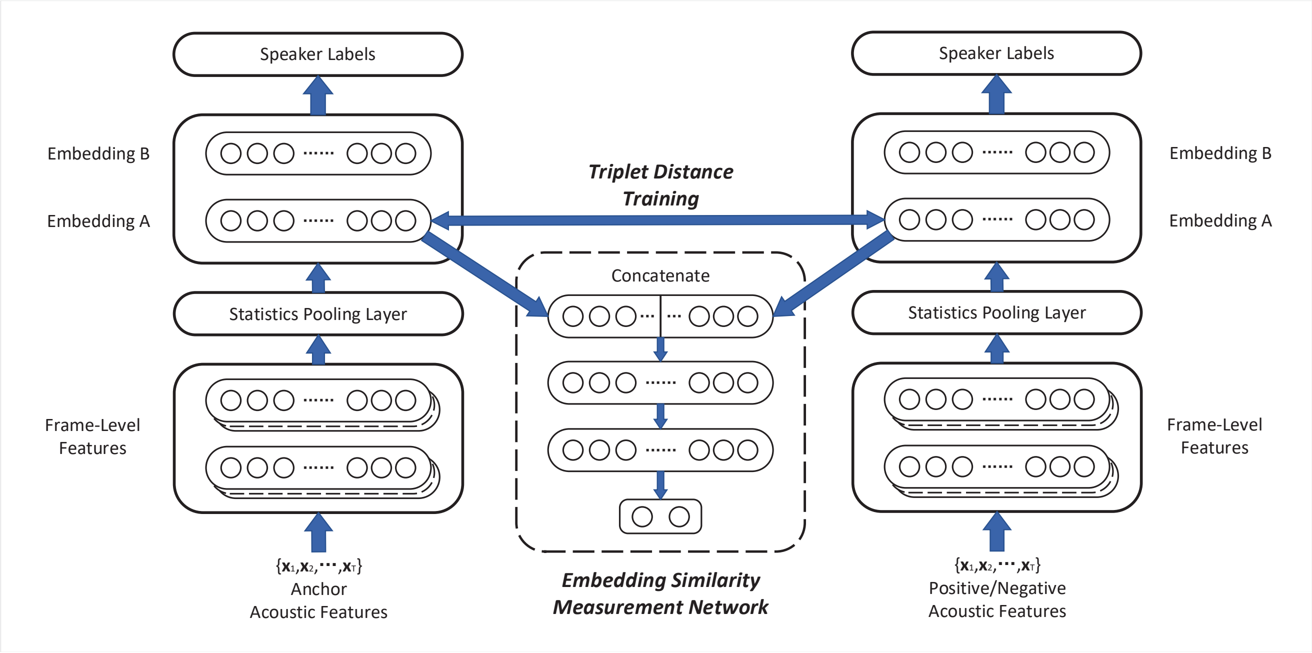

Fig. 1. Proposed triplet distance training and embedding similarity measurement network with x-vector architecture

TABLE I
BASELINE X-VECTOR SYSTEM ARCHITECTURE

Layer	Input-node	Input-dim	Output-dim
TDNN1	t-2,t-1,t,t+1,t+2	24*5	512
TDNN2	t-2,t,t+2	512*3	512
TDNN3	t-3,t,t+3	512*3	512
DNN4	t	512*3	512
DNN5	t	512	1500
stats pooling	T	1500	3000
embedding a	T	3000	512
embedding b	T	512	512
softmax	T	512	num of speakers

whole system consists of three parts and the training process is end-to-end. Firstly, 23-dimensional MFCC features are feed to the network as frame-level features. By the 5-layer time-delay neural networks (TDNNs), we can get the high-representation of frames. Then we compute the mean and standard deviation at the time dimension, and concatenate these two vectors, so-called statistics pooling layer. Fixed-dimensional utterance-level features are extracted through this operation. In the training stage, the network predicts the class of speaker by the last softmax layer with multi-class cross entropy loss. Besides, batch normalization and ReLU activation function are applied to all hidden layers.

$$\mathscr{L}_{multi-class} = -\sum_{i=1}^{M} \frac{\exp^{w_{c_i}^T x_i + b_{c_i}}}{\sum_{j}^{N} \exp^{w_j^T x_i + b_j}} \tag{1}$$

In the evaluation stage, we let the last two layers as speaker embeddings (embedding A and embedding B respectively). Techniques such as PLDA or cosine similarity techniques are then applied to the extracted embeddings for scoring the trials.

III. JOINT LEARNING OF TRIPLET DISTANCE AND SIMILARITY NETWORK

The entire architecture we proposed is as Fig 1. Since embedding A of x-vector has better performance then embedding B, we add some constraints at the layer of embedding A. Triplet distance training directly controls the distance of two embeddings and the embedding similarity measurement network uses DNN to adjust the gap between embeddings.

A. *Triplet Distance Training*

We random choice an utterance as the anchor, then choose an utterance from the same speaker as positive and an utterance from a different speaker as negative. Though cosine similarity is often used in back-end scoring, we find it can not converge if we use for restraining the triplet, thus we utilize Euclidean distance:

$$d_{Euclidean} = \sqrt{\sum_{i=1}^{N} (x_{1i} - x_{2i})^2} \tag{2}$$

$x_1$and x_2 are two vectors, the triplet loss function is as follows:

$$\mathscr{L}_{triplet} = \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + a \tag{3}$$

where (x_i^a, x_i^p, x_i^n) represents anchor, positive and negative embedding respectively, a is the empirical margin and we set the value $a = 0.8$.

B. *Embedding Similarity Measurement Network*

Euclidean distances cannot fully reflect the similarity between the two embeddings. So we try to use another neural network to measure the similarity, called embedding similarity

TABLE II
EER(%) AND DCF16 OF SRE16 EVALUATION DATASET

System Description			Pool		Tagalog		Cantonese	
ID	system	triplet	EER(%)	DCF16	EER(%)	DCF16	EER(%)	DCF16
1	i-vector [18]	×	13.6	0.711	17.6	0.842	8.3	0.549
2	x-vector	×	8.65	0.679	12.50	0.829	4.80	0.460
3	x-vector, similarity net, $\gamma = 0.3$	×	8.42	**0.633**	12.31	**0.790**	4.58	**0.424**
4	x-vector, triplet distance, $\beta = 0.1$	embedding a	**7.99**	0.686	**11.70**	0.834	**4.15**	0.425
5	Joint Training, $\beta = 0.1$, $\gamma = 0.3$	embedding a	8.04	**0.617**	12.03	**0.780**	**4.12**	**0.411**
6	Joint Training, $\beta = 0.3$, $\gamma = 0.1$	embedding a	**7.86**	0.681	**11.51**	0.834	4.17	**0.411**
7	Joint Training, $\beta = 0.3$, $\gamma = 0.1$, l2-norm	embedding a	8.78	0.687	12.62	0.833	4.88	0.474
8	Joint Training, , $\beta = 0.3$, $\gamma = 0.1$	embedding b	8.34	0.713	12.23	0.859	4.30	0.445
9	fusion of system 2 and 4	embedding a	7.39	0.644	10.87	0.807	3.82	0.397
10	fusion of system 3 and 4	embedding a	7.27	0.618	10.77	0.788	3.68	0.375

measurement network. We first obtain the embeddings though the x-vector network, because these two embeddings have the same length, then we just concatenate the two embeddings as one sequence and feed it to the new network without delimiter in between. The similarity evaluation network consists of two Bidirectional Long Short-term Memory (BLSTM) layers with 1024 nodes and two fully connected layers with 512 nodes. Each hidden layer follows batch normalization and ReLU activation function. Finally, the network gives the prediction of whether the two embeddings belong to one speaker. The loss function of this network is two-class cross-entropy loss:

$$\mathscr{L}_{similarity} = -y\log\hat{y} - (1-y)\log(1-\hat{y}) \tag{4}$$

where y is the ground truth label and $\hat{y}$ is the predicted result.

As the proposed two losses have a different optimizing aim with the original multi-class cross entropy loss, we can jointly optimize these three losses by setting different weights.

$$\mathscr{L}_{total} = \alpha\mathscr{L}_{x-vector} + \beta\mathscr{L}_{triplet} + \gamma\mathscr{L}_{similarity} \tag{5}$$

where α, β and γ are hyper-parameters and we set $\alpha = 1$, $\beta = 0.1$ and $\gamma = 0.3$ in our experiments.

IV. EXPERIMENTS AND RESULTS

A. Dataset

We use specific common corpora which are as follows for training: MIXER 6, 2010 NIST SRE and Follow-up Data, Switchboard 2 Phases 1, 2, and 3 as well as Switchboard Cellular. Data augmentation is an efficient way that makes the model more robust to the evaluation data. In our work, we add MUSAN dataset and room impulse responses (RIRs) data to the raw training data to expand the size and diversity of the training set by randomly choosing one among babble, music, noise and reverb. The RIRs and MUSAN are in 16k sampling rate and 16-bit precision. The process is applied using SRE16 recipe[1] of Kaldi speech recognition toolkit [19].

After data augmentation, the training set consists of 5145 speakers and 211062 utterances. We evaluate our system performance on NIST 2016 speaker recognition evaluations (SRE16), which consists of 802 speakers and 9294 utterances.

B. Setup

After voice activate detection (VAD), the augmentation audio is converted to 23-dimensional MFCC sequences with a frame-length of 25 ms, mean-normalized over a sliding window of 3 seconds. PyTorch toolkit is implemented for training the DNN. After getting the embedding, we use Kaldi PLDA in the evaluation stage as the backend for the experimental systems. Stochastic Gradient Descent (SGD) with weight decay=1e-8 and momentum=0.9 is used as the optimization method.

C. Results and Analysis

We report the results in terms of EER and the SRE16 official evaluation metric DCF16, which is averaged from two operation points with $P_{Target} = 0.01$ and $P_{Target} = 0.005$ respectively. The results are listed in Table II, in summary, systems with triplet distance training or embedding similarity measurement network outperform the baseline x-vector system that only trained with softmax loss. We first investigate the influence of a single improvement. The two methods show their superiorities on different evaluation indicators. More specifically, jointly training with embedding similarity measurement network (System 3) can reduce DCF obviously with a little effect of EER. At the same time, jointly training with triplet distance (System 4) remarkably decreases EER.

Then we further explore the systems of three losses. The system 5&6 evaluates the performances if we jointly train the baseline x-vector with the triplet distance and the embedding similarity measurement network. According to the results, we can observe that joint training achieves synthetically better performances compared with system 3&4. And we can adjust the EER and DCF by finetuning the hyper-parameters β and γ. Adding l2-norm is a common way in triplet loss training task, but it seems not to make sense (system 7) in our scenario. We also test the same experiment at the embedding B layer (system 8), and we can conclude that embedding A is a better choice. Finally, we test two fusion systems (System 9 and 10) equally weighted of the PLDA by using Bosaris toolkit [20], the results can also prove that the similarity measurement network can improve the performance.

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2

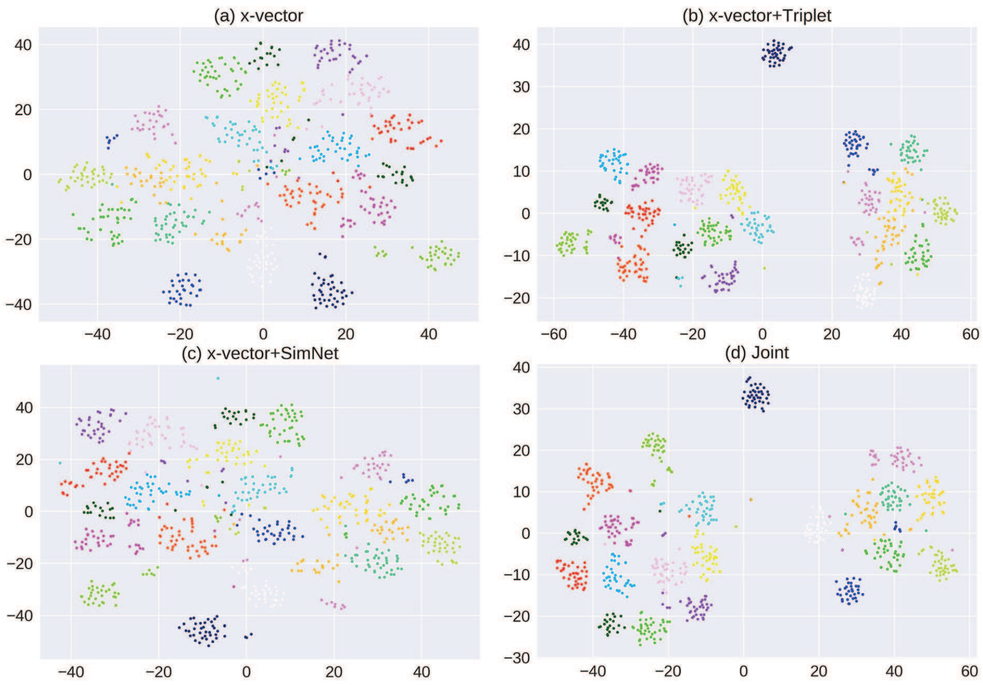

Fig. 2. visualization of different systems, plotted by the t-SNE, note the coordinate values of the longitudinal axis is different

Figure 2 shows the visualization of these systems. The observation is that the triplet distance loss shows reduced within-speaker variance, while the similarity measurement network shows its superior property on between-speaker distance. Figure 3 plots the DET curve of system 2 to system 5. The results are corresponding to the table II. The triplet distance can achieve better performance when the miss rate is very low, while the similarity measurement network performs better when the false alarm rate is low, and the performance of joint training is slightly better than the triplet distance. It is also observed that non-target scores distribution of system 3 is more negative than other systems in Figure 4, which is consistent with the results in Table II and Figure 3. Combining these factors, it is proved that the embedding similarity measurement network can truly improve the performance.

V. Conclusions

In this work, we focus on a text-independent speaker verification task and present two improvements based on triplet. We firstly implement triplet distance training and reduce EER. Getting inspiration of NLP, we design an embedding similarity measurement network for classifying whether the two embeddings belong to one speaker in the training stage, meaning we add another constraint on the embeddings. The performance shows that adding the new network can get a definite decrease of DCF while lowing the EER. Finally we get a better performance by jointly training with the three losses. The results shows benefit from our embedding similarity measurement network, therefore, the proposed methods can be applied to other end-to-end verification systems.

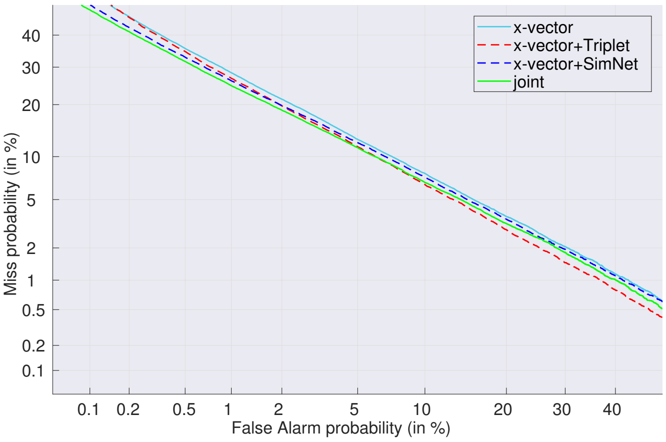

Fig. 3. DET curve for the baseline and proposed systems when the results are pooled across Cantonese and Tagalog

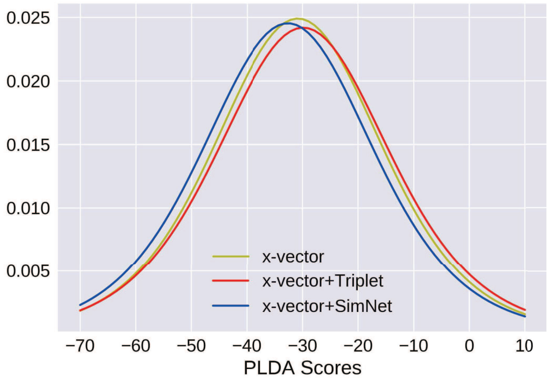

Fig. 4. PLDA scores distribution of negative trials

REFERENCES

[1] N. Dehak, P. J. 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, 2010.

[2] 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)*. IEEE, 2014, pp. 4052–4056.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[4] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Proc. Interspeech*, vol. 2018, 2018, pp. 3573–3577.

[5] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech 2018*, 2018, pp. 2252–2256. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-993

[6] L. You, B. Gu, and W. Guo, "Ustcspeech system for voices from a distance challenge 2019," *arXiv preprint arXiv:1903.12428*, 2019.

[7] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[8] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances." in *Interspeech*, 2017, pp. 1487–1491.

[9] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[10] C. Zhang, K. Koishida, and J. H. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 9, pp. 1633–1644, 2018.

[11] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular softmax loss for end-to-end speaker verification," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 190–194.

[12] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[13] S. Novoselov, V. Shchemelinin, A. Shulipa, A. Kozlov, and I. Kremnev, "Triplet loss based cosine similarity metric learning for text-independent speaker recognition," *Proc. Interspeech 2018*, pp. 2242–2246, 2018.

[14] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[15] S. Yadav and A. Rai, "Learning discriminative features for speaker identification and verification," in *Proc. Interspeech 2018*, 2018, pp. 2237–2241. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1015

[16] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training."

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[18] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification." in *Interspeech*, 2017, pp. 999–1003.

[19] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[20] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.