ROBUST SPEAKER VERIFICATION WITH JOINT SELF-SUPERVISED AND SUPERVISED LEARNING

Kai Wang[1], *Xiaolei Zhang*[1], *Miao Zhang*[1], *Yuguang Li*[1], *Jaeyun Lee*[2], *Kiho Cho*[2], *Sung-UN Park*[2]

[1]Samsung R&D Institute China Xian, [2]Samsung Advanced Institute of Technology

ABSTRACT

Supervised learning and self-supervised learning address different facets. Supervised learning achieves high accuracy, but it requires numerous expensive labeled data indeed. Correspondingly, self-supervised learning, makes use of abundant unlabeled data to learn, but the performance lags behind that of the supervised counterpart. To overcome the difficulty of acquiring annotated data and contain the high performance in the context of speaker verification, we propose in this work a self-supervised joint learning (SS-JL) framework which complements the supervised main task with self-supervised auxiliary tasks in joint training. These auxiliary tasks help the speaker verification pipeline to generate robust speaker representation that is closely relevant to voiceprints. Our model is trained on English dataset and tested on multilingual datasets, including English, Chinese and Korean datasets, and 13.6%, 12.7% and 13.5% improvement is achieved respectively in terms of equal error rate (EER) compared with the baselines.

Index Terms— speaker verification, supervised learning, self-supervised learning, joint learning, multilingual datasets

1. INTRODUCTION

Speaker verification refers to the process of identifying whether the input audio comes from the registered user. Recent state-of-the-art studies in this field leverage deep neural networks (DNN) to extract robust speaker representations [1–4]. These approaches focus mainly on learning from large corpora in a supervised manner, which comes with two major disadvantages. Firstly, collecting large amounts of annotated data is expensive and it raises concern on privacy. Secondly, the tag information tends to be overly exploited by the model. Since there is rich speaker information hidden in the untagged space, over-reliance on speaker-label will discourage the model from exploring untagged dimensions and thus limit the scope of learning and hinder its generalization.

To address these two problems, self-supervised learning is proposed to use unlabeled data and it has gained increasing attention from the academics. In [5, 6], reconstructed speech segments are employed to learn speaker representations and this approach benefits from largely available unlabeled data. Due to insufficient self-exploration of the data, current self-supervised models still cannot compete with supervised models in performance. In [7], Zhang et al. use the contrastive self-supervised learning approach to pre-train the model and then finetune it with a small amount of labeled data to achieve comparable performance. However, the training is complicated and the performance is still lower than that of supervised learning.

Therefore, we combine supervised and self-supervised learning to boost the accuracy and generalization ability of the model. More specifically, we propose the self-supervised joint learning (SS-JL) framework which combines supervised learning and self-supervised learning with multi-task learning (MTL). In this work, we also present two other modules in addition to speaker verification, namely speaker diarization and speaker invariant. With such sufficient mining of the data, we expect the model to learn richer speaker features. The experimental results show that both the performance and generalization ability of these supervised tasks are improved due to self-supervised tasks. Overall, the main contributions of this paper are as follows:

(1) A self-supervised joint learning framework is proposed to weave self-supervised learning into the training objective of supervised learning on speaker verification task. The purposed SS-JL framework aims to boost the robustness of speaker representation.

(2) A novel speaker invariant module is proposed to make full use of data and improve the representation ability of speaker representation without label.

(3) An empirical experiment is conducted by training our model on English dataset and testing it on datasets of three different languages. It can be concluded that self-supervised learning outperforms supervised learning and brings state-of-the-art results as well.

2. METHODS

The proposed SS-JL framework aims to significantly improve the performance and generalization ability of speaker verification by designing self-supervised sub-tasks in multi-task training. The speaker diarization module enhances speaker embedding by extracting the target speaker from a mixture of speech, aiming at expanding the distance between inter speakers. The speaker invariant module performs the voice

conversion task, which reduces the distance between speakers by maintaining the consistency of the same speaker's voiceprint in self-supervised learning. The SS-JL framework leads to a robust speaker embedding. The speaker representation extracts mean and standard deviation from speaker embedding. When a robust speaker embedding is obtained, the performance and generalization of speaker verification are improved. In this section, we describe how the SS-JL framework is combined with supervised learning and self-supervised learning. Moreover, we also show the detailed structure of the three modules.

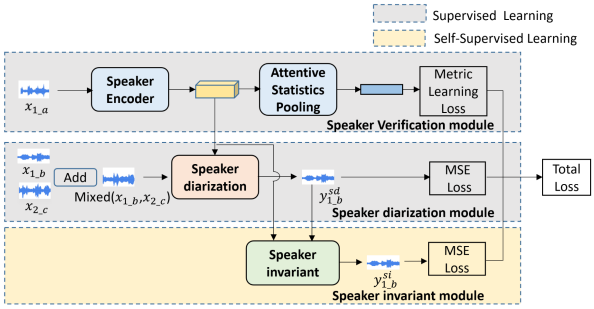

Fig. 1. Structure of the SS-JL framework. Speaker diarization and speaker invariant use the same ground truth x_{1_b}.

2.1. self-supervised joint learning framework

The SS-JL framework consists of a speaker verification module, a speaker diarization module and a speaker invariant module as shown in Fig.1. The speaker verification module is used to extract the speaker representation of the input speech. This module encodes an audio input sequence X_{1_a} into a speaker embedding E_{1_a} by speaker encoder $Enc^{sv}(\cdot)$ as shown below. X_{1_a} represents the utterance A of speaker 1.

$$E_{1\_a} = Enc^{sv}(X_{1\_a}) \tag{1}$$

The attentive statistics pooling [8] Asp^{sv} predicts a 256-length sequence of speaker representation R_{1_a} which is formulated by:

$$R_{1\_a} = Asp^{sv}(E_{1\_a}) \tag{2}$$

The speaker diarization module partitions an input mixed audio X_{mixed} into homogeneous segments according to the speaker embedding identity E_{1_a}. The mixed speech combines utterance B X_{1_b} of speaker 1 and utterance C X_{2_c} of speaker 2. Given the feature extractor $F^{sd}(\cdot)$, the output $Y^{sd}_{1\_b}$ can be written as:

$$Y^{sd}_{1\_b} = F^{sd}(X_{mixed}, E_{1\_a}) \tag{3}$$

The speaker invariant module under investigation involves voice conversion. This module uses self-supervised learning. And there are two inputs, one is the output speaker embedding E_{1_a} of speaker verification module, and the other is the output $Y^{sd}_{1\_b}$ of speaker diarization module. The speaker invariant module could be described as:

$$Y^{si}_{1\_b} = F^{si}(Y^{sd}_{1\_b}, E_{1\_a}) \tag{4}$$

A training framework for joint supervised learning and self-supervised learning is described. For each mini-batch, we randomly select N speakers. For each speaker, we draw M groups, and in each group we randomly draw one sentence of this speaker. We split the sentence into two completely non-overlapping speech segments (utterance A & utterance B) of 2seconds in length. Then, we need to extract a sentence from other speakers and cut 2seconds of speech segment as utterance C, with utterance A and utterance B as a group. Thus we get $3 \times M \times N$ utterance segments.

We adopt angular prototypical (AP) loss [9] for optimizing speaker verification module. We use the cosine similarity score as the distance between two speaker representations. Angular prototypical loss is defined as:

$$L_{ap} = -\frac{1}{N}\sum_{j=1}^{N} log \frac{e^{w*cos(x_{j,M},c_j)+b}}{\sum_{k=1}^{N} e^{w*cos(x_{j,M},c_k)+b}} \tag{5}$$

c_k is the centroid for speaker.

Mean square error (MSE) is used as the target speaker feature objective function of speaker diarization and speaker invariant modules, which aim to improve the expressive power of speaker embedding. Both modules have the same ground truth $y^{gt}_{1\_b}$. It can be defined as:

$$L^{sd}_{mse} = \frac{1}{N}\sum_{i=0}^{N} \parallel y^{gt}_{i\_b} - y^{sd}_{i\_b} \parallel^2 \tag{6}$$

$$L^{si}_{mse} = \frac{1}{N}\sum_{i=0}^{N} \parallel y^{gt}_{i\_b} - y^{si}_{i\_b} \parallel^2 \tag{7}$$

where $y^{sd}_{i\_b}$ and $y^{si}_{i\_b}$ are the predicted value, and N is the number of samples.

Total Loss. The speaker verification loss L_{ap} works together with the speaker diarization loss L^{sd}_{mse} and speaker invariant loss L^{si}_{mse}. The final loss is jointed by the coefficient λ:

$$L_{total} = \lambda_{ap}L_{ap} + \lambda^{sd}_{mse}L^{sd}_{mse} + \lambda^{si}_{mse}L^{si}_{mse} \tag{8}$$

2.2. Network Structure

Fig.2 shows the detailed structure of SS-JL framework. The speaker verification module uses Squeeze-and-Excitation (SE) Block [10] as the basic block and references the structure of ResSENet. The detailed implementation of the speaker diarization module is shown in Fig.2(b). Typically, speaker embedding aims to improve the performance of speaker diarization module [11–13]. In our framework, we combine speaker diarization with speaker verification to enrich the expression

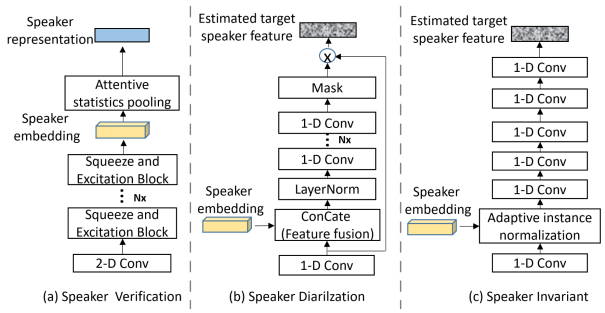

Fig. 2. Details of the network structures.

of speaker embedding by the advantage of MTL. The speaker diarization module implements a modified frequency-domain structure by referring to the Conv-TasNet [14] structure.

We utilize the speaker invariant module to replace the voiceprint in speech. In principle, two segments taken from a single utterance should yield identical speaker embedding since they come from the same speaker. If one segment is reconstructed from the speaker embedding extracted from the other segment, the reconstructed audio should be close to the original audio. By enforcing this reconstruction consistency, the speaker verification module is encouraged to generate more robust speaker embedding better capturing the voiceprint. The detailed implementation of the speaker invariant module is shown in Fig.2(c). This module uses multiple 1D-Conv and adaptive instance normalization (AdaIN) [15] for voice conversion. Chou et al. [16] map different speakers to different values of mean and standard deviation. A robust mean and standard deviation means a robust speaker representation. Hence, we utilize the AdaIN layer for voice conversion to optimize the mean and standard deviation of speaker embedding by self-supervised learning. AdaIN is defined as:$AdaIN(x, y) = \sigma(y)\frac{x-\mu(x)}{\sigma(x)} + \mu(y)$, where $\mu(\cdot)$ and $\sigma(\cdot)$ represent the mean value and standard deviation value of the embedding respectively.

3. CORPORA AND EXPERIMENTAL SETUP

3.1. Speech corpora

VoxCeleb2 [17] development set is used for training. The performance of the proposed network is assessed by VoxCeleb1 [18], AISHELL-1 [19], Zeroth Korean [20] and our private datasets. VoxCeleb2 dataset consists of 1,092,009 utterances spoken by 5,994 speakers. VoxCeleb1 dataset contains 147,935 utterances among 1,211 speakers. AISHELL-1 dataset is 178 hours long. Speakers from different areas of China were invited to participate in recording. Zeroth Korean dataset contains transcribed audio data in Korean. The training data has 22,263 utterances and the test data has 457 utterances. Our private dataset includes clean speech and noise background. 10 keywords and 130 utterances are spoken by 30 speakers. Near-field and far-field audios were recorded by 4 channel microphones at different distances ranging from 0.5m to 3m.

3.2. System setups

Model details. All networks are implemented with PyTorch. The models are trained on 2 NVIDIA Titan GPUs. The input features are 40-dimensional log mel-filterbank feature vectors generated by librosa library [21]. Pre-emphasis is applied to the audio with the coefficient 0.97. We use a hamming window with a width of 25 ms, a step of 10 ms and 512-point FFT. We randomly extract the segment of 2 seconds from each utterance for training. Mean and Variance Normalization (MVN) is applied to the input utterance. There is no voice activity detection (VAD) in training and testing.

The models are trained with Adam optimizer [22]. For the ResSENet model, 200 speakers are randomly selected in the training set while two utterances are provided by each speaker. Therefore, there are 400 utterances in a batch. We train the speaker verification baseline model with AP loss. For the ResSENet-SD and ResSENet-SD-SI models, we load the pre-trained ResSENet model, and then train the model with total loss. For the weights of the loss function, we adopt $\lambda_{ap} = \lambda_{mse}^{sd} = \lambda_{mse}^{si} = 1$. The initial learning rate is 0.001. The learning rate is reduced by 5% every 10 epochs.

Data augmentation. No data augmentation used in training phase for fair comparison, apart from the random sampling.

3.3. Evaluation metrics

The system will be measured by the following two performance metrics: (a) Equal Error Rate (EER) is a rate which the false acceptance rate (FAR) is equal to false rejection rate (FRR), (b) Minimum detection cost function (minDCF) with $P_{target} = 0.01$, $C_{FA} = 1.0$ and $C_{Miss} = 1.0$. The speaker representations are computed from enrollment recordings one by one. The cosine distance is used to calculate similarity.

4. RESULTS

4.1. Progressive system improvements on public corpora

Table 1 shows the improvement of different parts on the performance of speaker verification. ResSENet-SD-SI improves nearly 49.7% over the x-vector. It requires similar computational consumption for inference compared with the ResSENet model, while the performance gain is significant. We proceed with an ablation study of individual components introduced in Section 2. ResSENet-SD obtains the best minDCF, which means that the speaker diarization module makes FAR lower in the VoxCeleb1 validation set. Meanwhile, our method is easy to apply because it can be added to existing speaker verification backbones.

Table 1. Training only on VoxCeleb2 (English) and evaluation on VoxCeleb1 (English). (*: our re-implementation; Vox1: VoxCeleb1; ENG: English)

Method	Corpus	Lang	EER (%)	minDCF
x-vector* [1]	Vox1	ENG	3.29	0.35
TDNN [3]	Vox1	ENG	2.00	0.25
ResNet [9]	Vox1	ENG	2.21	-
ResSENet* [9]	Vox1	ENG	1.91	0.25
ResSENet-SD(ours)	Vox1	ENG	1.74	**0.21**
ResSENet-SD-SI(ours)	Vox1	ENG	**1.65**	0.24

To evaluate the discriminability of the speaker embedding obtained through ResSENet and ResSENet-SD-SI models, we use t-Distributed Stochastic Neighbor Embedding (t-SNE) [23] to visualize high dimensional representations as shown in Fig.3. The embeddings come from five speakers, each with over 60 utterances. The ratio of intra speaker cosine similarity over inter speaker cosine similarity for ResSENet and ResSENet-SD-SI is 0.412 and 0.361 respectively. The SS-JL framework is able to reduce the intra-class spacing and expand the inter-class spacing, possessing better speaker separation capabilities.

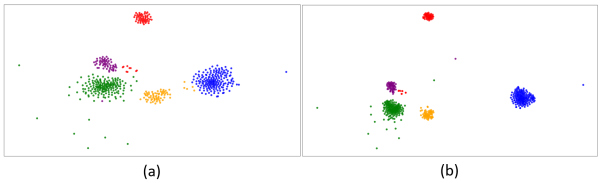

Fig. 3. The t-SNE speaker embedding visualization of five speakers on VoxCeleb1 dataset: (a) ResSENet embeddings, (b) ResSENet-SD-SI embeddings. A color correspond to a speaker while a dot correspond to an utterance.

4.2. Progressive system improvements on multilingual corpora

We have presented the techniques that help to improve the performance of speaker verification in multilingual conditions. As shown in Table 2, compared with the x-vector, our method brings 22.3% and 14.6% improvement to EER. Compared with ResSENet, ResSENet-SD-SI brings 12.7% and 13.5% improvement to EER. Testing on multilingual datasets remove the influence of language and semantics. This demonstrates that the SS-JL framework has greatly improved the generalization of model.

4.3. Progressive system improvements on private corpora

Our framework brings improvement in the two test scenarios of "same content" and "different content". The "same content" and "different content" are similar to the text-dependent and text-independent speaker verification respectively. As illustrated in Table 3, the best network improves EER by 10.8% and 1.6% over a strong ResSENet network. The results demonstrate that the SS-JL framework can improve performance continuously under different environmental conditions.

Table 2. Training only on VoxCeleb2 (English) dataset and evaluation on AISHELL-1 (Chinese) and Zeroth Korean (Korean) datasets. (*: our re-implementation; AISH: AISHELL-1; Zero_K: Zeroth Korean; CHN: Chinese; KOR: Korean)

Method	Corpus	Lang	EER (%)	minDCF
x-vector* [1]	AISH	CHN	5.23	0.67
ResSENet* [9]	AISH	CHN	4.65	0.57
ResSENet-SD(ours)	AISH	CHN	4.81	0.57
ResSENet-SD-SI(ours)	AISH	CHN	**4.06**	**0.51**
x-vector* [1]	Zero_K	KOR	2.48	0.38
ResSENet* [9]	Zero_K	KOR	2.45	0.32
ResSENet-SD(ours)	Zero_K	KOR	2.23	0.30
ResSENet-SD-SI(ours)	Zero_K	KOR	**2.12**	**0.28**

Table 3. Training only on VoxCeleb2 (English) dataset and evaluation on private (English) dataset. In the "Same Content" scenario, the content of enrolled utterance and test utterance is the same, while in the "Different Content" scenario, the content of enrolled utterance and test utterance is different. (*: our re-implementation)

Method	Condition	EER (%)	minDCF
x-vector* [1]	Same Content	10.24	0.77
ResSENet* [9]	Same Content	4.90	0.46
ResSENet-SD(ours)	Same Content	4.69	0.47
ResSENet-SD-SI(ours)	Same Content	**4.37**	**0.40**
x-vector* [1]	Different Content	15.09	0.97
ResSENet* [9]	Different Content	11.23	0.93
ResSENet-SD(ours)	Different Content	11.38	0.92
ResSENet-SD-SI(ours)	Different Content	**11.05**	**0.91**

5. CONCLUSION

In this paper, we develop a self-supervised joint learning framework based on speaker verification, speaker diarization and speaker invariant modules. We also propose a novel speaker invariant module based on self-supervised learning. Furthermore, we apply an end-to-end training method for the SS-JL framework. In experimental evaluations, the improvement of more than 12% on multilingual public and private corpora demonstrates the effectiveness of our proposed framework. In the future, we intend to focus more on fully self-supervised multi-task learning.

6. REFERENCES

[1] C.D. Jones, A.B. Smith, and E.F. Roberts, "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.

[2] Chunlei Zhang, Kazuhito Koishida, and John H. L. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1633–1644, 2018.

[3] Yi Liu, Liang He, and Jia Liu, "Large margin softmax loss for speaker verification," in *Proc. Interspeech 2019*, 2019, pp. 2873–2877.

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

[5] Arindam Jati and Panayiotis Georgiou, "Neural predictive coding using convolutional neural networks toward unsupervised learning of speaker characteristics," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1577–1589, 2019.

[6] T. Stafylakis, J. Rohdin, O. Plchot, P. Mizera, and L. Burget, "Self-supervised speaker embeddings," in *Proc. Interspeech 2019*, 2019, pp. 2863–2867.

[7] Haoran Zhang, Yuexian Zou, and Helin Wang, "Contrastive self-supervised learning for text-independent speaker verification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6713–6717.

[8] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Interspeech 2018*, 2018.

[9] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, "In defence of metric learning for speaker recognition," in *Proc. Interspeech 2020*, 2020, pp. 2977–2981.

[10] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu, "Squeeze-and-excitation networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[11] Mesut Toruk, Gokhan Bilgin, and Ahmet Serbes, "Speaker diarization using embedding vectors," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 2020.

[12] Y. Fathullah, C. Zhang, and P. C. Woodland, "Improved large-margin softmax loss for speaker diarisation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7104–7108.

[13] Yanpei Shi and Thomas Hain, "Supervised speaker embedding de-mixing in two-speaker environment," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 758–765.

[14] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[15] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1510–1519.

[16] Ju chieh Chou and Hung-Yi Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *Proc. Interspeech 2019*, 2019, pp. 664–668.

[17] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[18] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[19] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*.

[20] "Zeroth korean," http://www.openslr.org/40/.

[21] Brian McFee, Colin Raffel, Dawen Liang, D. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," 2015.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[23] Hinton G E. Der Maaten L V, "Visualizing data using t-sne," in *Journal of Machine Learning Research*, 2008, pp. 2579–2605.