VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

SV - VLSP2021: The Smartcall - ITS's Systems

Dinh Van Hung[1], Mai Van Tuan[1,*], Dam Ba Quyen[1], Nguyen Quoc Bao[1,2]

[1]Smartcall Joint Stock Company, Hai Dang, Nam Tu Liem, Hanoi, Vietnam
[2]Thai Nguyen University of Information and Communication Technology,
Z115, Quyet Thang, Thai Nguyen, Vietnam

Received 27 December 2021
Revised 5 April 2022; Accepted 5 May 2022

Abstract: This paper presents the Smartcall - ITS's systems submitted to the Vietnamese Language and Speech Processing, Speaker Verification (SV) task. The challenge consists of two tasks focusing on the development of SV models with limited data and testing the robustness of SV systems. In both tasks, we used various pre-trained speaker embedding models with different architectures: TDNN, Resnet34. After a specific fine-tuning strategy with data from the organiser, our system achieved the first rank for both two tasks with the Equal Error Rate respectively are 1.755%, 1.95%. In this paper, we describe our system developed for the booth two tasks in the VLSP2021 Speaker Verification shared-task.

Keywords: Speaker verification, X-vector, TDNN, Resnet34, Transfer learning.

1. Introduction

Recently, the field of speaker verification has advanced rapidly, due to the development of neural network based speaker embeddings called x-vectors [1] and their various improvements. The neural-network based models require large-scale speaker recognition training datasets that have been recently released [2]. In this year, the VLSP community has organized an evaluation campaign for the Vietnamese speaker verification task. A corpus contains speech from 1305 speakers was provided as training data. In task 1, participants can only use this dataset for model development. Any use of additional data for model training is prohibited, but the use of public pre-trained model is allowed. In task 2, there is no limitation to use other data resources.

During the challenge, we explored several speaker embedding model architectures, loss functions and data augmentation methods, transfer leaning techniques. This paper gives a detailed description of the models that we used in the final submission. The rest of the paper is organized as follows. Section 2 describes the speech corpus. Section 3 presents our proposed systems. Section 4 shows the results and Section 5 concludes.

[*] Corresponding author.
E-mail address: maituanbk2012@gmail.com

https://doi.org/10.25073/2588-1086/vnucsce.339

2. Datasets

2.1. Official Training Datasets

The provided training set consists of 34781 utterances which belong to 1305 different speakers. The transcript of each utterance is not provided. Number of utterance per speaker range from 2 to 500. The public test contains 2941 utterances with a trials of 20000 audio pairs. Meanwhile, the private test for both task1 and task2 share the same audio set which consists of 3983 audio files, with different trials of 40000 pairs for each task. Figure 2 shows the distribution in utterance’s length for the training set. The range of duration is from 2 to 14 seconds with the average duration of each utterance is 4.7 seconds. Since there are no official development datasets, we randomly split the official training datasets into training subsets and and development subsets. The development subsets contain 100 speakers and the training subsets consist of 1205 remaining speakers.

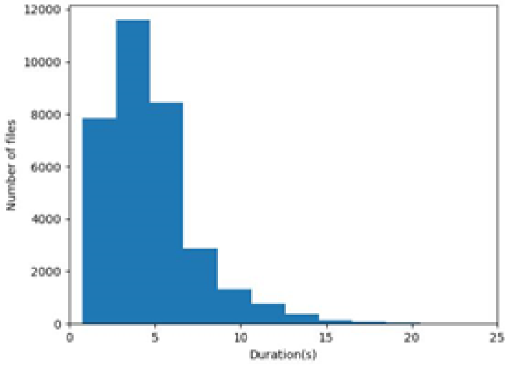

Figure 1: Duration distribution of training data.

2.2. Non-speech Datasets

The challenge allows using non-speech datasets for data augmentation purpose. Our data augmentation strategy uses additive noises and reverberation. For additive noise, we use the additive noise and noise subsets of the MUSAN corpus [3]. For reverbration, we use simulated small and medium room from Room impulse response and noise database [4]. Both MUSAN and the RIR datasets are freely available from http://www.openslr. org

2.3. Data Augmentation

Augmentation increases the amount and diversity of the existing training data. Our strategy employs additive noises and reverberation. We use a 4-fold augmentation that combines the original “clean” training list with three augmented copies. To augment a recording, we choose between one of the following randomly:

- babble: Three to seven speakers are randomly picked from MUSAN speech, summed together, then added to the original signal (13-20dB SNR).
- music: A single music file is randomly selected from MUSAN, trimmed or repeated as necessary to match duration,and added to the original signal (5-15dB SNR).
- noise: MUSAN noises are added at one second intervals throughout the recording (0-15dB SNR).
- reverb: The training recording is artificially reverberated via convolution with simulated RIRs.

3. Proposed Methods

Figure 2 describes the overview of our solution to the challenge. Several pre-trained embeddings were used to fine-tune along with the augmented datasets. We use different backends as score metrics to evaluate the correlation between two embedding vectors. And to improve the performance, we used score normalization techniques before the stage of combination results from different systems.

3.1. Utterance Embedding Extractors

Our goal is to design robust systems for the speaker verification. We will therefore present several systems including TDNN x-vector systems, Resnet 34 x-vector systems and Transfered x-vector systems.

Our first pre-trained extractor was a TDNN model in Kaldi [5] format. This extractor is trained on Voxceleb [6] 16Khz audio data. The input are 30-dimensional MFCC featues extracted using 25ms windows and 15ms overlap and further nomalized using short-term mean nomalization with a sliding window of 3s. The networks stacked 5 TDNN layers before the the pooling layer and the 512 dimensional x-vectors are extracted from the layer right after the pooling layer. The model is freely available at https://kaldi-asr.org/models/m7.

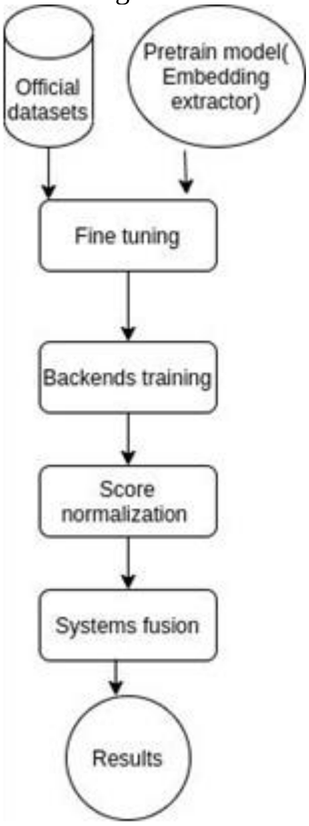

Figure 2: Pipeline of developments SV systems.

With the TDNN x-vector model, we replace the last softmax layer by a new softmax layer which have the number of unit same as the number of speaker in training set. Then the whole network is re-trained with the whole data by a smaller learning rate (0.0001). This work was done by using Kaldi toolkit. It took about 2 hours of training with a single GPU GTX 2080 Ti.

The second pre-trained embedding was a Resnet34 model. A fixed length 2-seconds temporal segment randomly extracted from each utterance was chosen as input audio. After that, the input for Resnet34 architecture was a fixed length 2-second temporal segment extracted randomly from each utterance. The input feature is 64 - dimensional log Mel filterbanks extracted by using a hamming window of width 25ms ans steps 10ms along with FFT size of 512. Mean and variance normalization (MVN) are performed by applying instance normalization to the network input. The pre-trained model was trained on Voxceleb dataset and freely available at https://www.robots.ox.ac. uk/~joon/data/bas -eline_v2_ap.model.

It's architecture is similar to TDNN - x vector but stacked TDNN layers are replaced by Resnet34 networks, the details are described in [7]. The network is then tuned using various types of loss functions:

- Additive angular margin softmax (AAM-softmax) which have been proposed in face recognition and successfully applied to speaker recognition.
- Angular Prototypical (AP Loss) loss which is a variant of the prototypical networks with an angular objective, has been used in [8]
- Softmax loss which consists of a softmax function followed by a multi-class cross-entropy loss.

Thanks to Pytorch toolkit1, it took abour 3 hours with a single GPU GTX 2080 TI to finish 50 epochs. The initial learning rate was set to 0.0001, and the Adam optimizer was used to train the network.

3.2. Backends Scoring

We used two different methods as backend for speaker verification task. The first one is cosine similarity2 which was used as a metric to compute the distance between the two embedding vectors. Beside the Cosine distance, we also use the same type of PLDA [9] classifier for the x-vector systems. The representations are centered, and projected using LDA. The LDA dimension was set to 200 for x-vectors. After dimensionality reduction, the representations are length-normalized and modeled by PLDA. This work was done by using Kaldi toolkit.

3.3. Score Normalization

To normalize the scores, we used adaptive symmetric score normalization (as-norm) [10]. We randomly selected one utterance per speaker to synthesis a cohort set of 1205 utterances.

3.4. Score Fusion

The linear logistic regression with the Bosaris toolkit [11] was used to fuse all the sub systems. This toolkit was totally written in Matlab. The fusion weights were tuned for minimizing EER in development set and then are applied for the two test sets.

4. Results

The detail results in term of EER for different testsets are shown in Table 1. Two sub-systems was used for submition: TDNN X-vector combined with PLDA backend and Resnet34 X-vector combined with cosine backend. As in the table, PLDA was not good with Resnet34, but brought significant improvement when combining with TDNN embedding. Because the learned embeddings from the softmax loss are optimized for inter-class separation alone without taking into account intra-class compactness, it need to combine with a trained PLDA to adapt the in-domain data. The is no clearly difference in term of EER between two private test sets (about 10% relative change), because the two sets share the same audio set. But we can see a big increasing in term of EER from public test to private test (about 50% relative change), maybe the private test contain more audios that are considered as out of domain data. The score normalization was also works with about 5% relative improvement on both private test sets. The fusion system has a big improvement compared to the two sub systems. This can be explained that it was combined from two different metrics (PLDA and cosine). By investigating all of the false reject cases, we found that in these cases the duration of audio is too small, there are only one or two words spoken in the utterance. So, the system is not good in short duration scenarios. With these results, our proposal ranks first in the 2021 VLSP competition on Vietnamese SV [12].

Table 1: EER for different experiments(%):

	System		Public test		Private T1		Private T2	
	Embedding	Backend	No-norm	AS-norm	No-norm	AS-norm	No-norm	AS-Norm
1	TDNN X-vector	PLDA	2.0	1.92	3.71	3.5	3.82	3.12
2	TDNN X-vector	Cosine	5.3		6.34		6.52	
3	Resnet34 X-vector	PLDA	1.52		2.49		2.55	
4	Resnet34 X-vector	Cosine	1.50	1.61	2.43	2.34	2.53	2.13
Fusion(1+4)			1.35	1.22	1.79	1.75	1.98	1.95

5. Conclusion

In this paper, we described the Smartcall - ITS winning system submitted for the Speaker verification task of VLSP 2021. Several strong embedding extractors, score metrics as well as transfer learning strategy are explored in our experiments. The linear fusion from various systems is also investigated and brought significant improvement. The final submission yielded EER of 1.75% on T1 task and EER of 1.95% on T2 task, which archived the best place in both tasks.

References

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, S. Khudanpur, X-Vectors: Robust DNN Embeddings for Speaker Recognition, doi = 10.1109/ICASSP.2018.8461375, 2018, pp. 5329–5333.

[2] J. S. Chung, A. Nagrani, A. Zisserman, VoxCeleb2: Deep Speaker Recognition, 2018, pp. 1086–1090. doi:10.21437/Interspeech. 2018-1929.

[3] D. Snyder, G. Chen, D. Povey, MUSAN: A Music, Speech, and Noise Corpus.

[4] T. Ko, V. Peddinti, D. Povey, M. Seltzer, S. Khudanpur, A study on data augmentation of

reverberant speech for robust speech recognition, 2017, pp. 5220–5224, doi:10.1109/ICASSP.2017. 7953152.

[5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlícek,ˇ Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, K. Vesel, The Kaldi speech recognition toolkit, IEEE 2011 Workshop on Automatic Speech Recognition and Understanding.

[6] A. Nagrani, J. S. Chung, A. Zisserman, VoxCeleb: A Large-Scale Speaker Identification Dataset,
doi: 10.21437/Interspeech.2017-950.

[7] H. Heo, B.-J. Lee, J. Huh, J. S. Chung, Clova Baseline System for the VoxCeleb Speaker Recognition Challenge, 2020.

[8] J. S. Chung, J. Huh, S. Mun, M. Lee, H. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, I. Han, In defence of metric learning for speaker recognition.

[9] S. Ioffe, Probabilistic Linear Discriminant Analysis, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 531–542.

[10] S. Cumani, P. Batzu, D. Colibro, C. Vair, P. Laface, V. Vasilakakis, Comparison of Speaker Recognition Approaches for Real Applications., Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH 2011, pp. 2365–2368

[11] N. Brummer, E. Villiers, The BOSARIS Toolkit: Theory, Algorithms and Code for Surviving the New DCF, arXiv 1304.

[12] V. T. Dat, P. V. Thanh, N. T. T. Trang, SV Challenge: Vietnamese Speaker Verification in noisy environments, VLSP, 2021.