The Krisp Diarization system for the VoxCeleb Speaker Recognition Challenge 2023

Davit Karamyan[1,2], *Grigor Kirakosyan*[2,3]

[1]Russian-Armenian University, Yerevan
[2]Krisp.ai, Yerevan
[3]Institute of Mathematics of NAS RA, Yerevan
{dkaramyan, gkirakosyan}@krisp.ai

Abstract

This report describes the Krisp team submission system at the 2023 VoxCeleb Speaker Recognition Challenge (VoxSRC) Track 4. Our system consists of fused voice activity detection, multi-scaled speaker embedding, clustering-based diarization, and overlapped speech detection models. The DOVER-Lap technique was employed for system fusion, with the subsequent step involving the detection and handling of overlapping speech. Our final submission involves an ensemble of seven systems, resulting in a diarization error rate (DER) of 4.71% on the challenge evaluation set, securing the 2nd place in the diarization track of the competition.

Index Terms: VoxSRC23, Speaker Diarization

1. Introduction

Speaker diarization (SD) is the process of dividing an input audio stream into homogeneous segments according to the speaker's identity. A typical speaker diarization system usually consists of several steps: (1) Speech segmentation, where the input audio is segmented into short sections that are assumed to have a single speaker and the non-speech sections are filtered out by Voice Activity Detection (VAD), (2) Speaker embedding extractor, where speaker embeddings are extracted from segmented sections, (3) Clustering, where the extracted audio embeddings are grouped into clusters based on the number of speakers present in the audio recording, and optionally, (4) Resegmentation step is performed to further refine clustering results. To further improve the performance, many research works have focused on overlapped speech detection (OSD) to reduce the missed speaker error.

In this report, we propose a clustering-based SD system for the Diarization Task of the 2023 VoxCeleb Speaker Recognition Challenge (VoxSRC23). The proposed system consists of several submodules such as voice activity detection, speaker embedding extraction, clustering, and OSD. In summary, the main ideas employed throughout the paper were as follows:

- *Different VAD modalities*: We tackle the VAD part using several approaches, including noise cancellation (NC) and automatic speech recognition (ASR). The outputs of these methods are then fused with the conventional supervised VAD models.
- *Multiscale segmentation*: To minimize speaker confusion errors that arise due to uniform segmentation, we create multiple affinity matrices for different window and shift sizes. After constructing the matrices, we proceed to compute their weighted average.
- *Noise Robustness*: We apply the Teacher-Student approach to enhance the resilience of the speaker embedding extractor against noise and reverberation. Additionally, we utilize a series of refinement steps to eliminate noise from the affinity matrix.
- *Different clustering algorithms*: We employ both spectral and agglomerative hierarchical clustering algorithms since their combination leads to more precise results.

2. System Configuration

2.1. Voice Activity Detection

We employ four different VAD models with different modalities.

2.1.1. GRU-based VAD (model #1)

We use a stack of 4 GRU layers [1], incorporating layer normalization [2] between each layer. The final dense layer with sigmoid activation is responsible for calculating the likelihood of speech occurrence. With this setup, we generate a probability score for every 30ms of speech. Higher values, nearing 1, signify the presence of speech, whereas values closer to 0 suggest its absence. We use Voxconverse [3] dev set for training and Voxconverse test set for validation.

2.1.2. NC-based VAD (model #2)

We adopt the Noise Cancellation [4] model to perform voice activity detection. First, we apply the NC model to remove any noise and non-speech signals from the original audio. Subsequently, for each 50ms interval, we calculate the energy of that interval and establish a threshold. If the energy level exceeds the threshold, we label the segment as speech; otherwise, it's categorized as non-speech. Additionally, we apply simple post-processing steps, to obtain homogeneous speech activity segments. The architecture of the NC model is the same as GRU-VAD architecture, with the distinction being that it generates a mask. This mask is subsequently applied to the input spectrogram and transformed into a waveform using the Inverse Fourier Transform.

2.1.3. ASR-based VAD (model #3)

Another approach to detect voice activity segments involves making use of an ASR model to generate timestamps at the level of individual words. We derive word-level timestamps by employing the Conformer-Medium checkpoint available in *NeMo*[1] package. Similar to NC-based VAD, here we also apply post-processing steps to obtain homogeneous speech segments.

[1]https://github.com/NVIDIA/NeMo

2.1.4. Pyannote VAD (model #4)

We use *pyannote.audio 2.1*[2] segmentation pipeline for computing the voice activity regions. We conduct a hyperparameter search for optimal values of the *onset*, *offset*, *minDurationOn*, and *minDurationOff* parameters on the Voxconverse test subset.

Table 1: *Detection Error Rate of the VAD model on Voxconverse test set.*

#Model	FA	MISS	Detection Error
#1	2.59%	1.40%	3.99%
#2	2.83%	2.09%	4.92%
#3	3.04%	1.74%	4.79%
#4	2.01%	1.19%	3.20%
Fusion	2.02%	0.82%	2.84%

2.1.5. Results

Table 1 shows that *NC-based* and *ASR-based* VAD models have inferior performance compared to systems trained with direct supervision. However, when we fuse these models using a majority vote, we achieve a reduction in detection error rate by 0.36%.

2.2. Speaker Embedding

We use several publicly available speaker embedding models, including TitaNet[3] [5], RawNet3[4] [6] and ECAPA-TDNN[5] [7]. Performance results of these models, along with the corresponding training datasets are presented in Table 2.

Table 2: *Equal Error Rate values for different embedding extraction models evaluated on the Voxceleb test benchmark.*

Embedding	EER	Training Datasets
TitaNet-Large	0.68% Vox1-Clean	Voxceleb1+Voxceleb2, Fisher, Switchboard, Librispeech
TitaNet-Small	1.08% Vox1-Clean	Voxceleb1+Voxceleb2, Fisher, Switchboard, Librispeech
RawNet3	0.89% Vox1-O	Voxceleb1+Voxceleb2
ECAPA-TDNN	0.80% Vox1-Clean	Voxceleb1+Voxceleb2

To increase the accuracy of speaker recognition and speaker diarization in noisy and reverberant environments, we finetune TitaNet-Small with Teacher-Student method [8] by adding L_2-regularization term to AAM loss [9], between embeddings for augmented and non-augmented versions of the same audio utterance. For fine-tuning we use VoxCeleb1 [10] and VoxCeleb2 [11] datasets. We apply noise and reverberation augmentations with MUSAN [12] and RIRs [13] corpus. The flow chart of teacher-student training is presented in Figure 1. By employing this approach, we achieved a comparable EER (1.03%) to the pre-trained TitaNet-Small model under normal conditions. However, the technique demonstrated superior performance in noisy conditions.

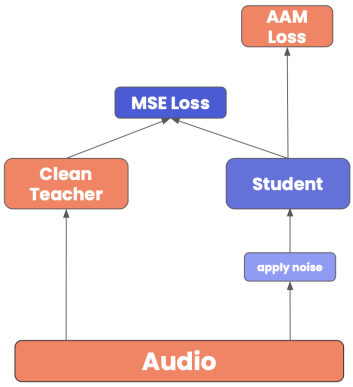

Figure 1: *The flow chart of teacher-student method for improving noise robustness, where the teacher is a pretrained TitaNet-Small model.*

2.3. Clustering

We use two different clustering algorithms for speaker diarization. One method relies on spectral clustering (SC) and another is based on agglomerative hierarchical clustering (AHC). Through our experiments, the spectral clustering achieves a lower DER and estimates the number of speakers more accurately, whereas agglomerative clustering is better at detecting the dominant speakers.

2.3.1. Spectral Clustering

Our SC based diarization is similar to [8]. We perform multiscale segmentation [14] and extract embeddings with different windows and shifts. The affinity matrices are constructed using the cosine similarity between segment embeddings and are then fused into a single matrix. We further apply the following sequence of refinement operations on the affinity matrix A:

- *Row-wise Thresholding*: For each row, keep *top-p* largest elements and set the rest to 0
- *Symmetrization*: $Y = \frac{1}{2}(A + A^T)$
- *Diffusion*: $Y = AA^T$

Afterwards, we apply the spectral clustering algorithm [15] to obtain speaker IDs. The number of speakers is determined using the maximal eigen-gap approach [16].

2.3.2. Agglomerative Hierarchical Clustering

First, we extract speaker embeddings from uniformly segmented speech regions. Then, we refine these embeddings through spectral dimensionality reduction[6] [17] and affinity aggregation (AA) [18] techniques. We merged consecutive segments into a longer one if the distance is greater than a *seg-*

[2] https://huggingface.co/pyannote/segmentation

[3] https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/titanet_large

[4] https://huggingface.co/jungjee/RawNet3

[5] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

[6] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.SpectralEmbedding.html

Table 3: *The performance of different speaker diarization systems.*

N	System	Window [s]	Shift [s]	Voxconverse Test	VoxSRC-23 Test	
				DER[%]	**DER[%]**	**JER[%]**
	VGG baseline	-	-	-	8.68	26.71
#1	Pyannote VoxSRC22	-	-	5.89	7.33	33.8
#2	Pyannote VoxSRC22+AA	-	-	5.30	-	-
#3	TitaNet-Large-SC	1.0	0.75	6.00	-	-
#4	TitaNet-Large-SC	2.0	1.0	5.59	-	-
#5	TitaNet-Large-SC	[2.0, 1.5, 0.75]	[1, 0.5, 0.25]	5.25	-	-
#6	ECAPA-TDNN-SC	1.0	0.75	6.05	-	-
#7	ECAPA-TDNN-SC	2.0	1.0	5.71	-	-
#8	ECAPA-TDNN-SC	[2, 1.5, 0.75]	[1, 0.5, 0.25]	5.38	-	-
#9	TitaNet-Small-SC	1.5	0.5	5.23	-	-
#10	TitaNet-Large-AHC	1.5	0.5	5.41	-	-
#11	ECAPA-TDNN-AHC	1.5	0.5	5.38	-	-
#12	RawNet3-AHC	1.5	0.75	5.32	-	-
	Fusion(3+4+5+6+7+8)+OSD	-	-	4.80	6.35	33.71
	Fusion(2+3+4+5+6+7+8)+OSD	-	-	4.76	5.98	31.56
	Fusion(2+5+8+9+10+11+12)+OSD	-	-	4.39	4.71	29.83

ment threshold. Afterwards, we perform a plain agglomerative clustering on the refined embeddings with a relatively high *stop threshold* to obtain the clusters with high confidence. The clusters from AHC were further processed using the short-duration filter [19, 20]. We categorize a cluster as "short" if the combined duration of that cluster is below the specified *duration threshold*. Later, each short cluster is assigned to the nearest long cluster based on the cosine distance of their central embeddings. Finally, if a short cluster significantly differs from all long clusters, which means that the distance between them is lower than a *speaker threshold*, we consider it as a new speaker.

2.4. Overlap Speech Detection

To detect regions where two or more speakers are speaking simultaneously, we use *pyannote overlap speech detection* pipeline[7] [21]. After an overlapped region is detected, we replace the label with the two closest speakers near this region in the time domain. The *onset* threshold for overlap decision is set to 0.91.

2.5. Fusion

We combine our diarization systems using the DOVER-Lap[8] [22] fusion method with the hungarian label mapping algorithm.

3. Experimental Results

Table 3 shows the results on the voxconverse test set and the challenge test set. We start with the pyannote VoxSRC22 pipeline $(\#1)$ as our initial system and enhance it by applying the affinity aggregation technique $(\#2)$ to refine the embeddings. This adjustment results in a reduction of 0.59% in DER on the voxconverse test set.

Next, we designed several diarization systems based on spectral clustering with different embedding extractors $(\#3 - \#9)$. These systems all rely on uniform speaker segmentation, which leads to speaker errors, mainly around the speaker turns. This occurs because segments with high resolution are very likely to contain speaker turn boundaries, while short segments carry insufficient speaker information. To mitigate this issue, we use different segmentation setups by changing both the window size and the shift size. Multi-scale segmentation $(\#5, \#8)$ is also designed to tackle this problem and to remove noisy entries from the affinity matrix. Furthermore, to make the systems more robust, we apply a sequence of refinement operations on the affinity matrix. In single-scale segmented setups, we establish the *top-p* value for row-wise thresholding as 8. In the case of multi-scale segmented setups, this value is adjusted to 30. As one can see from Table 3, multi-scale segmented systems outperform single-scale ones by a margin of 0.3%. Surprisingly, system $\#9$, which was finetuned with the Teacher-Student technique, achieves a similar score (5.23%) on the voxconverse test set without using multi-scale segmentation.

As noted in [20], SC-based and AHC-based clustering methods complement each other. Through our experiments, we also observed similar behaviour. Spectral clustering provides a more precise estimation of the number of speakers, whereas AHC-based clustering tends to consistently overestimate it. Conversely, AHC-based clustering excels at identifying the dominant speakers and demonstrates superior performance on shorter audio files compared to spectral clustering. We conduct a hyperparameter search for AHC-based systems $(\#10, \#11, \#12)$ on the voxconverse test subset to determine the optimal values for *segment threshold*, *stop threshold*, *duration threshold*, and *speaker threshold*. As it's illustrated in Table 3, AHC-based systems show slightly worse DER scores (5.32%-5.41%) compared to SC-based systems.

Our best system combines 7 different systems fused by DOVER-Lap. Among these, 3 systems are based on spectral clustering, while 4 systems are based on AHC (including pyannote system $\#2$). We fuse the systems first and then dealt

[7] https://huggingface.co/pyannote/overlapped-speech-detection

[8] https://github.com/desh2608/dover-lap

with the overlap, because fusing with overlapping labels did not demonstrate any improvement on the voxconverse test set. This fused system achieves 4.39% DER on the voxconverse test set and 4.71% DER on the challenge evaluation set, which ranks 2nd place in this challenge.

4. Conclusions

In this report, we described our submitted SD system for the diarization task of the 2023 VoxSRC challenge. Since we entered this contest for the first time, we have mainly focused on reducing speaker confusion errors. To achieve this, we used various methods such as multi-scale segmentation, affinity refinement operations, and teacher-student techniques, to make our SD systems robust with respect to background noise and errors that might arise from uniform speech segmentation. Our final system yielded notable results, reaching a DER of 4.39% on the voxconverse test set and 4.71% on the challenge evaluation set.

5. Acknowledgements

This work supported by Krisp.ai.

6. References

[1] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. [Online]. Available: https://aclanthology.org/W14-4012

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[3] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild," in *INTERSPEECH*, 2020.

[4] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[5] N. R. Koluguri, T. Park, and B. Ginsburg, "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8102–8106.

[6] J.-w. Jung, Y. J. Kim, H.-S. Heo, B.-J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," *Proc. Interspeech*, 2022.

[7] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech 2020*, 2020, pp. 3830–3834.

[8] D. S. Karamyan, G. A. Kirakosyan, and S. A. Harutyunyan, "Making speaker diarization system noise tolerant," *Mathematical Problems of Computer Science*, vol. 59, pp. 57–68, 2023.

[9] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[10] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[11] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[12] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[13] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[14] Y. Kwon, H.-S. Heo, J.-w. Jung, Y. J. Kim, B.-J. Lee, and J. S. Chung, "Multi-scale speaker embedding-based graph attention networks for speaker diarisation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8367–8371.

[15] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[16] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *2018 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[17] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[18] Y. Kwon, J.-w. Jung, H.-S. Heo, Y. J. Kim, B.-J. Lee, and J. S. Chung, "Adapting speaker embeddings for speaker diarisation," *arXiv preprint arXiv:2104.02879*, 2021.

[19] X. Xiao, N. Kanda, Z. Chen, T. Zhou, T. Yoshioka, S. Chen, Y. Zhao, G. Liu, Y. Wu, J. Wu *et al.*, "Microsoft speaker diarization system for the voxceleb speaker recognition challenge 2020," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5824–5828.

[20] W. Wang, X. Qin, M. Cheng, Y. Zhang, K. Wang, and M. Li, "The dku-smiip diarization system for the voxceleb speaker recognition challenge 2022," in *Voxsrc Workshop*, 2022.

[21] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, Brno, Czech Republic, August 2021.

[22] D.Raj, P.Garcia, Z.Huang, S.Watanabe, D.Povey, A.Stolcke, and S.Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021.