2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP) | 979-8-3503-9796-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISCSLP57327.2022.10037955

The X-Lance Speaker Diarization System for the Conversational Short-phrase Speaker Diarization Challenge 2022

Tao Liu[1], *Xu Xiang*[2], *Zhengyang Chen*[1], *Bing Han*[1], *Kai Yu*[1,*], *Yanmin Qian*[1,*]

[1]MoE Key Lab of Artificial Intelligence, AI Institute, X-LANCE Lab, Shanghai Jiao Tong University
[2]AISpeech Ltd, Suzhou China

{liutaw, zhengyang.chen, hanbing97, kai.yu, yanminqian}@sjtu.edu.cn,
{xu.xiang}@aispeech.com

Abstract

This paper describes X-Lance Speaker Diarization System submitted to the Conversational Short-phrase Speaker Diarization Challenge. The system outputs the ensemble results of the four modules: self-attentive-based VAD, uniform segmentation, ECAPA-TDNN-based embedding extractor, and spectral clustering. We evaluated our system on the Conversational Short-phrase Speaker Diarization (CSSD) dataset, which is based on MagicData-RAMC and contains plenty of conversational short-phrase segments. Besides being different from other diarization challenges, the challenge proposes a metric called Conversational Diarization Error Rate (CDER), which focuses on evaluating short segments. In this paper, we will analyze this metric and conduct extensive experiments. Finally, our system achieves CDER of 13.2% and 8.0% in the CSSD_dev and unseen CSSD_eval set, respectively.

Index Terms: speaker diarization, conversational, short-phrase

1. Introduction

Based on the short phase containing semantic information and current metric[1, 2] unable to measure short-phase segments, CSSD proposes a CDER metric and a short-phase dataset. This technical report describes the X-Lance system submitted to the Conversational Short-phrase Speaker Diarization Challenge[3, 4]. We adopt a modulized speaker diarization pipeline, also called the clustering-based method. The speaker diarization pipeline contains VAD, segmentation, embedding extractor, and clustering. In this paper, we organize our paper in the following order. First, we introduce the corpus used in our system. Second, we make a brief overview and analysis of CSSD. Third, detailed model configurations on each module are explained. Finally, we make conclusions and analyses on the result.

2. Data Resources

We will briefly introduce the corpus allowed to use in this challenge.

- VoxCeleb1 [5] contains over 100,000 utterances for 1,251 celebrities, extracted from videos uploaded to YouTube.
- VoxCeleb2 [6] contains over 1 million utterances for over 6,000 celebrities, also extracted from videos uploaded to YouTube.
- CNCeleb1 [7] specially focuses on Chinese celebrities, and contains more than 130,000 utterances from 1,000 persons.
- CNCeleb2 [8] publishes a new large-scale multi-genre corpus, called CN-Celeb2. CN-Celeb2 shares the same 11 genres as CN-Celeb1, but the data size is much larger. It contains over 520,000 utterances from 2,000 Chinese celebrities.
- MagicData-RAMC[3] is a rich annotated mandarin conversational speech dataset containing 180 hours of dialog speech. The dataset is divided into 149.65 hours of the training set, 9.89 hours of the development set, and 20.64 hours of the test set.
- CSSD dataset[4] is a conversational short-phrase speaker diarization. The dataset is based on MagicData-RAMC[3]. Before the final evaluating stage, the dataset is also split into three datasets, including training, dev, and test, which is directly converted from the ASR transcript of MagicData-RAMC. In the final evaluating state, the CSSD dataset provides an additional evaluation set for testing, which contains a 20-hour conversational speech test with careful annotations. The dataset splits and notations used in this paper are in Table 1.
- MUSAN [9] and RIRs[10]. MUSAN is a publicly available corpus comprising music, speech, and noise. RIRs[10] is a room impulse responses. MUSAN and RIRs are used as data augmentation in our speaker embedding module.

Table 1: *A brief overview and notions about the CSSD dataset that is based on MagicData-RAMC.* ***splits****: Official CSSD dataset split.* ***notation****: Notation used in this paper.* ***duration****: The total audio duration of the dataset split.* ***speech****: The speech occupation to the duration. Specifically, the eval split is unseen for participants before the final evaluation.*

dataset	splits	notation	duration	speech [%]
CSSD [11]	$train$	$CSSD_{train}$	149h39m	83.85
	dev	$CSSD_{dev}$	9h53m	82.79
	$test$	$CSSD_{test}$	20h38m	82.72
	$eval^*$	$CSSD_{eval}$	19h12m	72.43

3. CSSD Analysis

Before diving into our speaker diarization system, we briefly introduce the main features of the CSSD dataset and evaluation metric proposed in this challenge in this section.

[*] Kai Yu and Yanmin Qian are the corresponding authors.

979-8-3503-9796-3/22/$31.00 ©2022 IEEE

CSSD dataset[4] is a conversational short-phrase speaker diarization dataset. Different from the previous speaker diarization dataset, CSSD has three features. First, CSSD focuses on daily conversation, which contains plenty of spontaneous speech, some even not recognizable. Second, the CSSD dataset contains a large amount of short-phrase speeches. Third, different from widely-used diarization metric, DER[1] and JER[2], CSSD proposes a CDER for the final diarization evaluation. CDER, short for conversational diarization error rate, is designated for better evaluating short segments, where the result of DER and JER can not measure well. The pipeline in CSSD mainly contains merging, optimal mapping, and IoU matching. The merging operation is to merge adjacent segments with the same speakers. The optimal mapping step is the same as DER and JER, and the Hungarian algorithm is adopted here. The last step, IoU matching, is to count matching segments between the reference and the hypothesis. The matching stands when the IoU of a reference and a hypothesis is under a threshold(0.5 is used in this challenge). S_i and S_j represents two segments with speaker id i and j. Those two segments match if and only if Equation. 1 holds.

$$\frac{\text{Intersection}\left(S_i, S_j\right)}{\text{Union}\left(S_i, S_j\right)} \geqslant \text{ IoU threshold} \tag{1}$$

Other segments are all categorized into unmatched segments. We use notation unmatched_k to represent unmatched segments k. The final CDER can be calculated with Equation. 2. # unmatched and # reference represent the total number of $unmatched\_k$ and the total number of references, respectively.

$$\text{CDER} = \frac{\text{\# unmatched}}{\text{\# reference}} \tag{2}$$

After analyzing the metric, we get two conclusions. First, because of merging and IoU operation, CDER is less sensitive to segment boundaries, especially when the segment is long. Second, CDER is susceptible to speaker turns, especially in speaker confusion.

4. Detailed Model Configuration

4.1. Experiment setups

4.1.1. Dataset setups

CDER_dev is the development of CSSD. CDER_eval, different from CDER_test, is the final test set unseen before the challenge ends. Due to the original CSSD having $G00000000$, which represents unknown contents, we remove those segments on CDER_dev before we calculate the CDER, and there is no problem with CDER_eval. The occupation of $G00000000$ is 2.67% of the whole CSSD dataset. So a slight mismatch exists between CDER_dev and CDER_eval, and the baseline result is slightly different from the original baseline result. Besides, for simplicity, CDER_test is similar to CDER_dev, and we do not report CDER_test in our paper.

4.1.2. Ablation study setups

We make a grid search to get the best parameter for each module. But, for simplicity, we only report the ablation study result on a specific condition. For example, when we make the ablation study on the VAD threshold, the parameter for other modules, like segmentation, embedding extractors, and clustering, is the best-tuned result for each module.

4.2. VAD

The purpose of voice activity detectors(VAD) identifies speech and non-speech segments. From the Table 1, the dataset contains around 20% non-speech duration and VAD is important in this dataset. We adopt self-attentive VAD [12] for VAD, which has three main components: embedding layer and multi-head attention layer. Before feeding into the embedding layer, audio is converted to log-mel with 80 mel bins. The key component of the first embedding layer is a sinusoidal positional encoding layer which adds positional information to audio embedding. Then those embeddings are fed into a multi-head attention layer. In the experiment, we only use one head and two attention layers. So the final main architecture in the multi-head attention layer is a two-layer self-attention module. Finally, the cross-entropy loss is adopted in the training stage, and a threshold is set in the inference stage. By combining the positional coding module and self-attention module, we can better model short and long dependency, which is quite important in the CSSD dataset.

The experiment result on VAD is shown in Table. 2. Baseline VAD[1] uses the TDNN-Stats SAD model, trained on Chime-6[13] data. Different from Baseline VAD, we train our VAD model solely on the CSSD train set without data augmentation. The VAD label is directly converted from the ASR transcript by only keeping active speech duration($G00000000$ is kept here) and removing speaker identities. The experiment results are in Table. 2 shows that our VAD model reduces the false alarm rate from 12.53% to 6.47%, with only a slight worse in miss detection rate.

Table 2: *The false alarm (FA), miss detection (MISS) and detection error rate of the VAD model on CSSD dev set.*

Method	**FA [%]**	**MS [%]**	**Detection Error [%]**
Baseline VAD [4]	12.53	0.20	12.73
Self-attentive VAD [12]	6.47	1.33	**7.80**

On the VAD module, we also test several different VAD thresholds. From Table 3, we find that a higher VAD threshold can achieve a better result, and we use VAD threshold: 0.9 in our system. A higher VAD threshold reduces false alarms, leading to lower speaker confusion.

Table 3: *Ablation Study on VAD threshold*

VAD threshold	**FA [%]**	**MS [%]**	**Detection Error [%]**	**CDER_dev [%]**	**CDER_eval [%]**
0.5	6.47	1.34	7.81	11.7	11.1
0.6	4.86	2.57	7.44	11.7	10.7
0.7	3.50	4.79	8.29	12.3	10.3
0.8	2.34	9.06	11.4	12.6	9.7
0.9	1.36	15.61	16.97	13.2	**8.0**

4.3. Segmentation

Segmentation divides the speech segment, detected by VAD, into smaller segments. The duration of smaller segments is not fixed, and two hypotheses exist on those smaller segments. First, the segment should smaller enough so that there is only one person speaking in the segment. Second, there is no overlapped speech in the segments. Due to there being no overlap in the CSSD dataset, the second hypothesis stands. For the first hypothesis, the duration is a hyperparameter for our system, which is the trade-off between the time resolution and the accuracy of

[1] https://github.com/MagicHub-io/MagicData-RAMC

the embedding extractor. In our system, we find a sliding window of two seconds without overlap step is the optimal parameter in the CSSD dataset. If not specified, all experiments in this paper use this type of sliding strategy.

4.4. Embedding extractors

The embedding extractor extracts speaker-discriminative representation. The robust speaker embedding has higher inter-speaker distance and lower intra-speaker distance, typically reducing speaker confusion in speaker diarization. DNN-based speaker representations, like x-vector [14], can better capture speaker discriminative characteristics compared to traditional methods like GMM-UBM. ECAPA-TDNN [15] is an extended version of Time Delay Neural Network (TDNN) module. In our system, we adopt ECAPA-TDNN to extract speaker embedding.

In this challenge, we use VoxCeleb [5, 6] and CNCeleb [7, 8] to train our model. We use an online data augmentation strategy with noises and reverberation sampled from MUSAN [9] and RIRs [10] respectively. The training loss is the additive angular margin (AAM) softmax loss, and the margin is set to 0.2. Other unmentioned setups or parameters follows [16]. In the first training stage, we train our model on VoxCeleb2, CNCeleb1, and CNCeleb2, with 8994 speaker numbers in total. Then we evaluate on Vox-O, and the result is shown in Table 4. In the final evaluation stage, to get a better embedding extractor result, we train our model on all available speakers, with 10365 speaker numbers in total. The training strategy is the same as the previous training stage, and we use the result of the epoch 186.

Table 4: *The EER and minDCF for speaker embedding extractors via ECAPA-TDNN trained on VoxCeleb2, CNCeleb1, and CNCeleb2 (8994 speakers in total)*

Test set	EER [%]	minDCR [%]
Vox-O	0.97	0.0757

However, the discriminative ability of the speaker embedding will weaken as the input speech duration decreases. So, we conduct an ablation study to find the duration on which the embedding extractor fails. We set several skip duration parameters, which means segments less than such duration will skip without calculating the embedding to verify the best result. Table. 5 shows the ablation study for our experiment. The skip duration is set to 0.93 seconds, which achieves the final best CDER result on the evaluation set.

Table 5: *Ablation study on skip duration for embedding extractor.*

Skip Duration	CDER_dev [%]	CDER_eval [%]
0.6	14.2	10.4
0.7	12.9	9.7
0.8	13.2	8.9
0.9	13.4	8.2
0.93	13.2	**8.0**

4.5. Clustering

The clustering module is to cluster homogeneous segments with the same speaker. Our system tests two clustering methods: agglomerative hierarchical clustering[17](AHC) and spectral clustering[18](SC). We set two as the cluster number for AHC and SC because the speaker number is fixed to two. Other parameters are adjusted on CSSD_dev.

AHC. AHC is a bottom-up clustering method by iteratively merging the sample group with the shortest similarity. We adopt the cosine similarity metric to measure the distance, and the linkage is $complete$.

SC. The spectral cluster aims to expand the distance of inter graph group as large as possible and reduce the distance of the internal node in the same group as small as possible. A Laplace matrix is built to solve this problem. In our system, we use unnormalized graph Laplacian and construct the affinity matrix by cosine similarities. We only keep top 15 value in affinity matrix. Speakers are divided by the maximum gap of eigenvalue.

Table 6: *Ablation study on clustering methods.*

Clustering method	CDER_dev [%]	CDER_eval [%]
AHC[17]	14.4	15.9
SC[18]	13.2	**8.0**

Table 7: *Final results of our methods compared with the baseline.*

Clustering method	CDER_dev [%]	CDER_eval [%]
VBx (Baseline) [19]	21.6 †	26.5 ‡
ours	**13.2**	**8.0**

5. Results and analysis

Due to EEND[20] requiring large amounts of data and the corpus in this challenge is constrained, we utilize a clustering-based speaker diarization pipeline. The pipeline is a modulized pipeline, where modules are optimized dependently. We carefully tune the parameters in our system to achieve the best result. Compared with the baseline system, using TDNN-based VAD, x-vector-based embedding and AHC-VBx-based clustering, we make the following attempts. In the VAD module, we adopt a self-attentive-based module, achieving 4.93% absolute improvement on CSSD_dev set compared with the baseline. To best improve the embedding extractor performance, we use all the speakers permitted in this challenge to train a speaker discriminative embedding extractor via ECAPA-TDNN, achieving a SOTA result on the speaker verification task. In the clustering stage, we test two classic clustering methods, agglomerative hierarchical clustering and spectral clustering, on the dataset. Experiment result in Table 7 shows that our system is superior to the baseline system. Besides, our architecture is simple and without fusing module, which is convenient to reproduce.

6. Conclusions

In this paper, we reported the systems developed by the X-Lance team for the Conversational Short-phrase Speaker Di-

† The CSSD_dev result of the baseline in this table is slight different from the original baseline: 26.9 because we remove the ambiguous identity from the original transcript. The reason is illustrated in Section 4.1.1.

‡ The CSSD_test result of the baseline in this table is generated by the baseline code repo (https://github.com/MagicHub-io/MagicData-RAMC/tree/main/sd) with the original parameters.

arization Challenge. We adopt a clustering-based speaker diarization pipeline including several modules: self-attentive-based VAD, segmentation, ECAPA-TDNN-based embedding extractor, and spectral clustering. To better solve short-phrase settings, we conduct extensive experiments to get the optimal parameters on the dataset. Without any fusion strategy, our best submission on the CSSD evaluation set is 8.0% of CDER.

7. ACKNOWLEDGEMENTS

This work was supported by State Key Laboratory of Media Convergence Production Technology and Systems Project (No. SKLMCPTS2020003), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and National Natural Science Foundation of China (Grant No. 92048205).

8. References

[1] NIST, "The 2009 (rt-09) rich transcription meeting recognition evaluation plan," 2009.

[2] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second dihard diarization challenge: Dataset, task, and baselines," *arXiv preprint arXiv:1906.07839*, 2019.

[3] Z. Yang, Y. Chen, L. Luo, R. Yang, L. Ye, G. Cheng, J. Xu, Y. Jin, Q. Zhang, P. Zhang *et al.*, "Open source magicdata-ramc: A rich annotated mandarin conversational (ramc) speech dataset," *arXiv preprint arXiv:2203.16844*, 2022.

[4] G. Cheng, Y. Chen, R. Yang, Q. Li, Z. Yang, L. Ye, P. Zhang, Q. Zhang, L. Xie, Y. Qian *et al.*, "The conversational short-phrase speaker diarization (cssd) task: Dataset, evaluation metric and baselines," *arXiv preprint arXiv:2208.08042*, 2022.

[5] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[6] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Proc. Interspeech 2018*, pp. 1086–1090, 2018.

[7] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[8] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vipperla, T. F. Zheng, and D. Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, vol. 137, pp. 77–91, 2022.

[9] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[10] 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.

[11] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild," in *International Speech Communication Association (Interspeech)*, 2020.

[12] Y. R. Jo, Y. K. Moon, W. I. Cho, and G. S. Jo, "Self-attentive vad: Context-aware detection of voice from noise," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6808–6812.

[13] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj *et al.*, "Chime-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.

[14] 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.

[15] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[16] R. K. Das, R. Tao, and H. Li, "Hlt-nus submission for 2020 nist conversational telephone speech sre," *arXiv preprint arXiv:2111.06671*, 2021.

[17] F. Nielsen, "Hierarchical clustering," in *Introduction to HPC with MPI for Data Science*. Springer, 2016, pp. 195–211.

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

[19] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian hmm clustering of x-vector sequences (vbx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, p. 101254, 2022.

[20] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7198–7202.