TOWARDS END-TO-END SPEAKER DIARIZATION WITH GENERALIZED NEURAL SPEAKER CLUSTERING

Chunlei Zhang[1], *Jiatong Shi*[2], *Chao Weng*[1], *Meng Yu*[1], *Dong Yu*[1]

[1]Tencent AI Lab, Bellevue, USA [2]Carnegie Mellon University, USA

{cleizhang, cweng, raymondmyu, dyu}@tencent.com, jiatongs@cs.cmu.edu

ABSTRACT

Speaker diarization consists of many components, e.g., front-end processing, speech activity detection (SAD), overlapped speech detection (OSD) and speaker segmentation/clustering. Conventionally, most of the involved components are separately developed and optimized. The resulting speaker diarization systems are complicated and sometimes lack of satisfying generalization capabilities. In this study, we present a novel speaker diarization system, with a generalized neural speaker clustering module as the backbone. The whole system can be simplified to contain only two major parts, a speaker embedding extractor followed by a clustering module. Both parts are implemented with neural networks. In the training phase, an on-the-fly spoken dialogue generator is designed to provide the system with audio streams and the corresponding annotations in categories of non-speech, overlapped speech and active speakers. The chunk-wise inference and a speaker verification based tracing module are conducted to handle the arbitrary number of speakers. We demonstrate that the proposed speaker diarization system is able to integrate SAD, OSD and speaker segmentation/clustering, and yield competitive results in the VoxConverse20 benchmarks.

Index Terms— speaker diarization, speaker embedding, speaker clustering, overlap speech detection

1. INTRODUCTION

Speaker diarization is the process of automatically detecting "who spoke when" in an audio stream [1,2]. It becomes an important technology in many applications such as speaker retrieval in audio/video streams, meeting transcriptions, and conversation analysis etc [3–6]. As a sequential process, speaker diarization normally involves several modules such as front-end processing (e.g., speech enhancement or speech separation), speech activity detection (SAD), segmentation/speaker change detection, speaker clustering, and post-processing (e.g., re-segmentation, fusion) [7–14].

With the advancements being made in many individual speech processing techniques, increasing efforts have been made forward to more practical speech systems. Speaker diarization is such a topic that achieves rapid progress recently [2, 14–16]. The development of robust speaker representations firstly excited the research interests with clustering based methods [17, 18]. Subsequently, Diez et al. proposed a Variational Bayes Hidden Markov Model (VB-HMM) based clustering method to handle unknown number of speakers, and they also found a better speaker diarization result when VB-HMM was employed as a re-clustering procedure [10, 13]. Li et al. formulated the unsupervised speaker clustering problem to a supervised learning scheme, where a Transformer based encoder-decoder model was applied to map speaker embeddings to their relative positions [19]. To consider overlapped speech, Bullock et al. proposed to perform overlap-aware resegmentation, which improved the clustering based baseline on AMI dataset [20]. Target-Speaker Voice Activity Detection (TS-VAD) was proposed to act as a post-processing module after the initial speaker clustering. With iterative i-vector extraction and segmentation refinement, impressive diarization performance was achieved in the latest CHiME challenge [12]. In VoxSRC 2020, the winning system applied continues speech separation (CSS) as the front-end processing, which is then followed by a standard speaker diarization pipeline [14, 16].

Although widely investigated, there still remain some challenges in developing speaker diarization systems: a) individual components that are separately optimized, module coordination is complicated during inference; b) to address unknown speaker number and overlapped speech, compensation techniques (e.g., speech separation, iterative re-segmentation/re-clustering etc.) have to be added, which makes the overall system unlikely to be small-footprint. In that context, Region Proposal Networks (RPN) was proposed to jointly perform segmentation, speaker embedding extraction, and re-segmentation by a single neural network [21], while leaving the speaker clustering as a stand-alone process. Unbounded Interleaved-State Recurrent Neural Networks (UIS-RNN) and Discriminative Neural Clustering (DNC) were proposed that replace the segmentation and clustering procedure by a supervised learning process [19, 22]. Compared with traditional clustering counterparts, both methods have illustrated performance boosts when there are no domain mismatches. However, SAD and overlapped speech were not considered in their systems. Recently, the framework noted as End-to-End Neural Diarization (EEND) has drawn much attention [23, 24]. By supervised Permutation Invariant Training, the system could directly produce speaker diarization result in two-speaker scenarios. Subsequently, EEND was extended to flexible number of speakers with the encoder-decoder based attractor (EDA) and speaker-wise conditional model (SC-EEND) [25, 26]. Online inference was realized by introducing the speaker-tracing buffer (STB) [27]. While improved performance has been achieved in CALLHOME test set, challenges remains when the speaker number increases, leading to a more serious chunk-wise permutation issue.

In this study, we investigate a novel end-to-end speaker diarization (EESD) framework from the perspective of neural speaker clustering. Inspired by the DNC [19], we propose a more generalized neural speaker clustering method that can perform SAD, OSD and speaker segmentation/clustering simultaneously. To facilitate the training of supervised neural clustering, we implement an on-the-fly spoken dialogue generator that can simulate dialogues with non-speech, overlapped speech and active speaker labels. We also explore a speaker verification based tracing module to handle flexible number of speakers. The proposed EESD is evaluated on the VoxConverse20 dataset [28]. We show promising speaker diarization results with a reduced inference cost. Note that this paper serves as

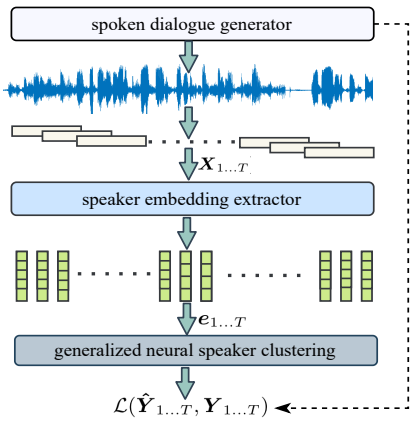

Fig. 1. A flow diagram of EESD.

a preliminary research work and it is flexible to incorporate other techniques to further improve speaker diarization performance.

The rest of the paper is organized as follows. In Sec.2, we first introduce the EESD framework, then we present the details of each component. We describe our experimental setups and evaluate the proposed system in Sec.3. We discuss the future directions for EESD and conclude this work in Sec.4.

2. THE PROPOSED EESD SYSTEM

2.1. The EESD overview

The overall training diagram is depicted as Fig.1. At the beginning of training system, we have a generator that simulates spoken dialogues and corresponding labels on-the-fly. For that purpose, we need to prepare a corpus with speaker and SAD labels. Once the dialogue audio is generated, it is framed and segmented into consecutive 2D acoustic feature snippets $\boldsymbol{X}_i \in \mathrm{R}^{\mathrm{T}\times\mathrm{F}}, i = 1...N$ for embedding extraction, where T the frame number in each snippet, F is the acoustic feature dimension and N is the total feature snippets in a dialogue. The speaker embedding extractor is pretrained in advance. Then, the speaker embedding $\boldsymbol{e}_i \in \mathrm{R}^{1\times\mathrm{D}}, i = 1...N$ is fed into the generalized neural speaker clustering module to predict labels $\hat{\boldsymbol{Y}}_{\boldsymbol{i}} \in \mathrm{R}^{1\times\mathrm{C}}, \mathrm{i} = 1...\mathrm{N}$, where C is the maximum class number designed in the neural clustering module. Finally, we calculate the loss between $\hat{\boldsymbol{Y}}$ and ground truth label $\boldsymbol{Y}$ and update the network.

For inference, we chunk the audio stream into a fixed length to generate prediction $\hat{\boldsymbol{Y}}$. The chunk-wise prediction $\hat{\boldsymbol{Y}}$ and speaker embedding $\boldsymbol{e}$ is further processed in the speaker-tracing and post-processing to handle unknown speaker number and long audio streams encountered in speaker diarization.

2.2. On-the-fly spoken dialogue generator

As a key procedure to controls what is learned with EESD, it is critical for the generator to simulate realistic dialogues. A general generation process is described in Algorithm 1. Given a set of speakers $\mathcal{S}$, we first produce its frame-level SAD labels $\mathcal{L}$. For each audio mixture and label pair $(\boldsymbol{X}, \boldsymbol{Y})$, we randomly sample N_{spk} speakers with N_{utt} utterances per speaker to formulate a speaker subset $(\mathcal{S}', \mathcal{L}')$. We start the generation process by adding the first utterance of $\mathcal{S}'$ and its corresponding SAD labels to $(\boldsymbol{X}, \boldsymbol{Y})$. Note that original SAD labels only contain "0" and "1", indicating non-speech and speech respectively. In order to facilitate speaker clustering, we convert all "1" labels to its *absolute* speaker labels with $\mathcal{F}_{S2S}$ (i.e., speech to speaker labels) and keep all the "0" label unchanged. For the following utterances in the queue $\mathcal{S}'$, we employ a random variable L to control the overlap length between current utterance $\mathcal{S}'[u]$ and $\boldsymbol{X}$, where L is determined by $min(d, 0)$. The random variable

Algorithm 1: Spoken dialogue generator

Input: $\{\mathcal{S}, \mathcal{L}\}$ // set of spk, SAD labels
Output: $\{\boldsymbol{X}, \boldsymbol{Y}\}$ // dialogue, EESD labels
/* Initialization: */
Sample N_{spk} spk with N_{utt} utt/spk to subset $\mathcal{S}'$,
/* $N_{spk} \in [2, 4], N_{utt} \in [2, 5]$ */
$(\mathcal{S}', \mathcal{L}') \leftarrow (\mathcal{S}, \mathcal{L})$, shuffle $(\mathcal{S}', \mathcal{L}')$
$\boldsymbol{X}$.add$\{\mathcal{S}'[0]\}$, $\boldsymbol{Y}$.add$\{\mathcal{F}_{S2S}(\mathcal{L}'[0])\}$
/* Accumulate spoken dialogue: */
for $u = 1$ *to* $N_{S'}$ // $N_{S'}$: number of utt in $\mathcal{S}'$
do
  $L = -min(d, 0)$ // overlap length
  /* L = len$(\mathcal{S}'[u] \cap \boldsymbol{X})$, $d \sim N(\mu, \sigma^2)$ */
  $\boldsymbol{X}$.add$\{\mathcal{S}'[u]\}$, $\boldsymbol{Y}$.add$\{\mathcal{F}_{S2S}(\mathcal{L}'[u])\}$
  if len$(\boldsymbol{X}) \geq$ 60s **then**
    $\boldsymbol{X} = \mathrm{Chunk}(\boldsymbol{X}), \boldsymbol{Y} = \mathrm{Chunk}(\boldsymbol{Y})$
$\boldsymbol{Y} \leftarrow$ EESD_labeling$(\boldsymbol{Y})$ // label mapping

d follows a Gaussian distribution parameterized by mean μ (-16000 by default, i.e., one second overlap) and standard deviation σ (16000 by default), which is tuned to control the overlapping rate in the generated audio. To enable batch processing in GPUs, we have to constrain the length of each dialogue. Here, we set 60s as the maximum length, and stop the generation process once the limit is reached. To this end, we have created a spoken dialogue with non-speech, overlapped speech and speaker labels. Finally, we convert the *absolute* speaker labels to *relative* EESD speaker labels (Fig.2), which is required in the generalized neural speaker clustering.

2.3. Speaker embedding extractor

Traditional speaker diarization systems rely on a robust speaker embedding extractor to ensure accurate speaker clustering. In order to provide consistent feature representations (not conventional single-speaker embedding since non-speech and overlapped speech are also included in the spoken dialogue) for sequential neural clustering, we also employ speaker embedding extractors and investigate their impact on the speaker diarization performance in EESD.

We evaluate three speaker embedding models in this study, i.e., a TDNN, a Resnet-34 and a ECAPA-TDNN, which represent the most popular speaker embedding architectures in recent studies [17, 29, 30]. The training corpus is Voxceleb 2, 1-fold data augmentation is applied to incorporate more environmental dynamics [31, 32]. The speaker embedding model is trained by a multi-task loss, which employs both the large margin cosine loss (LMCL) and the triplet loss [33, 34]. 40-D log Mel filterbank features are extracted with a 32ms window and the time shift of feature frames is 16ms. The utterance is randomly segmented into 100-200 frames to control the duration variability in the training phase. The speaker embedding is a L_2 normalized 128D vector. It is noted that we utilize a 2D instance normalization to normalize feature snippets before the network mapping. We find a better clustering stability than the traditional frame-level mean normalization since the dialogue is simulated from multiple sessions (indicating a more dynamic variability across time in the dialogue). To better understand the speaker embedding models, we list the results on Voxceleb1 test set in Table 1.

Table 1. *Speaker verification results with three models.*

model	EER	size
TDNN	1.97%	3.2 M
Resnet-34	1.33%	8.0 M
ECAPA-TDNN	0.93%	14.7 M

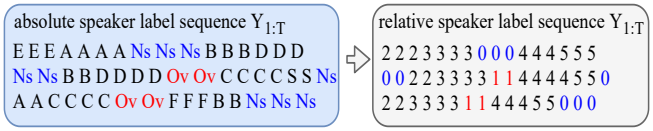

Fig. 2. EESD labeling examples.

2.4. Generalized Neural speaker clustering

Given the speaker embeddings $\boldsymbol{e}$ extracted from the whole dialogue, the objective for the generalized neural speaker clustering becomes to map single-speaker embedding (representing $absolute$ speaker information) to its $relative$ cluster ID, while retaining straight forward mapping for non-speech ("Ns") to "0" and overlapped speech ("Ov") to "1". Incorporating the DNC labeling method, we are arrived at the EESD labeling method for generalized neural speaker clustering, as illustrated in Fig.2. The generalized neural speaker clustering requires segment level labels per embedding, two different methods are explored to convert frame-level labels to segment-level ones. The first one is one-hot encoding, which operates $argmax$ to find the most frequent labels in T consecutive frames to represent the embedding. The second one is k-hot encoding, which maps the frequencies to soft probabilities (noted as label smoothing).

It's natural to choose maximum likelihood (ML) as the training objective in this sequential labeling model. The specific type can be flexible according to the chosen networks. In [19], the authors formulate the ML as an auto-regressive way $p(\boldsymbol{Y}_{1:T}|\boldsymbol{X}_{1:T}) = \prod_{i=1}^{T} p(y_i|y_{0:i-1}, \boldsymbol{X}_{1:T})$ with an encoder-decoder Transformer model. If we choose to support a streaming mode, the probability assignment can be $p(\boldsymbol{Y}_{1:T}|\boldsymbol{X}_{1:T}) = \prod_{i=1}^{T} p(y_i|\boldsymbol{X}_{1:i})$, where a Long Short-Term Memory (LSTM) network can be utilized in this situation. Here, we choose a bidirectional LSTM (BLSTM) as our preliminary try since BLSTM is a simple yet effective sequential model that enables better model power than the unidirectional LSTM. The probability model is $p(\boldsymbol{Y}_{1:T}|\boldsymbol{X}_{1:T}) = \prod_{i=1}^{T} p(y_i|\boldsymbol{X}_{1:T})$. Decode such a model is simple, one can either apply greedy search or beam-search to find the clustering result.

2.5. Causal speaker-tracing

In order to have an efficient neural clustering training, we set a maximum cluster number C, which constrains the EESD system to process long recordings with unknown speaker numbers. In this section, we propose a speaker verification based speaker-tracing module to solve the issue. Given a long recording, we can split it to fixed-length chunks, neural clustering is performed for each chunk, $\boldsymbol{e}^i$ is the speaker embedding and $\hat{\boldsymbol{Y}}^i$ is the decoding result in chunk i (both $\boldsymbol{e}^i$ and $\hat{\boldsymbol{Y}}^i$ are with length T). A high-level description of speaker-tracing is presented in Algorithm 2. An empty dictionary spk_dict with key to represent cluster ID and value to represent corresponding speaker embedding is initialed. The core part in Algorithm 2 is the SV function, which is used to decide whether $\boldsymbol{E}[j]$ should be assigned to a new cluster ID or to an existing cluster ID in the spk_dict. The decision is made by calculating the cosine distance between $\boldsymbol{E}[j]$ and existing cluster centroid. If the minimum distance is less than a predefined threshold Th=0.5, the j^{th} segment in chunk i is assigned to the existing cluster ID. Otherwise, we assign a new cluster ID to segment j. The spk_dict and $\hat{\boldsymbol{Y}}^i$ is updated.

2.6. Post-processing

The proposed EESD system is a framework that provides joint segmentation and clustering. In Sec.2.5, the causal speaker-tracing is conducted at the segment level, which is not reliable if the segment duration is short. In practice, the causal speaker-tracing tends to over-estimate the number of speakers. To alleviate this problem, we apply a simple SV based post-processing method to suppress the over-estimated speakers, which works as a non-causal offline compensation to EESD. To mitigate the issue, we formulate a post-processing as a multi-enrollment speaker verification problem. Specifically, we first split the whole cluster ID set into 2 subsets: (1) cluster IDs whose total duration is less than a certain threshold D (10s); (2) rest of cluster IDs. The small-duration cluster ID subset is treated as the test speakers, while the large-duration cluster ID subset is used for enrollments. We use the same SV threshold Th to decide if the "test" cluster ID should be merged to "enrolled" clusters. The merged embedding will be used to the "enrollment" speaker embedding. This process is repeated until no more cluster merge happens. We note this refinement as the duration based agglomerative hierarchical clustering (D-AHC). Practically, the D-AHC is as effective as the normal global AHC with a reduced computational cost.

Algorithm 2: Causal speaker-tracing

Input: $\{\boldsymbol{e}^i, \hat{\boldsymbol{Y}}^i, \hat{\boldsymbol{Y}}^{0:i-1}, \text{spk\_dict}\}$
/* Input for chunk i */
Output: $\{\hat{\boldsymbol{Y}}^{0:i}, \text{spk\_dict}\}$ // uodate to i
/* Initialization: */
find mean embd $\boldsymbol{E} \in \mathrm{R}^{J\times D}$ with J-1 spk change points in $\hat{\boldsymbol{Y}}^i$, EESD segmentation index $(\text{Idx}_j^s, \text{Idx}_j^e)$,
/* speaker-tracing in chunk i: */
for $j = 1$ *to* J **do**
  if $\hat{\boldsymbol{Y}}^i[\text{Idx}_j^s] \geq 1$ **then**
    /* only tracing single-spk */
    spk_dict, rewrite_id = SV ($\boldsymbol{E}[j]$,spk_dict,Th)
    $\hat{\boldsymbol{Y}}^i[\text{Idx}_j^s : \text{Idx}_j^e]$ = rewrite_id
$\hat{\boldsymbol{Y}}^{0:i} = \hat{\boldsymbol{Y}}^{0:i-1} + \hat{\boldsymbol{Y}}^i$ // append updated $\boldsymbol{Y}^i$

3. EXPERIMENTS

3.1. Data

Voxceleb dataset provides diverse choices of speakers in the wild, which is considered to be close to the VoxConverse dataset in acoustic conditions. In this study, both speaker embedding extractor and generalized neural speaker clustering utilize Voxceleb 2 (6K speakers) in the training stage. The generated spoken dialogue is augmented by MUSAN dataset and RIRS_NOISES with a balanced "reverberation", "noise", "music" and "babble" distribution [35, 36]. The Voxceleb 1 test set is employed as the validation set. We report the speaker diarization performance of EESD systems on VoxConverse20 development and test set.

3.2. Configurations of EESD

Given the EESD procedures in the previous sections, the actual hyperparameters are summarised as follows:

Speaker embedding extractor: We use 40×40 feature snippets as the input to the extractor, the shift between two consecutive snippets is 10 frames. Pretrained speaker embedding extractor is froze throughout the entire training process.

SAD: One strong SAD model is investigated to provide frame-level speech/non-speech labels, which a TDNN model trained with Fisher English and its forced alignment (SAD_{tdnn}).

Generalized neural speaker clustering: The neural clustering module takes 128D speaker embedding $\boldsymbol{e}$ as the input and expand the dimension to 256D with a fully connected layer. Following with 4 BLSTM layers (256D each), the final clustering is conducted by a softmax layer with C ($C = 6$, with N_{spk}=4 as the maximum speakers in the spoken dialogue generator) output classes. We utilize an Adam optimizer with an initial learning rate of 1e-3, the learning rate is annealed in half when the loss of validation set is not decreased.

3.3. Kaldi baseline SD system

We employ Kaldi DIHARD V2 recipe as the baseline system because of the same training data (i.e., Voxceleb 2) and similar clustering algorithm (i.e., AHC) is utilized [37]. The difference between our implementation and the Kaldi recipe remains in the speaker embedding extractor (ECAPA-TDNN V.S. standard TDNN) and the SAD model ($\mathrm{SAD}_{\mathrm{tdnn}}$ V.S. energy based SAD).

3.4. Results

We report our results on both VoxConverse20 Dev and test set to show how individual component performs in EESD and its impact to speaker diarization. For SAD and OSD, the results are based on the ECAPA-TDNN embedding extractor, we report accuracy, false alarm error (FA) and miss error (MI) for SAD. For speaker diarization, we use Diarization Error rate (DER) as the system metric.

Table 2. Frame-level SAD V.S. EESD SAD (non-s>0.6s, in %).

	$\mathbf{SAD_{tdnn}}$			**EESD SAD**		
test set	Acc	FA	MI	Acc	FA	MI
VoxCon Dev	95.3	3.5	1.2	95.7	3.4	0.9
VoxCon Test	94.4	4.3	1.3	94.5	4.5	1.0

3.4.1. EESD speech activity detection

As illustrated in Table 2, in contrast with the frame-level $\mathrm{SAD}_{\mathrm{tdnn}}$ system, the the EESD SAD achieves comparable performance in speech detection for both VoxConverse20 dev and test set. It is worth to note that the data augmentation at the spoken dialogue level is crucial to reduce the FA for VoxConverse20 dev & test set. As in both datasets, there are many audio segments with background music/noise being annotated as non-speech, which causes a performance degradation with the clean version of EESD SAD. By data augmentation, we are able to compensate for this portion of loss.

3.4.2. EESD overlap speech detection

As we described in Sec.2.2, the average overlap length L (controlled by (μ, σ) is the key factor that influences the OSD in the EESD system. As overlapped speech ratio is only around 3% for VoxConverse20 dev & test set, we report precision and recall. Three different (μ, σ) parameter pairs are investigated in Table 3. As we can see, generating more overlaps in the training helps to find the actual overlapped speech. However, when we increase the average overlapping rate during training, the precision rate is declined quickly. From the relatively low recall rate, we argue that OSD is very difficult, mainly because the overlapped speech in the speaker embedding space is still very close to the individual speakers within the mixture. At the same time, adding more overlapped speech in the simulation will not necessary help the overall OSD performance. From Table 3, $L = 1.3$s gives the best OSD performance. To unitize OSD result for EESD, we apply the same heuristic as [13], which considers the two closest speakers in time to produce diarization labels.

Table 3. *OSD performance in EESD with different overlap generation controllers (in %).*

(μ, σ)	L	precision	recall
(-4000,16000)	0.8s	72.4	27.6
(-16000,16000)	1.3s	67.3	34.8
(-20000,20000)	1.6s	60.5	36.3

3.4.3. Speaker diarization results

Finally, we list the SD results of our proposed EESD systems in Table 4, and point out the best performed configurations. From Table 4, our best model is able to outperform the audio-visual speaker diarization system baseline on dev set, and even create a bigger margin on test set [16, 28], which indicates that our proposed system has a good generalization capability r.w.t. different domains. Compared with the traditional pipeline system (i.e., Kaldi baseline, with the same SAD and speaker embedding extractor), there remains a small gap. Meanwhile, the gap becomes larger when comparing to the state-of-the-art (SOTA) systems reported in VoxSRC 2020 [13, 14]. However, The SOTA systems are much more complicated with individual OSD, speech separation or re-clustering/segmentation techniques etc. In this study, we are trying to provide an alternative way to conduct speaker diarization. By achieving a promising system performance with a simplified inference pipeline, we want to stimulate more flavors in continuing to improve the challenging speaker diarization tasks.

Table 4. *SD results with different configurations (in %).*

spk embd	label smo.	L	D-AHC	DER (dev)	DER (test)
TDNN	No	1.6s	No	12.5	16.7
TDNN	Yes	1.3s	No	11.6	15.2
TDNN	Yes	1.3s	Yes	9.4	12.5
Resnet-34	Yes	1.3s	Yes	8.6	11.6
ECAPA-TDNN	Yes	1.3s	Yes	7.3	10.9
Kaldi (improved)				6.3	9.8
VoxCon [16, 28]				7.7	21.8
BUT [13, 16]				4.0	8.1(fusion)
Microsoft [14]				3.8	6.2(fusion)

4. DISCUSSIONS AND CONCLUSIONS

In this study, we proposed a novel end-to-end speaker diarization system. By handling SAD, OSD and speaker segmentation/clustering with the generalized neural speaker clustering, the speaker diarization pipeline was largely simplified. The performance of EESD was further improved with the exploration in major components, such as on-the-fly spoken dialogue generator, speaker embedding extractor and post-processing. We believe that this paper shows the potential towards end-to-end speaker diarization. Meanwhile, there are still a lot of room for improvements. For example, the high FA (non-speech being predicted as speech) errors in high-volume music/noise condition can be improved using a speaker embedding extraction module. The complex pattern of overlapped speech is still under investigation. How effective online speaker diarization is supported with the current EESD framework retains to be an interesting topic. The results here show both meaningful advancements, as well as a direction for the future research.

5. REFERENCES

[1] X. Anguera, S. Bozonnet, and et al., "Speaker diarization: A review of recent research," *IEEE Trans. on Aud., Spe., and Lan. Pro.*, vol. 20, no. 2, pp. 356–370, 2012.

[2] T. J. Park, N. Kanda, and et al., "A review of speaker diarization: Recent advances with deep learning," *arXiv preprint :2101.09624*, 2021.

[3] Marijn H., *Segmentation, diarization and speech transcription: Surprise data unraveled*, Ph.D. dissertation, Univ. Twente, The Netherlands, 2008.

[4] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. on Aud., Spe., and Lan. Pro.*, vol. 15, no. 7, pp. 2011–2022, 2007.

[5] D. A Reynolds and P Torres-C., "Approaches and applications of audio diarization," in *IEEE ICASSP*, 2005, vol. 5, pp. v–953.

[6] T. Yoshioka, I. Abramovski, C. Aksoylar, and et al., "Advances in online audio-visual meeting transcription," in *2019 IEEE ASRU*. IEEE, 2019, pp. 276–283.

[7] L. Sun, J. Du, and et al., "Speaker diarization with enhancing speech for the first dihard challenge.," in *Interspeech*, 2018, pp. 2793–2797.

[8] D. Raj, P. Denisov, Z. Chen, and et al., "Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis," in *2021 IEEE SLT*, 2021, pp. 897–904.

[9] T. Pfau, D. PW Ellis, and A. Stolcke, "Multispeaker speech activity detection for the icsi meeting recorder," in *IEEE ASRU*, 2001, pp. 107–110.

[10] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Cernockỳ, "Bayesian hmm based x-vector clustering for speaker diarization.," in *INTERSPEECH*, 2019, pp. 346–350.

[11] D. Garcia-Romero, D. Snyder, and et al., "Speaker diarization using deep neural network embeddings," in *2017 IEEE ICASSP*, 2017, pp. 4930–4934.

[12] I. Medennikov, M. Korenevsky, and et al., "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," *arXiv preprint:2005.07272*, 2020.

[13] F. Landini, O. Glembek, P. Matějka, and et al., "Analysis of the but diarization system for voxconverse challenge," *arXiv preprint arXiv:2010.11718*, 2020.

[14] X. Xiao, N. Kanda, and et al., "Microsoft speaker diarization system for the voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2010.11458*, 2020.

[15] G. Sell, D. Snyder, and et al., "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge.," in *Interspeech*, 2018, pp. 2808–2812.

[16] A. Nagrani, J. S. Chung, and et al., "Voxsrc 2020: The second voxceleb speaker recognition challenge," *arXiv preprint arXiv:2012.06867*, 2020.

[17] D. Snyder, D. Garcia-Romero, and et al., "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE ICASSP*, 2018, pp. 5329–5333.

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

[19] Q. Li, F. L Kreyssig, C. Zhang, and P. C Woodland, "Discriminative neural clustering for speaker diarisation," in *2021 IEEE SLT*, 2021, pp. 574–581.

[20] L. Bullock, H. Bredin, and L. P. Garcia-Perera, "Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection," in *2020 IEEE ICASSP*, 2020, pp. 7114–7118.

[21] Z. Huang, S. Watanabe, and et al., "Speaker diarization with region proposal network," in *2020 IEEE ICASSP*, 2020, pp. 6514–6518.

[22] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *2019 IEEE ICASSP*, 2019, pp. 6301–6305.

[23] Y. Fujita, N. Kanda, and et al., "End-to-end neural speaker diarization with permutation-free objectives," *arXiv preprint arXiv:1909.05952*, 2019.

[24] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE ASRU*, 2019, pp. 296–303.

[25] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," *arXiv preprint arXiv:2005.09921*, 2020.

[26] Y. Fujita, S. Watanabe, S. Horiguchi, Y. Xue, J. Shi, and K. Nagamatsu, "Neural speaker diarization with speaker-wise chain rule," *arXiv preprint arXiv:2006.01796*, 2020.

[27] Y. Xue, S. Horiguchi, Y. Fujita, S. Watanabe, P. García, and K. Nagamatsu, "Online end-to-end neural diarization with speaker-tracing buffer," in *2021 IEEE SLT*, 2021, pp. 841–848.

[28] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild," *arXiv preprint arXiv:2007.01216*, 2020.

[29] Y. Kwon, H. Heo, B. Lee, and J. S. Chung, "The ins and outs of speaker recognition: lessons from voxsrc 2020," *arXiv preprint arXiv:2010.15809*, 2020.

[30] 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.

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

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

[33] C. Zhang, K. Koishida, and J. HL Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Trans. on Aud., Spe., and Lan. Pro.*, vol. 26, no. 9, pp. 1633–1644, 2018.

[34] C. Zhang, M. Yu, C. Weng, and D. Yu, "Towards robust speaker verification with target speaker enhancement," *arXiv preprint arXiv:2103.08781*, 2021.

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

[36] T. Ko, V. Peddinti, and et al., "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE ICASSP*, 2017, pp. 5220–5224.

[37] D. Povey, A. Ghoshal, G. Boulianne, and et al., "The kaldi speech recognition toolkit," in *IEEE ASRU*, 2011.