Interspeech 2022
18-22 September 2022, Incheon, Korea

From Simulated Mixtures to Simulated Conversations as Training Data for End-to-End Neural Diarization

Federico Landini[1], *Alicia Lozano-Diez*[2], *Mireia Diez*[1], *Lukáš Burget*[1]

[1]Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia
[2]AUDIAS (Audio, Data Intelligence and Speech), Universidad Autónoma de Madrid, Spain

{landini,mireia,burget}@fit.vutbr.cz, alicia.lozano@uam.es

Abstract

End-to-end neural diarization (EEND) is nowadays one of the most prominent research topics in speaker diarization. EEND presents an attractive alternative to standard cascaded diarization systems since a single system is trained at once to deal with the whole diarization problem. Several EEND variants and approaches are being proposed, however, all these models require large amounts of annotated data for training but available annotated data are scarce. Thus, EEND works have used mostly simulated mixtures for training. However, simulated mixtures do not resemble real conversations in many aspects. In this work we present an alternative method for creating synthetic conversations that resemble real ones by using statistics about distributions of pauses and overlaps estimated on genuine conversations. Furthermore, we analyze the effect of the source of the statistics, different augmentations and amounts of data. We demonstrate that our approach performs substantially better than the original one, while reducing the dependence on the fine-tuning stage. Experiments are carried out on 2-speaker telephone conversations of Callhome and DIHARD 3. Together with this publication, we release our implementations of EEND and the method for creating simulated conversations.

Index Terms: speaker diarization, end-to-end neural diarization, simulated conversations

1. Introduction

Since the introduction of end-to-end neural diarization (EEND) [1] and its extension to deal with arbitrary amounts of speakers [2, 3], it has been established as a state-of-the-art alternative to the standard cascaded diarization systems based on different submodules, i.e. voice activity detection (VAD), uniform segmentation, speaker embeddings extraction, clustering and overlapped speech detection (OSD) with handling. EEND formulates the speaker diarization problem as a per-speaker-per-time-frame binary classification problem where a permutation-free objective is used to minimize the speech activity error for all speakers. Therefore, EEND models generate one speech activity probability output for each speaker per time-step, which effectively solves VAD, speaker labeling and OSD at once.

Most works following the EEND principle have focused on improvements on the architecture or modeling. Some by using self-attention layers [4] or conformer layers [5] instead of the original BLSTM layers for feature encoding; others have focused on more complex diarization scenarios such as its online fashion [6, 7] or when more than one microphone is available [8] or by improving the model iteratively using pseudo-labels [9]. Some have used EEND together with more standard approaches by using EEND-inspired models to find overlaps among pairs of speakers in the output of a cascaded system [10] or leveraging EEND's VAD performance by using an external VAD system [11] or combining short duration diarization outputs to produce better whole-utterance diarization [12, 13, 14, 15]. However, none have yet tackled one of the main aspects of EEND: training data generation.

EEND models require large amounts of training data and datasets manually annotated for diarization do not amount for thousands of hours. When presenting the first version of EEND, Yusuke et al. proposed a strategy for constructing simulated mixtures using telephone conversations from different collections and this strategy has been used with both telephony data or read books to create simulated mixtures by mixing speakers from different recordings. However, little analysis has been presented about how the simulations were devised nor what impact have the used augmentations. Furthermore, the mixtures do not resemble real conversations, specially when more than two speakers are included. In this work we revise the approach and propose an alternative method that, based on statistics from real conversations, emulates some of the attributes of natural conversations. We also analyze the impact of using different types of augmentations on the diarization performance and show that our approach performs better than the original one while significantly reducing the dependence on the fine-tuning stage.

2. Training data generation strategy

Diarization training data consist of audio recordings and their corresponding speaker segment annotations. In this section, we describe the original approach [1] for creating the training data to which, respecting the original terminology, we will refer as simulated mixtures (SM) as well as our approach to which we will refer as simulated conversations (SC).

2.1. Simulated mixtures

In order to create the mixtures with their corresponding annotations, Yusuke et al. have used a VAD system run on each side of each conversation in a pool of thousands of telephone conversations. Assuming a single speaker per telephone channel, this means that speech segments and their speaker labels can be gathered from the pool to construct mixtures.

To create a mixture, as many speakers as wanted in the mixture (N_{spk}) are sampled from the total pool. The pool is represented by $\mathcal{U} = \{U_s\}_{s \in \mathcal{S}}$ where U_s is an utterance of speaker s formed by all the segments denoted by the VAD that belong to that speaker in the utterance[1]. For each selected speaker, one of their utterances is randomly sampled and N_u consecutive segments selected randomly from it. Pauses are introduced in between the selected segments of a speaker to simulate turns in a conversation, where the length of the pause is sampled from an exponential distribution with parameter β. The resulting audio

[1]Please note that Yusuke et al. have referred as "utterances" to the segments of speech in a real conversation.

10.21437/Interspeech.2022-10451

defines a channel for that speaker in the mixture. This process is repeated for all speakers in the mixture and finally the channels are summed to obtain a single utterance. This procedure is enriched by adding background noises and altering the room impulse responses (RIRs) of each speaker channel. For the sake of space we do not include the algorithm but refer the reader to Algorithm 1 of [1]. Note that since the channel of each speaker is generated independently, depending on the choice of β, the output can contain a high percentage of overlapped speech (usually higher than in real conversations).

2.2. Simulated conversations

One of the main concerns with SM is that each speaker in the mixture is treated independently. Although the lengths of the pauses are randomly drawn from an exponential distribution, a sensible choice as it represents the inter-arrival times between independent events, this does not resemble the dynamics of a real conversation where speakers do not take turns independently but collaboratively. For this reason, the proposed approach to create SC uses statistics about frequencies and lengths of pauses and overlaps from real conversations. Statistics are:

- Pauses between consecutive same speaker segments. The histogram of the pause lengths defines the distribution $D_{=\text{speaker}}$.
- Number of pauses ds between consecutive different speaker segments. The histogram of the pause lengths defines the distribution $D_{\neq\text{speaker}}$.
- Number of overlaps ov between consecutive different speaker segments. The histogram of the overlap lengths defines the distribution D_{overlap}.
- $p = \frac{ds}{ds+ov}$ is the probability of having a pause in between two segments of different speakers.

The proposed approach is described in Algorithm 1. In this case, utterances are sampled without replacement. When creating a large set of simulated conversations, this ensures that an utterance is not used more than once[2]. Furthermore, for a given utterance all segments are used as part of a single SC. In contrast, in the original approach, only a subset of the segments from each original conversation is used at a time.

The segments (defined by the VAD labels) of the selected utterances (one per speaker) are randomly interleaved, guaranteeing that the per-speaker order is kept while assigning random order to the speaker turns of the different speakers. Then, for each pair of consecutive segments after interleaving them, a gap is defined depending on the nature of the two segments as shown in Algorithm 1. Pauses or overlaps lengths are sampled using the estimated distributions. In the pseudo-code, there is a single channel initialized with 0's and segments are added to it in the obtained order by means of $out.addFromPosition(pos, in)$ which adds the signal in onto out starting from position pos.

In the analysis that we present in this work we consider the addition of background noises and reverberation (already present in [1]) as augmentations. For analyzing the effect of using or not using each of these augmentations, different training datasets are generated for each combination. The same EEND model is trained on each one independently and its performance evaluated on real data. We also explored scaling the energy levels between speakers in the conversations to resemble those in real conversations but this did not impact the performance.

For the sake of making the comparison between the two approaches as fair as possible, several aspects of the data preparation follow those of the original SM approach [1]:

[2]In practice, the code is prepared to run until exhausting all utterances and re-start again until a given amount of times is reached or until generating a specific amount of audio.

Algorithm 1 Conversation simulation

Input: $\mathcal{S}, \mathcal{N}, \mathcal{I}, \mathcal{R}$ ▷ Set of speakers, noises, RIRs, and SNRs
$\mathcal{U} = \{U_s\}_{s\in\mathcal{S}}$ ▷ Set of utterances
N_{spk} ▷ #speakers per conversation
$p, D_{=\text{speaker}}, D_{\neq\text{speaker}}, D_{\text{overlap}}$ ▷ Estimated distributions

$G \leftarrow \{\}$ ▷ Dictionary with list of segments per speaker
Sample a set of N_{spk} speakers $\mathcal{S}'$ from $\mathcal{S}$
for all $s \in \mathcal{S}'$ **do**
  Sample u from U_s without replacement
  Sample $\mathbf{i}$ from $\mathcal{I}$ ▷ RIR
  $u' \leftarrow u * \mathbf{i}$ ▷ Reverberate all segments in the utterance
  $G[s] \leftarrow u'$
$L \leftarrow$ Randomly interleave G into a single list
$\mathbf{y}.addFromPosition(0, L[1])$ ▷ Start signal with first segment
$pos \leftarrow \text{len}(L[1])$
for $t = 2$ to $\text{len}(L)$ **do**
  if $\text{Speaker}(L[t-1]) = \text{Speaker}(L[t])$ **then**
    Sample gap from $D_{=\text{speaker}}$
  else if Sample of $Bernoulli(p)$ is 1 **then**
    Sample gap from $D_{\neq\text{speaker}}$
  else
    Sample $-gap$ from D_{overlap}
  $\mathbf{y}.addFromPosition(pos + gap, L[t])$
  $pos \leftarrow pos + gap + \text{len}(L[t])$
Sample $\mathbf{n}$ from $\mathcal{N}$ ▷ Background noise
Sample r from $\mathcal{R}$ ▷ SNR
Determine a mixing scale p from r, $\mathbf{y}$, and $\mathbf{n}$
$\mathbf{n}' \leftarrow$ repeat $\mathbf{n}$ until reaching the length of $\mathbf{y}$
$\mathbf{y} \leftarrow \mathbf{y} + p \cdot \mathbf{n}'$

- The set of utterances used: comprised of Switchboard-2 (phases I, II, III) [16, 17, 18], Switchboard Cellular (parts 1 and 2) [19, 20], and NIST Speaker Recognition Evaluation datasets (from years 2004, 2005, 2006, 2008) [21, 22, 23, 24, 25, 26, 27, 28]. All the recordings are sampled at 8 kHz and, out of 6381 speakers, 90% are used for creating training data.
- The VAD used to obtain time annotations: based on time-delay neural networks and statistical pooling[3].
- The set of background noises and mechanism for augmenting data: 37 noises labeled as "background" in the MUSAN collection [29] are added to the signal scaled with a signal to noise ratio (SNR) selected randomly from {5, 10, 15, 20}.
- The set of room impulse responses (RIRs) used to reverberate utterances: a RIR is sampled from the collection used in [30] and with 0.5 probability used to reverberate the utterances of each speaker.

In order to shed some light on how the proposed SC resemble real conversations more than SM, some statistics about the recordings are presented in Table 1 where we see that SC is more similar to real sets in terms of percentages of silence, speech from a single speaker and overlap. We selected a subset of the SC to match the amount of hours using with SM in previous works. An analysis on the performance with different amounts of hours of data is presented in 4.4. The code for generating SC can be found in https://github.com/BUTSpeechFIT/EEND_dataprep

3. Experimental setup

3.1. Diarization models

The experiments were performed using the self-attention EEND with encoder-decoder attractors for showing superior performance in previous works. In all cases the architecture used was exactly the same as that described in [3][4]. For the sake

[3]http://kaldi-asr.org/models/m4

[4]We refer the reader to [3] for a scheme of the model.

Table 1: *Statistics for synthetic and real datasets. All listed sets have only 2 speakers per recording.*

Dataset	#files	Total audio (h)	Average dur. (s)	Average % sil.	1spk	over.
SM (β=2)	100k	2480	89.30	18.61	49.62	31.77
SC	25k	2480	356.15	12.80	78.83	8.37
CH Part1	155	3.19	74.02	9.05	77.90	13.05
CH Part2	148	2.97	72.14	9.84	78.34	11.82
DH3 dev	61	10.17	599.95	10.56	77.27	12.17
DH3 eval	61	10.17	599.95	10.89	78.62	10.49

of making the code more efficient, we used our PyTorch implementation[5]. 15 consecutive 23-dimensional log-scaled Mel-filterbanks (computed over 25 ms every 10 ms) are stacked to produce 345-dimensional features every 100ms. These are transformed by 4 self-attention encoder blocks (with 4 attention heads each) into a sequence of 256 dimensional embeddings. These are then shuffled in time and fed into the LSTM-based encoder-decoder module which decodes as many attractors as speakers are predicted. A binary linear classifier is used to obtain speech activity probabilities for each speaker (represented by an attractor) at each time (represented by an embedding).

During inference time, classifiers' outputs are thresholded at 0.5 to determine speech activities. To ease the comparison, this parameter was not tuned in any experiment but tuning it could lead to better results. Each training was run for 100 epochs on a single GPU. The batch size was set to 64 or 32 with 100000 or 200000 minibatch updates of warm up respectively. Following [3], the Adam optimizer [31] was used and scheduled with noam [32]. For fine-tuning on a development set, 100 epochs were run and the Adam optimizer was used with learning rate 10^{-5}. For the inference as well as for obtaining the model from which to fine-tune, the models from the last 10 epochs were averaged. During training and fine-tuning phases, batches are formed by sequences of 500 Mel-filterbank outputs, corresponding to 50 s. During inference, the full recordings are fed to the network one at a time.

Diarization performance is evaluated in terms of diarization error rate (DER) as defined by NIST [33].

3.2. Data

Two real telephone conversations datasets were used to report results. First, the 2000 NIST Speaker Recognition Evaluation [34] dataset, usually referred as "Callhome" [35]. We report results on the subset of 2-speaker conversations using the standard Callhome partition[6]. We will refer to the parts as CH1-2spk and CH2-2spk. Results on Callhome consider all speech (including overlap segments) for evaluation with a forgiveness collar of 0.25 s. Second, the CTS domain from the recent Third DIHARD Challenge [36], which consists of previously unpublished telephone conversations from the Fisher collection. Both development and evaluation sets consist of 61 10-minute recordings each (full set). We will refer to the sets as DH-dev and DH-eval. Originally 8 kHz signals, they were upsampled to 16 kHz for the challenge and downsampled to 8 kHz to be used in this work. As usual on DIHARD, all speech is evaluated with a collar of 0 s.

[5]https://github.com/BUTSpeechFIT/EEND

[6]Each of these sets are listed in https://github.com/BUTSpeechFIT/CALLHOME_sublists

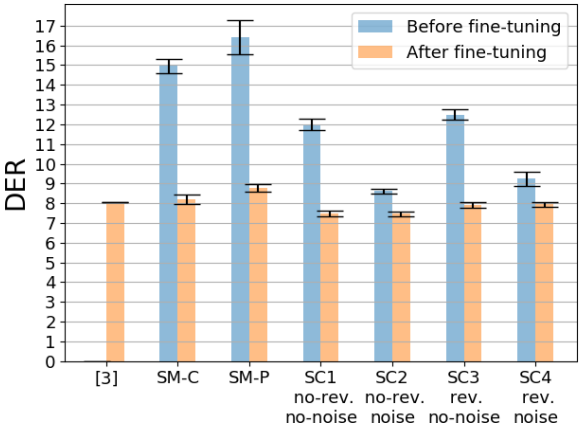

Figure 1: *DER (%) comparison for SM (C stands for Chainer and P for PyTorch) and SC options on CH2-2spk and fine-tuning to CH1-2spk. [3] is a single run. All other experiments were repeated 5 times and we show means and error bars.*

4. Results

4.1. SC augmentations analysis

We evaluated all combinations of the augmentations, namely reverberate and add noises. For this analysis, statistics to generate SC were estimated on DH-dev. Figure 1 presents results on recordings with 2 speakers from Callhome Part 2 comparing the best published result with the original Chainer implementation [3], our runs with the Chainer implementation, our PyTorch implementation with SM and the PyTorch implementation with the different SC options. Our runs with Chainer correspond with the result in [3]. The results with PyTorch are slightly worse, note that there might be small implementation differences between Chainer and PyTorch and that the same hyperparameters tuned for training with Chainer were used with the PyTorch implementation to ease the comparison and that adjusting them could lead to further improvement.

Using any of the SC options for generating training data outperforms using SM. Even more, the performance of the models trained using the best SC option (SC 2) is close to the results obtained with the models trained on SM after fine-tuning on the real sets. The models trained on the best SC option can, in turn, be further improved by means of fine-tuning, significantly outperforming those trained on SM, showing that a model trained on better quality SC data not only performs better but can also be still leveraged in combination with fine-tuning to real data.

When comparing the four options, the main gains are observed when adding background noises; as expected, this simulates noisier conditions and adds variability to the set. Reverberating the signals with the default parameters actually harms the performance. We hypothesize that the effect might be different in other scenarios such as meetings or interviews where speakers do not have specific microphones close to the mouth such as in telephone conversations. Furthermore, the choice of RIRs could be narrowed down to take into account only rooms that resemble those in real applications. These aspects need to be further studied, especially for the more diverse scenarios seen in wide-band data. Results in following sections are obtained using the best result with SC 2 and SM (P).

4.2. DER breakdown analysis

Table 2 presents a detailed comparison of the errors from SM (P) and SC 2. In terms of VAD, both systems perform very similarly before and after fine-tuning. However, larger differences

can be seen in terms of OSD. When using SC for training, the model makes considerably less false alarms for OSD, specially before fine-tuning. This can be explained by the larger percentage of overlap seen in SM (Table 1). Mechanisms for favoring slightly higher percentages of overlap when creating SC are left for future studies.

Table 2: *Error analysis before and after fine-tuning on recordings with 2 speakers of CH2-2spk.*

System	DER	DER breakdown Miss	FA	Conf.	VAD Miss	FA	OSD Miss	FA
SM (P)	15.09	2.83	8.24	4.01	0.48	7.70	4.84	10.71
+ FT	8.44	5.23	2.32	0.90	3.39	4.06	6.20	4.22
SC 2	8.64	3.11	4.84	0.69	0.49	7.53	4.68	9.03
+ FT	7.28	4.72	1.98	0.58	3.23	4.03	6.02	3.82

4.3. Statistics source and fine-tuning analysis

To analyze the effect of the set used to estimate the statistics for creating SC data, we made a comparison using either DH-dev or CH1. As seen in Table 3, the effect of the set used for the estimation of the statistics is not large. Both datasets present several recordings that amount to few hours of speech which is enough to have a reasonable amount of data to estimate the statistics. At the same time, the fine-tuning step improves the performance on both datasets suggesting that real conversations still differ from SC, leaving room for improving SC quality.

Table 3: *DER (%) for models trained with SC 2 using statistics estimated on DH-dev or CH1. Fine-tuning for each test set is done in the corresponding dev set. Numbers in gray denote results where the test data were used for training. In underlined results, test data were used to compute the statistics.*

System	Callhome 2 speakers Part 1	Part 2	DIHARD 3 CTS full dev	eval
DH stats	8.16	8.64	23.47	22.06
+ FT	6.20	7.28	16.99	17.00
CH stats	8.26	8.73	22.29	21.53
+ FT	6.13	7.28	17.38	17.14

4.4. Amount of data analysis

One aspect to analyze is the effect that the amount of training data has on EEND performance when using SC. Figure 2 shows that the performance when training with *as little as* 310 hours degrades considerably before fine-tuning. However, using 1240 hours (half of the amount used in all other experiments) already allows for similar performance as using more data. Even more, using more data allows the model after fine-tuning to reach as little as 7.03% DER on average and 6.8% DER in the best run.

4.5. Comparison with previous results

In this section we compare our results to the best results published with the same architecture. Table 4 shows that when training with SC it is possible to attain similar performance as when training with SM after fine-tuning on Callhome. This gain is smaller in the case of DIHARD CTS where fine-tuning has a larger effect. One of the aspects to consider with the architecture used is that the input is downsampled so that one output every 100 ms is produced. When evaluating with a 0 s forgiveness collar, such as in DIHARD, this severely impacts the results. Table 4 presents results when evaluating at the standard

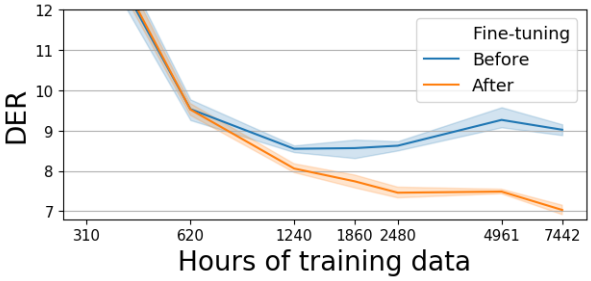

Figure 2: *DER (%) on CH2-2spk when training with different amounts of hours of SC 2. Each experiment is repeated 5 times to obtain the mean and confidence intervals.*

Table 4: *DER (%) for different systems.*

System	CH 2 speakers Part 1	Part 2	DIHARD 3 CTS full dev		eval	
S.H. et al. [3]	–	8.07	–		–	
			Downsample rate (ms) 50	100	50	100
SM (P)	13.62	15.09	25.36	25.46	22.16	23.58
+ FT	7.61	8.44	12.97	17.98	11.99	17.44
SC 2	8.16	8.64	21.16	23.47	19.81	22.06
+ FT	6.2	7.28	11.68	16.99	11.20	17.00

downsampling of one output every 100 ms and 50 ms[7], showing that a considerable error reduction is possible after fine-tuning.

5. Conclusions

EEND-based diarization systems are being thoroughly explored with many different variants but they all require a vast amount of data with diarization labels for training. We presented an alternative strategy for generating training data which uses statistics of real conversations to resemble real data. Our approach significantly outperforms the original one, proving specially useful in cases where there would not be a development set for fine-tuning. However, with fine-tuning, the performance can still be leveraged showing that there is still room for improving the generation of synthetic conversations. In our future work we plan to explore the use of RIRs that resemble real scenarios and relevance of VAD and overlap related statistics. Also, we plan to further exploit our approach for creating conversations with more than 2 speakers and use it with wide-band data where there are far less hours of single-speaker recordings in conversational setups available to create synthetic conversations.

6. Acknowledgements

The work was supported by Czech Ministry of Interior project No. VJ01010108 “ROZKAZ”, Czech National Science Foundation (GACR) project NEUREM3 No. 19-26934X and the author from the UAM was supported by project RTI2018-098091-B-I00, granted by MCIU/AEI/FEDER, UE. Computing on IT4I supercomputer was supported by the Czech Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project “e-Infrastructure CZ – LM2018140”.

[7]This is done only at inference time (not during training), the system is exactly the same and cannot address the issue completely. Having finer granularity not only increases the inference time considerably but the memory requirements as well.

7. References

[1] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *arXiv preprint arXiv:1909.05952*, 2019.

[2] 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.

[3] 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.

[4] 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 Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[5] Y. Chieh Liu, E. Han, C. Lee, and A. Stolcke, "End-to-end neural diarization: From transformer to conformer," *arXiv e-prints*, pp. arXiv–2106, 2021.

[6] E. Han, C. Lee, and A. Stolcke, "Bw-eda-eend: Streaming end-to-end neural speaker diarization for a variable number of speakers," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7193–7197.

[7] 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 Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 841–848.

[8] S. Horiguchi, Y. Takashima, P. Garcia, S. Watanabe, and Y. Kawaguchi, "Multi-channel end-to-end neural diarization with distributed microphones," *arXiv preprint arXiv:2110.04694*, 2021.

[9] Y. Takashima, Y. Fujita, S. Watanabe, S. Horiguchi, P. García, and K. Nagamatsu, "End-to-end speaker diarization conditioned on speech activity and overlap detection," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 849–856.

[10] S. Horiguchi, P. Garcia, Y. Fujita, S. Watanabe, and K. Nagamatsu, "End-to-end speaker diarization as post-processing," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7188–7192.

[11] S. Horiguchi, N. Yalta, P. Garcia, Y. Takashima, Y. Xue, D. Raj, Z. Huang, Y. Fujita, S. Watanabe, and S. Khudanpur, "The hitachi-jhu dihard iii system: Competitive end-to-end neural diarization and x-vector clustering systems combined by dover-lap," *arXiv preprint arXiv:2102.01363*, 2021.

[12] 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.

[13] ——, "Advances in integration of end-to-end neural and clustering-based diarization for real conversational speech," *arXiv preprint arXiv:2105.09040*, 2021.

[14] S. Horiguchi, S. Watanabe, P. Garcia, Y. Xue, Y. Takashima, and Y. Kawaguchi, "Towards neural diarization for unlimited numbers of speakers using global and local attractors," *arXiv preprint arXiv:2107.01545*, 2021.

[15] K. Kinoshita, M. Delcroix, and T. Iwata, "Tight integration of neural-and clustering-based diarization through deep unfolding of infinite gaussian mixture model," *arXiv preprint arXiv:2202.06524*, 2022.

[16] D. Graff, A. Canavan, and G. Zipperlen, "Switchboard-2 phase I, LDC98S75," 1998.

[17] D. Graff, K. Walker, and A. Canavan, "Switchboard-2 phase II, LDC99S79," *Web Download. Philadelphia: Linguistic Data Consortium*, 1999.

[18] D. Graff, D. Miller, and K. Walker, "Switchboard-2 phase III, LDC2002S06," *Web Download. Philadelphia: Linguistic Data Consortium*, 2002.

[19] D. Graff, K. Walker, and D. Miller, "Switchboard Cellular Part 1 audio LDC2001S13," *Web Download. Philadelphia: Linguistic Data Consortium*, 2001.

[20] ——, "Switchboard Cellular Part 2 audio LDC2004S07," *Web Download. Philadelphia: Linguistic Data Consortium*, 2004.

[21] N. M. I. Group, "2006 nist speaker recognition evaluation evaluation test set part 1 ldc2011s10," 2011.

[22] ——, "2006 nist speaker recognition evaluation evaluation test set part 2 ldc2012s01," 2012.

[23] ——, "2008 nist speaker recognition evaluation training set part 1 ldc2011s05," 2011.

[24] ——, "2008 nist speaker recognition evaluation test set ldc2011s08," 2011.

[25] ——, "2005 nist speaker recognition evaluation test data ldc2011s04," 2011.

[26] ——, "2006 nist speaker recognition evaluation training set ldc2011s09," 2011.

[27] ——, "2004 nist speaker recognition evaluation ldc2006s44," 2006.

[28] ——, "2005 nist speaker recognition evaluation training data ldc2011s01," 2006.

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

[30] 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.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[33] "NIST Rich Transcription Evaluations," https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation, version: md-eval-v22.pl.

[34] M. Przybocki and A. Martin, "Nist speaker recognition evaluation ldc2001s97," *Philadelphia, New Jersey: Linguistic Data Consortium*, 2001.

[35] "NIST SRE 2000 Evaluation Plan," https://www.nist.gov/sites/default/files/documents/2017/09/26/spk-2000-plan-v1.0.htm_.pdf.

[36] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third dihard diarization challenge," *arXiv preprint arXiv:2012.01477*, 2020.