INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Robust End-to-end Speaker Diarization with Conformer and Additive Margin Penalty

Tsun-Yat Leung, Lahiru Samarakoon

Fano Labs, Hong Kong, China
ty.leung@fano.ai, lahiru@fano.ai

Abstract

Traditionally, a speaker diarization system has multiple components to extract and cluster speaker embeddings. However, end-to-end diarization is more desirable as it facilitates optimizing one model in contrast to multiple components in a traditional set up. Moreover, end-to-end diarization systems are capable of handling overlapped speech. Recently proposed self-attentive end-to-end diarization model with encoder-decoder based attractors (EEND-EDA) is capable of processing speech from an unknown number of speakers, and has reported comparable performances to traditional systems. In this work, we aim to improve the EEND-EDA model. First, we increase the robustness of the model by incorporating an additive margin penalty for minimizing the intra-class variance. Second, we propose to replace the Transformer encoders with Conformer encoders to capture local information. Third, we propose to use convolutional subsampling and upsampling instead of manual subsampling only. Our proposed improvements report 21.6% relative reduction in DER on the evaluation full set of the track 2 of the DIHARD III challenge.

Index Terms: End-to-end speaker diarization, Attractors, Additive margin penalty, Conformer, Convolutional subsampling and upsampling

1. Introduction

Given a multi-speaker recording, speaker diarization focuses on solving the problem "who spoke when". In the traditional method, where speaker embeddings are clustered to detect different speakers, we need to perform a sequence of steps that use multiple machine learning models [1, 2, 3]. First, voice activity detection (VAD) is performed to identify speech segments and then speaker embeddings are extracted for these segments [4, 5]. Finally, these embeddings are clustered based on a selected similarity score and a clustering algorithm. One of the major disadvantages of traditional systems is that having to optimize and maintain multiple components. In addition, traditional systems fall short in handling speaker overlaps as the embedding model expects speaker-homogeneous speech segments [4, 5]. Moreover, the clustering method (e.g. agglomerative hierarchical clustering) cannot assign multiple speaker memberships for an overlapped segment.

End-to-end systems are focusing on handling these shortcomings of traditional diarization systems. In [6], End-to-end neural diarization system (EEND) was proposed to handle a fixed number of speakers. Then, self-attentive EEND (SA-EEND) [7] was proposed where the bidirectional LSTMs [8] in the EEND encoder were replaced by Transformer encoders [9]. This SA-EEND model has been extended to handle unknown number of speakers by the end-to-end diarization model with encoder-decoder based attractor (EEND-EDA) [10] and the speaker-wise conditional EEND (SC-EEND) [11]. The EEND-EDA system achieved the state-of-the-art performance on CALLHOME [10]. Moreover, EEND-EDA significantly outperforms the traditional baseline released with the DIHARD II challenge. In this work, we are focused on further improving the EEND-EDA model.

In the recently concluded DIHARD III challenge, end-to-end diarization systems performed well. In [12], authors extended the EEND-EDA model with an external VAD and an iterative inference mechanism to achieve a strong performance. Furthermore, they combined their model with other diarization systems to achieve the state-of-the-art performance. Our submission for DIHARD III was also based on the EEND-EDA model [13]. In that work, we focused on using one model to guarantee the end-to-end nature of the method. We achieved our best performance for our DIHARD III submission by extending the EEND-EDA model with a combination of techniques. Namely, we replaced the EEND-EDA Transformer encoders layers with Conformers [14], introduced an attention mechanism for attracter's encoder-decoder model, added a convolutional subsampling and upsampling, and included an additive margin penalty to reduce the intra-class variance during the fine-tuning. Even though we achieved a competitive performance, given the timeline of the challenge, we were not able to provide valuable insights.

In this paper, we carefully analysed multiple techniques used in our DIHARD III submission. We found that incorporating an attention mechanism is not helpful as used in [13], given we pre-train the EEND-EDA model for a longer time. Therefore, attractor attention based experiments are excluded in this work. We focus our experimental analysis mainly on three techniques. First, we explore various ways to incorporate an additive margin penalty to increase the robustness by reducing the intra-class variance. Second, replacing the Transformer encoder layers with Conformer layers is explored. Finally, the model is allowed to learn how to subsample and upsample the data via convolutional modules. To the best of our knowledge, this is the first work that explores these modifications for the enhancement of the EDA-EEND model.

2. EEND-EDA Review

The encoder of the EEND-EDA is composed of Transformer encoders. It first takes a T-length sequence of log-scaled Mel-filterbank features as input, and converts them to T number of embeddings $\mathbf{e_t} \in \mathbb{R}^D$ with dimension D at each time step t.

To estimate the number of speakers in a recording, an LSTM-based encoder-decoder takes the embeddings $E := [\mathbf{e}_1, ..., \mathbf{e}_T] \in \mathbb{R}^{D \times T}$ as input and generates $S + 1$ number of attractors $a_s \in \mathbb{R}^D$ recurrently, where S is the number of speakers. The probability of whether the attractor exists for an actual speaker or represents the termination of the calculation is computed by a linear layer with a sigmoid func-

http://dx.doi.org/10.21437/Interspeech.2021-1377

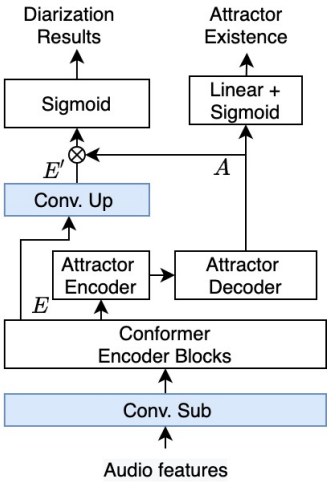

Figure 1: *Proposed system with conformer blocks, convolutional subsampling and upsampling.*

tion. The last attractor is used for stopping the calculation, so the encoder-decoder produces S number of speaker attractors $A := [\mathbf{a}_1, ..., \mathbf{a}_S] \in \mathbb{R}^{D \times S}$.

After obtaining E and A, the EEND-EDA calculates the posterior probability of the diarization result for each speaker s at time t using the the following equation:

$$\hat{y}_{t,s} = \sigma(\mathbf{a_s}^T \mathbf{e_t}) \in (0, 1) \tag{1}$$

where σ is the sigmoid function. During training, the diarization results in the EEND-EDA is optimized using the Permutation Invariant Training (PIT) scheme [15] and the loss is calculated between the posterior probabilities $\hat{\mathbf{y}}_t := [\hat{y}_{t,1}, ..., \hat{y}_{t,S}]^T \in (0, 1)^S$ and the ground-truth labels $\mathbf{y}_t := [y_{t,1}, ..., y_{t,S}]^T \in \{0, 1\}^S$ as follows:

$$L_d = \frac{1}{TS} \min_{p \in \text{perm}(1,...,S)} \sum_{t=1}^{T} H(\mathbf{y}_t^p, \hat{\mathbf{y}}_t), \tag{2}$$

where $\text{perm}(1, ..., S)$ is the set of all the possible permutation of speakers, $\mathbf{y}_t^p$ is the permuted labels at t, and $H(\mathbf{y}_t^p, \hat{\mathbf{y}}_t)$ is the binary cross entropy.

3. Proposed Method

Our proposed modifications are described in the following subsections.

3.1. Additive Margin Penalty

Our additive margin penalty technique is highly inspired by the angular softmax with margin penalty [16, 17, 18]. It intends to reduce the intra-class variance by introducing a margin penalty to the target class logit. Different papers have shown its effectiveness in face recognition and speaker verification [17, 19]. We focus on the additive margin penalty version in [17] because of its performance and ease of implementation. Although the additive margin originally comes with softmax [17], it can be applied on the sigmoid outputs in EEND-EDA with a slight modification. In general, we could add the additive margin penalty to a sigmoid-based posterior probability calculation as follows:

$$\hat{\phi} = \sigma(\gamma(\psi - m\phi + m(1 - \phi))), \tag{3}$$

where $\phi \in \{0, 1\}$ is the label, $\hat{\phi}$ is posterior probability with additive margin, ψ is the input logit, γ is the scale factor, m is the additive margin value. γ and m are the hyperparameters. In [17], ψ is the cosine value between the normalized embedding and the normalized target class weight vector.

3.1.1. Additive Margin on Diarization Output

There are two type of outputs in EEND-EDA, as shown on the Figure 1. One is the diarization results output and another is the speaker attractor existence output. To incorporate the additive angular margin into the diarization output calculation, the encoder embeddings and the attractors are first normalized as in [17]. The normalization enforces that the model learns to decrease the angle between embedding and the target attractor. It also encourages the model to increase the angle between attractors for reducing the false alarm speeches. In the meantime, we need to know whom the attractors represent such that we can add the margin penalty accordingly. The optimal permutation p' of speaker labels is determined in the same way as [10] using the PIT with the normal diarization loss, i.e. the equations 1 and 2. That is calculated as follows:

$$p' = \underset{p \in \text{perm}(1,...,S)}{\arg\min} \sum_{t=1}^{T} H(\mathbf{y}_t^p, \hat{\mathbf{y}}_t) \tag{4}$$

After obtaining the optimal permutation p', the posterior probability with additive margin $\hat{y}'_{t,s}$ of the system speaker s at time t is defined as:

$$\hat{y}'_{t,s} = \sigma(\gamma(\mathbf{a_s}^T \mathbf{e_t} - y_{t,s}^{p'} m + (1 - y_{t,s}^{p'})m)), \tag{5}$$

where $y_{t,s}^{p'}$ is the label at t of the reference speaker s permuted with the optimal permutation p'. Finally, instead of optimizing with the normal diarization loss in equation 2, we optimize the model by the following equation:

$$L_d = \frac{1}{TS} \sum_{t=1}^{T} H(\mathbf{y}_t^{p'}, \hat{\mathbf{y}}'_t), \tag{6}$$

where $\hat{\mathbf{y}}'_t$ is the posterior probabilities with additive margin at t.

3.1.2. Embedding Normalization

The diarization output could have speaker overlaps. Unlike the angular softmax in the face recognition and speaker verification [17, 19], the diarization outputs at each time step are formulated as a multi-label classification instead of a multi-class classification. For example, if the speakers s_1 and s_2 both speak at time t, then $\hat{y}_{t,1}$ and $\hat{y}_{t,2}$ should be close to 1. However, under the normalization constraint on the embeddings and the attractors, it is only possible when the attractors $\mathbf{a}_1$ and $\mathbf{a}_2$ are close to each other. This may hinder our model from maximizing the angle or distance between attractors, which is key to the good generalization in the recognition tasks [17, 18, 5]. Therefore, we explore the setting where the embeddings are not normalized despite that it no longer directly forces the model to minimize the angle between embedding and the target attractor.

3.1.3. Additive Margin on Attractor Existence Output

We also explore the system where the additive margin is added on the attractor existence calculation according to the equation 3. ψ in this case is the dot product between the normalized attractors and a normalized weight vector. No bias is used. For

the system where the additive margin is not used on the attractor existence output, the posterior probabilities of the attractor existence is still calculated by the un-normalized attractors followed by a linear layer with a sigmoid function.

3.2. Conformer

Transformers are good at capturing global features of a sequence while lack in capturing fine-grained local features. To address this issue, Conformer was proposed, which combines transformers and convolutions networks in a parameter efficient way [14]. It achieved state-of-the-art performance of Librispeech speech recognition task. By using Conformer blocks instead of Transformer blocks in the EEND-EDA, it enables that the model explores the local information effectively. In our previous challenge system [13], we have first shown the effectiveness of Conformer. [12] also used stacked Conformer encoders instead of Transformers for SC-EEND but no comparison is provided.

3.3. Convolutional Subsampling and Upsampling

The input features in the EDA-EEND [10] are manually subsampled by a factor of ten to lower the computation cost. To subsample the data while maximizing the performance, we use convolutional subsampling instead of manual subsampling. This allows the model to learn in keeping most salient information. Besides, a convolutional upsampling module is placed after the conformer encoders to give the model the flexibility in producing results with enough precision. So, the input features are the 23-dimensional log-Mel filterbanks with a 25-ms frame length and 10-ms frame shift. Finally, the diarization posterior probabilities are calculated by multiplying the upsampled embeddings $\mathbf{E}'$ with the attractor matrix $\mathbf{A}$ followed by a sigmoid non-linear transformation, as shown on the Figure 1.

4. Experiments

4.1. Data

Table 1: *Datasets*

Dataset	#Mixtures
Pretraining Training Set	
Librispeech Simulated (β=2, 2, 5, 9)	400,000
Fine-Tuning Training Set	
VoxConverse Development	216
DIHARD III Development - Training	203
Fine-Tuning Validation Set	
DIHARD III Development - Validation	51
Evaluation Set	
DIHARD III Evaluation	259

The datasets used are summarized in Table 1. First, the Librispeech [20][1] recordings are segmented using WebRTC VAD [2]. Then, the simulated training set is created from the Librispeech 960 hour training set using the scripts provided in [10][3]. We created 1, 2, 3 and 4-speaker simulated mixtures. For the fine-tuning datasets, we split the whole DIHARD III development set into training and validation sets with a ratio of 80%:20% per domain. We also include VoxConverse [21] into the fine-tuning dataset [4]. Different from our previous challenge system [13], we removed duplicated data and we followed the β settings in [10]. The β determines the average duration of silence between utterances.

In DIHARD III, the development set and the evaluation set include recordings from 11 domains with different recording equipment, recording environment, number of speakers, etc. Each dataset is partitioned into full set and core set. The full set is the set that includes all available audio for each domain. The core set is a subset of the full set, and it is a "balanced" dataset where the total duration of each domain is approximately equal. The detail of DIHARD III data is in [22, 23].

4.2. Experimental Setup

We basically followed the training procedure in [10]. The model has 4 layers of Transformer or Conformer encoders with 256 hidden units. Our model were pretrained on Librispeech 2-speaker mixtures for 100 epochs. Then, they were fine-tuned on all Librispeech mixtures for 25 epochs. Finally, they were adapted on the Fine-Tuning Training Set with a maximum of 500 epochs, and were validated on the full set of Fine-Tuning Validation Set. For each fine-tuning step, the model was initialized with the average weight of the last 10 epochs in the previous step. The batch size is 32. The default values of m and γ in the equation 3 are 0.35 and 10, respectively.

During the final fine-tuning, for each recording we sampled ten different audio segments of 50s randomly to formulate one epoch. Chunk Shuffling mentioned in [13] was used. Adam optimizer with a fixed learning rate of 5×10^{-5} was used. The weighting parameter mentioned in the total loss of the EEND-EDA [10] was set to 0.1. In addition, the model was trained only to output four most dominant speakers because of the difficulty of PIT. We used a 6-fold augmentation for DIHARD development set that combines the original DIHARD development set with five augmented copies. The detail of the augmentation is mentioned in [13].

In our Conformer setting, relative position encoding and Macaron style feed-forward modules are used. The convolution kernel size is 15 and the number of attention heads is 4. The number of hidden units in feed-forward module is 1024.

The convolutional subsampling module consists of two convolutional layers, where the kernel sizes are $\{3, 5\}$ and the strides are $\{2, 5\}$. The upsampling module consists of two transposed convolution layers with batch normalization and ReLU activation. The kernel sizes are $\{3, 5\}$, the strides are $\{2, 5\}$, and the output paddings are $\{1, 0\}$. The number of output channels equals to the number of hidden units in the conformer encoder.

The evaluation metric we used is the diarization error rate (DER). Following the evaluation practice in DIHARD III, no collar is used. In the following experiments, we are focusing on the DIHARD III track 2 evaluation, where no oracle speech segmentation is provided.

4.3. Experiments on Conformer and Convolutional Subsampling & Upsampling

Table 2 reports the diarization results of using conformer encoders and convolutional subsampling and upsampling. Conformer significantly improves the DER on all sets of data com-

[1] https://www.openslr.org/12

[2] https://github.com/staplesinLA/denoising_DIHARD18

[3] https://github.com/hitachi-speech/EEND

[4] https://www.robots.ox.ac.uk/~vgg/data/voxconverse/

Table 2: *Result of Conformer and Convolutional Subsampling & Upsampling. The column "Conformer" indicates if replacing the Transformer with Conformer in the EEND-EDA. "Conv. Sub. & Up." indicates if the convolutional subsampling and upsampling modules are used. "Val" refers to our fine-tuning validation set.*

Part	Conformer	Conv. Sub. & Up.	DER(%)	
			Val	Eval
core	No	No	27.58	29.72
core	Yes	No	26.34	28.48
core	Yes	Yes	**24.45**	**26.25**
full	No	No	25.80	25.50
full	Yes	No	23.76	24.03
full	Yes	Yes	**21.84**	**21.74**

pared to EEND-EDA. It showed that the relative local information is important for the diarization. When combining the convolutional subsampling and upsampling module, in total it gives a relative 14.7% DER reduction on the evaluation full set. Thus, we continue our experiments with the current best setting.

4.4. Experiments on Additive Margin

Table 3: *Result of additive margin penalty. "Diar. Output" indicates if using additive margin penalty on the diarization output. "Norm. Emb." indicates if the embedding is normalized. "Attractor Output" indicates if using additive margin penalty on the attractor existence.*

Part	Diar. Output	Norm. Emb.	Attractor Output	DER(%)	
				Val	Eval
core	No	N.A.	No	24.45	26.25
core	Yes	Yes	No	22.97	25.03
core	Yes	No	No	**21.02**	**24.30**
core	Yes	No	Yes	23.60	26.41
full	No	N.A.	No	21.84	21.74
full	Yes	Yes	No	20.21	20.85
full	Yes	No	No	**19.48**	**19.99**
full	Yes	No	Yes	21.08	21.35

Table 3 shows different variants of applying additive margin penalty. The additive margin penalty is applied during all training steps instead of only the last fine-tuning step in our previous system [13]. First, the additive margin penalty with embedding normalization is added on the diarization outputs. On the evaluation full set, the DER decreases from 21.74% to 20.85%. By not normalizing the embedding, it further improves the system and gives the best result on all datasets. This improvement supports our hypothesis that the normalization on the embeddings and the attractors together is sub-optimal. Finally, the result shows that adding the margin penalty into the attractor existence calculation decreases the performance. It indicates that the diarization output is more important to the robustness and performance.

4.5. Experiments on the Hyper-parameter m

The experiments in Table 4 are based on the best setting in Table 3. We can see the performance improves significantly from $m = 0.1$ to $m = 0.35$ on the evaluation full and core set. In general, the margin penalty with margin values ranging from 0.2 to 0.4 are able to significantly improve the performance.

Table 4: *Result of the effect of the hyper-parameter m*

m	DER (%)			
	Val Core	Val Full	Eval Core	Eval Full
0.1	21.39	19.76	26.09	21.35
0.2	21.76	**18.94**	25.57	20.90
0.3	21.36	19.20	24.38	20.10
0.35	**21.02**	19.48	**24.30**	**19.99**
0.4	22.80	21.04	24.99	20.57

5. Conclusions

In this work, we improved the EEND-EDA model with three techniques. First, we incorporated the additive margin penalty into the diarization results calculation to minimize the intra-class variance. During the experiments, we found that normalizing both the embeddings and the attractors is sub-optimal, so the embeddings are not normalized in our best setting. Second, we replaced the Transformer encoders with Conformer encoders to capture local information. Third, we used the convolution networks for subsampling and upsampling instead of manual subsampling. Our proposed modifications achieved our best performance on the DIHARD III. The improvement reported 21.6% relative reduction in DER on the evaluation full set of the track 2 of the DIHARD III challenge.

6. References

[1] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 413–417.

[2] 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.

[3] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," in *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Barcelona, Spain, May 2020.

[4] 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.

[5] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[6] 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.

[7] 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.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] 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, pp. 5998–6008, 2017.

[10] 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," *Proc. Interspeech 2020*, 2020.

[11] 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.

[12] 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.

[13] T.-Y. Leung and L. Samarakoon. (2021) End-to-end speaker diarization system for the third dihard challenge system description. [Online]. Available: https://dihardchallenge.github.io/dihard3/system_descriptions/dihard3_system_description_team106.pdf

[14] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, 2020.

[15] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 241–245.

[16] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[17] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[18] 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.

[19] D. Garcia-Romero, D. Snyder, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "X-Vector DNN Refinement with Full-Length Recordings for Speaker Recognition," in *Proc. Interspeech 2019*, 2019, pp. 1493–1496. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2205

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

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

[22] N. Ryant, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "Third dihard challenge evaluation plan," *arXiv preprint arXiv:2006.05815*, 2020.

[23] 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.