Speaker count: a new building block for speaker diarization

Thanh Thi-Hien Duong*, Phi-Le Nguyen†, Hong-Son Nguyen‡, Duc-Chien Nguyen‡, Huy Phan§, Ngoc Q. K. Duong¶
*Hanoi University of Mining and Geology, Vietnam
†Hanoi University of Science and Technology, VietNam
‡Aimenext Join Stock Company, Vietnam
§Queen Mary University of London, UK
¶InterDigital, France
E-mail: duongthihienthanh@humg.edu.vn, lenp@soict.hust.edu.vn, sonnh@aimenext.com, chiennd@aimenext.com
h.phan@qmul.ac.uk, quang-khanh-ngoc.duong@interdigital.com

***Abstract*—In daily communication, several people sometimes talk simultaneously, resulting in overlapped speech segments. Such segments challenge machine listening tasks like speaker diarization or speech recognition. This paper presents a speaker diarization framework where speaker count, a building block to predict the number of active speakers in each analyzing audio window, is integrated. Such speaker count block can be developed independently with existing speaker diarization systems; its output is then used in the re-segmentation step of existing systems to better label active speakers in each considered window. We further investigate the effect of analyzing window size in diarization performance in an oracle setting. Our preliminary theoretical analysis shows that the overlap speech detection, a special case of speaker count, is helpful to reduce diarization error rate when the window size is small enough. Finally, experiment results obtained from two state-of-the-art diarization systems on a benchmark dataset confirm the potential benefit of the proposed approach.**

I. Introduction

In natural conversations, there are often instances where multiple people speak at the same time. Such overlapped speech instances often be problematic for automatic speech processing tasks such as speech recognition, blind source separation, and speaker diarization. As an example in speaker diarization, which aims to answer the question "Who spoke when?" [5], conventional clustering algorithms tend to output only the most likely spoken person and miss the others in overlapped segments. Thus, even the best performing systems struggle to identify who are speaking in real-world situation with the presence of noise and highly overlapped speeches [14], [15], [22], [23].

This paper addresses speaker diarization task in multi-speaker audio recordings. Conventional approach are often based on segmenting the input audio stream into uniform speech segments with the help of the voice activity detection (VAD), followed by extracting fixed-length speaker embeddings from those segments, and finally performing speaker clustering over these embeddings [2]. Some recent systems use more building blocks such as overlap detection and speaker change detection to support clustering algorithm [28], [8]. It is worth noting that there have been several studies on overlapped speech detection and its application on speaker diarization. For instance, Boakye *et al.* use an Hidden Markov Model (HMM) based segmenter to detect overlapped segments and demonstrate a relative improvement of about 7.4% diarization error rate (DER) over a baseline [5]. Huijbregts *et al.* [19] propose a Gaussian Mixture Model (GMM) based speaker model for the overlap detection and investigate a "two-pass" system to first detect overlap, then use it to refine speaker models and make assignments. In [18], the authors propose a region proposal network-based speaker diarization (RPNSD) system for which a DNN simultaneously generates overlapped speech segment proposals and computes their speaker embeddings. Compared with standard diarization approaches, RPNSD is argued to offer shorter pipeline and can handle the overlapped speech.

One of the biggest challenges in speaker diarization is to determine the total number of speakers for each audio segment. In current systems, both the number of speakers and the segment-wise speaker labels are determined by clustering algorithm, making it a critical and most challenging block for diarization [29], [8]. Thus, in this paper we propose to investigate the use of a building block named speaker count to independently predict the number of active speakers in each considered audio segment. It will allow clustering algorithm to assign enough speakers in each audio segment, and therefore potentially offers better diarization performance. Note that, there have been studies about counting the number of speakers in single-channel recording [24], [25] or multichannel setting [4], [17]. But in these works, speaker count is considered as a separated task and is not investigated in the context of speaker diarization. Our contributions are three-fold and are summarized as follows:

- We consider speaker count as a building block for the diarization workflow (Section II-B). This block can be developed independently with other processing blocks, making it flexible to be integrated into any existing systems. Recent studies, such as in DIHARD challenge [22], [23], show that handling overlapped speech, a special case of speaker count, is crucial and remains an open problem.

- We provide a preliminary theoretical analysis (Section III) and investigate the effect of analyzing window size to draw the potential benefit of the overlap speech detection, a special case of speaker count, in terms of the diarization error rate (DER).
- We perform experiments on a benchmark dataset AMI Headset mix where the oracle speaker count block and the oracle overlap detection block is integrated into two state-of-the-art speaker diarization systems (Section IV). Diarization results confirm the potential benefit gained by the proposed block. Note that, as a preliminary study we focus on analyzing potential benefits offered by an ideal speaker count model and leave its practical development for future work.

The rest of the paper is organized as follows. We present the considered speaker diarization workflow with a new re-segmentation algorithm and two baseline systems in Section II. We then provide some theoretical analysis to show the upper benefit obtained by an ideal overlap speech detection in Section III. Experiment results on two benchmark datasets are discussed in Section IV. Finally we conclude in Section V.

II. Speaker diarization approach

A. Speaker count integrated workflow

Speaker diarization pipelines often contain three major building blocks: voice activity detection (VAD), speaker embedding, and speaker clustering [2]. VAD [16] is used as an important pre-processing step to eliminate non-voice segments from the input recording, while speaker embedding aims to extract discriminative speaker features for each speech segment. The clustering algorithm is crucial to label speaker identities in each segment. State-of-the-art approaches [22], [8] exploit additional blocks such as speaker change detection [27], overlapped speech detection [5], or even re-segmentation [10] for a better diarization performance. A general pipeline is illustrated by the black boxes in Fig. 1.

B. Resegmentation

In the considered approach, the speaker count block is integrated into diarization workflow to estimate the number of active speakers at each analysis window. Then, instead of just labeling one or two speakers with the highest scores resulted from the clustering block (as in case the conventional overlapped detection is used), resegmentation block will label speaker identities according to the exact speaker numbers reported from the speaker count block. The blue dotted boxes in Fig. 1 indicate such steps considered in the paper.

C. Baseline systems

In the following, we present Pyannote [8] and UIS-RNN [29] as two baseline systems where the speaker count can be integrated as a new building block.

1) Pyannote: The first baseline system we consider is based on a recently released PyTorch library named Pyannote. Pyannote incorporates a set of state-of-the-art trainable end-to-end neural building blocks that can be either trained separately or combined and jointly optimized to build speaker diarization pipelines: end-to-end neural voice activity detection (VAD) [20], speaker change detection [27], overlapped speech detection [10], speaker embeddings [6], and Bayesian model-based clustering [8]. While the first three blocks were all trained on the AMI datasets [11], the speaker embedding was trained on the VoxCeleb1 [1] and the VoxCeleb2 [13] datasets. Instead of relying on *x-vectors* extracted from a fixed-length sliding window as input to the clustering step like conventional speaker diarization systems [9], [21], Pyannote uses metric learning approach to train speaker embeddings that are directly optimized for a predefined (usually is cosine) distance. Thus it reduces the need for techniques like probabilistic linear discriminant analysis (PLDA) before clustering.

It is worth noting that, while each building block has to be initially trained separately, Pyannote combines them into a speaker diarization pipeline whose hyper-parameters are optimized jointly to minimize diarization error rate. This joint optimization process has been confirmed leading to better results than the late combination of multiple building blocks that were tuned independently from each other [28]. As a preliminary study, in this paper, we do not consider a joint optimization of the speaker count block with the others.

2) UIS-RNN: Our second considered baseline exploits the powerful DNN architectures for both three major steps: VAD, speaker embedding, [26], and speaker clustering [29]. For VAD, we use a pre-trained model developed for the Google WebRTC project[1]. It is reportedly one of the best available VAD for real-time processing[2]. In our implementation, speaker embeddings are extracted from the state-of-the-art speaker recognition deep network [26] trained on the VoxCeleb1 [1] and the VoxCeleb2 [13] datasets. This network modifies ResNet in a fully convolutional way to encode 2D spectrograms of audio signals, followed by a NetVLAD/GhostVLAD layer [3] for feature aggregation along the temporal axis. It produces a fixed-length 512-dimensional output d-vector for each input audio segment. The implementation is provided by the authors[3].

Unlike the Pyannote and other conventional unsupervised clustering approaches, this baseline uses the recently proposed supervised unbounded interleaved-state recurrent neural networks (UIS-RNN) [29] as clustering method. In UIS-RNN each speaker is modeled by an instance of RNN with shared parameters. As the total number of speakers in an audio recording is generally unknown, an unbounded number of RNN instances can be generated. It is claimed that within a fully supervised framework, UIS-RNN can better handle complexities in speaker diarization since it automatically learns

[1] https://webrtc.org/
[2] https://github.com/wiseman/py-webrtcvad
[3] https://github.com/WeidiXie/VGG-Speaker-Recognition

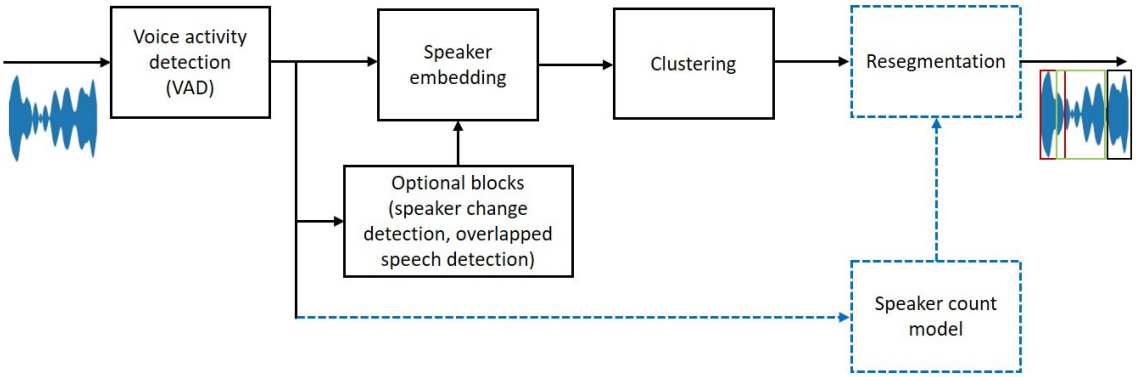

Fig. 1. The general workflow of the speaker diarization system. The blue square-dot boxes indicate the additional steps investigated in the paper.

both the speaker changes and the number of speakers within each utterance via a Bayesian non-parametric process. In our experiment, we use the UIS-RNN implementation provided by the Google AI Blog[4]. Note that, in this investigated baseline, DNN architectures for the VAD and the speaker embedding could be considered to be more advanced than the ones used in the original UIS-RNN paper [29]. Thus, we argue that this can be served as a strong baseline for speaker diarization.

III. Theoretical Analysis

We analyse the DER obtained by an *oracle* overlap speech detection-integrated diarization system and draw its upper gain compared to a basic setup where only one speaker is considered to be active at each analysis window. Let I be an audio file, according to [7], the diarization error rate (DER) of I is defined by:

$$\mathrm{DER}(I) = \frac{\sum_{\forall s} \mathrm{dur}(s)\{\max\{N_{ref}(s), N_{sys}(s)\} - N_{cor}(s)\}}{\sum_{\forall s} \mathrm{dur}(s) N_{ref}(s)}, \tag{1}$$

where s are all segments with duration $\mathrm{dur}(s)$ divided at every speaker change point; $N_{ref}(s)$, $N_{sys}(s)$ and $N_{cor}(s)$ are the ground-truth, the detected number of speakers, and the number of the speakers that are correctly determined at segment s, respectively. Throughout this paper, we denote by $\mathrm{DER}_n(I)$ and $\mathrm{DER}_d(I)$ the numerator and the denominator of $\mathrm{DER}(I)$. Note that only the numerator which contains $N_{sys}(s)$ is affected by the the overlap speaker detection or speaker count block. Thus in the following, we will focus on the $\mathrm{DER}_n(I)$ only.

Lemma 1
Suppose I is divided into k arbitrary segments $I_1, ..., I_k$, then:

$$\mathrm{DER}_n(I) = \sum_{i=1}^{k} \mathrm{DER}_n(I_i).$$

We skip the proof because it is trivial.

[4]https://github.com/ultralytics/yolov5

In the following, we will derive $\mathrm{DER}_n(I)$ in two cases: with and without the use of overlap detection. To ease the presentation, we denote by $\mathrm{DER}_n^w(I)$ and $\mathrm{DER}_n^o(I)$ the $\mathrm{DER}_n(I)$ in these two cases, respectively; N_{ref}^w, N_{ref}^o and N_{cor}^w, N_{cor}^o the values of N_{ref} and N_{cor} with and without the use of overlap detection, respectively. Moreover, we denote $\Delta\mathrm{DER}_n(I)$ the gain of the DER's nominator when using overlap detection, *i.e.,* $\Delta\mathrm{DER}_n(I) = \mathrm{DER}_n^o(I) - \mathrm{DER}_n^w(I)$.

Let wlen be the windows size which is used for both the diarization and overlap detection models. Suppose $I_1, ..., I_k$ are the segments obtained by dividing I using window with the size of wlen. To obtain the upper gain by the overlap detection, let us consider the *oracle* setting where the overlap detection is assumed to work perfectly. Specifically, a segment is detected as *overlapped segment* if and only if it contains an overlapped duration. We assume additionally that the confidence score determined by the diarization model is proportional with the speaking duration of the speaker, *i.e.,* speaker who is active more will have a higher confidence score.

Lemma 2
Considering a segment I_i which contains more than one speaker. Suppose I_i is comprised of $O_i^1, ..., O_i^{k_i}$, where O_i^j is the duration containing exact j speakers. Moreover, suppose S_1 and S_2 are two speakers whose confidence scores are the highest and the second-highest among all the speakers appearing in I_i. Let us denote by $\mathbb{S}_1, \mathbb{S}_2$ and $\mathbb{S}_3$ be the sets of duration containing only S_1, only S_2, and only one speaker that is neither S_1 nor S_2; $\mathbb{S}_{1,2}$ be the set of duration contain both S_1 and S_2. The following statement concerning the gain/loss when using overlap detection compared to a basic setup where only one speaker is assigned at each analysis window holds:

$$\Delta\mathrm{DER}_n(I_i) = \mathrm{dur}(\mathbb{S}_{1,2}) - \mathrm{dur}(\mathbb{S}_1) - \mathrm{dur}(\mathbb{S}_3). \tag{2}$$

Proof
According to our ideal assumption, the prediction results of the models with and without overlap detection are $\{S_1, S_2\}$

and $\{S_1\}$, respectively. According to the Lemma 1, we have

$$\begin{aligned}\mathrm{DER}_n(I) &= \sum_{j=1}^{k_i} \mathrm{DER}_n(O_i^j) \\ &= \textstyle\sum_{\forall s \in O_i^j} \mathrm{dur}(s)\{\max\{N_{ref}(s), N_{sys}(s)\} - N_{cor}(s)\}.\end{aligned}$$

For every O_i^j, we have $N_{ref}^w(s) = N_{ref}^o(s) = j$, $N_{sys}^w(s) = 2$ and $N_{sys}^o(s) = 1$. The value of $N_{cor}(s)$ is decided by s and the used model as follows.

- $N_{cor}^o(s) = 1$ for all $O_i^j \in \mathbb{S}_1 \bigcup \mathbb{S}_{1,2}$, and $N_{cor}^o(s) = 0$, otherwise.
- $N_{cor}^w(s) = 1$ for all $O_i^j \in \mathbb{S}_1 \bigcup \mathbb{S}_2$, $N_{cor}^w(s) = 2$ for all $O_i^j \in \mathbb{S}_{1,2}$, and $N_{cor}^w(s) = 0$, otherwise.

Therefore,

$$\begin{aligned}\mathrm{DER}_n^o(I_i) = &\sum_{\forall s \in O_i^j \& O_i^j \notin \mathbb{S}_1 \bigcup \mathbb{S}_{1,2}} \mathrm{dur}(s) \times j \\ &+ \sum_{\forall s \in O_i^j \& O_i^j \in \mathbb{S}_{1,2}} \mathrm{dur}(s), \qquad (3)\\ \mathrm{DER}_n^w(I) = &\sum_{\forall s \in O_i^j \& O_i^j \in \mathbb{S}_1 \bigcup \mathbb{S}_2} \mathrm{dur}(s) \\ &+ \sum_{\forall s \in O_i^j \& O_i^j \in \mathbb{S}_3} 2 \times \mathrm{dur}(s) \\ &+ \sum_{\substack{\forall s \in O_i^j \\ \& O_i^j \notin \mathbb{S}_1 \bigcup \mathbb{S}_2 \bigcup \mathbb{S}_3 \bigcup \mathbb{S}_{1,2}}} \mathrm{dur}(s) \times j. \qquad (4)\end{aligned}$$

Then, the gain when using overlap detection is as follows

$$\begin{aligned}\Delta\mathrm{DER}_n(I_i) &= -\sum_{\substack{\forall s \in O_i^j \\ \& O_i^j \in \mathbb{S}_1 \bigcup \mathbb{S}_3}} \mathrm{dur}(s) + \sum_{\substack{\forall s \in O_i^j \\ \& O_i^j \in \mathbb{S}_{1,2}}} \mathrm{dur}(s) \qquad (5)\\ &= \mathrm{dur}(\mathbb{S}_{1,2}) - \mathrm{dur}(\mathbb{S}_1) - \mathrm{dur}(\mathbb{S}_3). \qquad (6)\end{aligned}$$

Corollary 1

If I_i does not contain any speaker change point, then:

- $\Delta\mathrm{DER}_n(I_i) = 0$*, if I_i is a non-overlap segment,*
- $\Delta\mathrm{DER}_n(I_i) = \mathrm{dur}(I_i)$*, otherwise.*

Proof

If I_i is a non-overlap segment, then $\mathrm{DER}_n^o(I_i) = \mathrm{DER}_n^w(I_i) = 0$; thus, $\Delta\mathrm{DER}_n(I_i) = 0$. If I_i is an overlap segment without of speaker change point, then $\mathbb{S}_1 = \mathbb{S}_3 = \varnothing$. According to Lemma 2, $\Delta\mathrm{DER}_n(I_i) = \mathrm{dur}(\mathbb{S}_{1,2}) = \mathrm{dur}(I_i)$.

Theorem 1

For an arbitrary window size $wlen > 0$, there exists a window size $wlen' < wlen$ such that the upper DER gain obtained when using the overlap detection with the window size of $wlen'$ is greater than that when using $wlen$.

Proof

For each I_i $(i = 1, ..., k)$, let us denote by s_i^j $(j = 1, ..., l_i)$ all the segments obtained by dividing I_i at every speaker change point. We prove the following hypothesis by contradiction: *"we can choose a window size $wlen_1$ such that when we divide the input audio file I by a window with the size of $wlen_1$, then, every obtained segment contains at most one speaker change point"*. Let us choose wlen_1 as a divisor of wlen (i.e., there is an integer number n such that $\mathrm{wlen} = n \times \mathrm{wlen}_1$) that is smaller than $\min_{\forall i,j} \mathrm{dur}(s_i^j)$ (1). Let $\mathbb{S}'$ denote the set of all segments obtained by dividing I using a window with the size of wlen_1. Suppose there is an item of $\mathbb{S}'$ containing more than one speaker change point, that segment must contain at least one segment among s_i^j $(j = 1, ..., l_i; i = 1, ..., k)$, say s_{i*}^{j*}. Then, the length of that segment is greater than $\mathrm{dur}(s_{i*}^{j*})$. This contradicts with condition (1). The hypothesis is proved.

Now, let's use a window with the size of wlen_1 to divide I. For each I_i, let $\mathbb{A}_i$ be the set of all segments that do not contain any speaker change point, and $\mathbb{B}_i$ be the set of segments containing only one speaker change point. From Lemma 1, we deduce that

$$\Delta\mathrm{DER}_n(I_i) = \sum_{\forall s \in \mathbb{A}_i} \Delta\mathrm{DER}_n(s) + \sum_{\forall s \in \mathbb{B}_i} \Delta\mathrm{DER}_n(s). \qquad (7)$$

From Corollary 1, we have

$$\begin{aligned}\Delta\mathrm{DER}_n(I_i) &= \sum_{\substack{\forall s \in \mathbb{A}_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) + \sum_{\forall s \in \mathbb{B}_i} \Delta\mathrm{DER}_n(s) \\ &= \sum_{\substack{\forall s \in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) - \sum_{\substack{\forall s \in \mathbb{B}_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) \\ &\quad + \sum_{\forall s \in \mathbb{B}_i} \Delta\mathrm{DER}_n(s) \qquad (8)\end{aligned}$$

According to Lemma 2, we have

$$\Delta\mathrm{DER}_n(s) \geq -\mathrm{dur}(s). \qquad (9)$$

From (8) and (9), we deduce that

$$\Delta\mathrm{DER}_n(I_i) \geq \sum_{\substack{\forall s \in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) - 2 \sum_{\forall s \in \mathbb{B}_i} \mathrm{dur}(s). \qquad (10)$$

Note that as $\mathbb{B}_i$ is the set of all segments that consist of exactly one speaker change point, the cardinality of $\mathbb{B}_i$ cannot exceed the number of speaker change points of I_i. Let n_i denote the number of speaker change points of I_i, then from (9), we have

$$\Delta\mathrm{DER}_n(I_i) \geq \sum_{\substack{\forall s \in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) - 2n_i \mathrm{wlen}_1. \qquad (11)$$

From (6) and (11), it can be seen that, when wlen_1 is smaller than $\min_{\forall i} \frac{\mathrm{dur}(\mathbb{S}_1) + \mathrm{dur}(\mathbb{S}_3)}{2n_i}$, the gain obtained by using wlen_1 is greater than that when using wlen.

Theorem 2

Let $wlen$ be the window size, then the following holds

$$\lim_{wlen \to 0} \Delta\mathrm{DER}_n(I) = \sum_{\substack{\forall s \in I \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s).$$

Proof

Let the window size $wlen$ be a sufficiently small number, then from Lemma 2, we deduce that

$$\Delta\mathrm{DER}_n(I_i) \leq \sum_{\substack{s \in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s). \qquad (12)$$

On the other hand, from Theorem 1, we have

$$\Delta\mathrm{DER}_n(I_i) \geq \sum_{\substack{s\in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) - 2n_i\mathrm{wlen}. \quad (13)$$

From (12) and (13) the following holds

$$\sum_{\substack{s\in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) \geq \Delta\mathrm{DER}_n(I) \geq \sum_{\substack{s\in I \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) - 2\mathrm{wlen}\sum_{i=1}^{k} n_i.$$

It means that

$$\sum_{\substack{s\in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) \geq \lim_{\mathrm{wlen}\to 0} \Delta\mathrm{DER}_n(I) \geq \sum_{\substack{s\in I \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s) - \lim_{\mathrm{wlen}\to 0} 2w\sum_{i=1}^{k} n_i.$$

Consequently,

$$\lim_{\mathrm{wlen}\to 0} \Delta\mathrm{DER}_n(I) = \sum_{\substack{s\in I_i \\ \&\ s \text{ is overlap}}} \mathrm{dur}(s). \quad (14)$$

As can be seen from the Theorem 1 and the Theorem 2, the benefit of the overlap speech detection is greater when (a) the input audio contains more segments with overlap and (b) the analyzing window size is smaller. These intuitions still hold for the use of speaker count, a more general setting, proposed in this paper as shown in our experiment results in Section IV. We leave such proof for future work.

IV. Experiment

A. Data and evaluation metrics

We evaluate the speaker diarization performance on the AMI Headset mix dataset [12]. This is a widely used dataset for speaker diarization over the last decade, which consists of 98 hours of meeting recordings from 180 speakers in total. The meetings were in English and recorded in three rooms with different acoustic properties. In the dataset 81% of the total speech in voiced periods is single-speaker and 15% of the time is two-speaker, leaving approximately 4% of the time to three or more speakers. This implies that the two-speaker situation accounts for about 75% of the overlap regions. The dataset was split into 70% for training (68.6 hours), 15% for validation (14.7 hours), and 15% for evaluation (14.7 hours).

We use the *pyannote.metrics toolkit* [7] to evaluate the speaker diarization performance. Three widely-used metrics are computed: diarization error rate (DER), Jaccard error rate (JER), and B3-F1 score. DER is defined in equation (1). JER is a similarity measure typically used to evaluate the output of segmentation systems and is defined as the ratio between the intersection and union of two segmentations. For DER and JER the lower value the better, while for B3-F1 score the higher the better.

B. Implementation details

Pyannote baseline: we use configuration files, implementation codes, and pre-trained models for all processing blocks as they were already trained and validated on the AMI datasetby the authors[5]. Speaker embeddings of dimension 512 are extracted every 1-second sliding window for clusteringWe then evaluate Pyannote baseline performance on the test set of the dataset.

UIS-RNN baseline: For training the *d-vectors* speaker embedding, the VoxCeleb1 [1] and the VoxCeleb2 [13] datasets are further augmented with approximately 1,000 hours of English speeches from ST Chinese Mandarin Corpus[6], and approximately 34 hours of Japanese speeches collected from Youtube. We set a varying size spectrogram corresponding to 2-6 second temporal segment, extracted randomly from each utterance. Spectrograms are computed via the short-term Fourier transform (STFT) with 256 frequency bins, a sliding Hamming window of size 25 ms, and a window shift of 10 ms. The spectrograms are then normalized by subtracting the mean and dividing by the standard deviation of all frequency components in a single time step. For training the UIS-RNN clustering on the AMI headset mix dataset, we use the parameter settings in the original implementation, except the sliding window for speaker embedding extraction is 1 second instead of 240 ms. Similar to the Pyannote baseline setting, during the evaluation, speaker embeddings of dimension 512 are extracted every 1-second sliding window for clustering.

TABLE I
SPEAKER DIARIZATION RESULTS OBTAINED BY THE PYANNOTE-BASED METHODS ON THE AMI HEADSET MIX DATASET. THE ORIGINAL MODEL IS TRAINED AND PROVIDED BY THE AUTHORS.

Pyannote-based methods	Window size (seconds)	DER%	JER%	B3-F1
Original model	1	32,09	99.15	0.59
One speaker assignment	1	29.36	59.36	0.63
Oracle overlap detection	1	34.28	59.68	0.56
	0.8	32.46	58.09	0.57
	0.6	30.12	57.58	0.58
	0.4	28.85	57.64	0.6
	0.2	27.83	56.18	0.63
	Oracle speaker change	25.6	55.98	0.64
Oracle speaker count	1	29.13	58.44	0.61
	0.8	26.24	55.98	0.64
	0.6	25.33	54.82	0.65
	0.4	23.75	53.91	0.65
	0.2	21.87	52.37	0.65
	Oracle speaker change	20.62	51.05	0.65

C. Diarization results

In order to evaluate the potential benefit of the speaker count integration, for each considered baseline, we investigate four system setups as follows:

[5]https://github.com/pyannote/pyannote-audio
[6]http://openslr.org/38

TABLE II
SPEAKER DIARIZATION RESULTS OBTAINED BY THE UIS-RNN-BASED METHODS ON AMI HEADSET MIX DATASET. THE ORIGINAL UIS-RNN MODEL IMPLEMENTATION IS PROVIDED BY THE GOOGLE AI BLOG, BUT TRAINED BY OURSELVES.

UIS-RNN-based methods	Window size (seconds)	DER%	JER%	B3-F1
Original model	1	30.87	59.06	0.61
One speaker assignment	1	28.52	64.91	0.59
Oracle overlap detection	1	30.96	58.69	0.56
	0.8	30.7	58.09	0.57
	0.6	28.7	57.33	0.58
	0.4	27.64	56.92	0.6
	0.2	26.18	55.74	0.63
	Oracle speaker change	24.7	54.8	0.64
Oracle speaker count	1	28.41	55.44	0.65
	0.8	27.32	53.48	0.67
	0.6	24.27	51.78	0.67
	0.4	22.6	50.36	0.67
	0.2	21.03	49.12	0.67
	Oracle speaker change	18.74	46.32	0.7

- Original model: This is the baseline diarization workflow where speaker embeddings and speaker clustering are performed for every 1 s audio segment along with each input audio file. For Pyannote, all processing blocks use the pre-trained model provided by the authors. For UIS-RNN, only VAD uses a pre-trained model while speaker embedding and clustering models are trained by ourselves as detailed in Section IV-B.
- One speaker assignment: In this setup, all blocks are similar to the original model, except that the clustering algorithm assigns only one speaker with the highest appearance probability for each 1 s audio segment. This simple clustering setting does not require thresholding to specify active speakers in each audio segment and shows us the diarization result without the use of overlap detection and speaker count blocks.
- Oracle overlap detection: In this setup, again all blocks are similar to the original model, except for the use of the *oracle* overlap detection instead of the trained DNN model. Here we assume that the overlap detection for each analyzing audio window is perfect (*i.e.,* known from the ground-truth) in order to evaluate the upper-bound diarization performance with the use of overlap detection block with different analyzing window sizes. We vary the window size as 0.2s, 0.4s, 0.6s, 0.8s, 1s to investigate its effect. In windows with more than two active speakers, we choose the two ones with the longest active duration. The best diarization performance is obtained when oracle speaker change is used where the speaker activity (*i.e.,* active or inactive) boundary is perfectly specified.
- Oracle speaker count: This setup allows us to investigate the potential benefit of the speaker count block when it is integrated into current baseline systems. Similarly to the oracle overlap detection case, we vary the speaker count window size as 0.2s, 0.4s, 0.6s, 0.8s, 1s to analyze its effect, and the best performance is obtained with the use of the oracle speaker change. In each window, the number of active speakers, which can be more than two, is perfectly specified given the ground-truth.

Speaker diarization results obtained by the different variants of the Pyannote baseline and the UIS-RNN baseline are summarized in Table I and Table II, respectively. First, it is interesting to see that the simple one speaker assignment setting offers better DER, the most important evaluation metric, than the two original baseline models on the AMI Headset mix dataset which contains about 19% of multiple speaker cases. This shows that clustering is still very challenging for overlapping speeches, and reveals the need for the speaker count building block. It is also not surprising that the average result is better when the overlap detection is used, especially with a small window size. Finally, as expected, the proposed approach integrating speaker count block offers the best diarization performance (in terms of DER and B3-F1 score) in both two baselines.

It is worth noting that, in this oracle overlap detection and the oracle speaker count settings, the general diarization results are not better than those obtained by the original models or the one speaker assignment setting in both two baselines when the window size is 1 second. This is due to the fact that in many analyzing windows, the actual speech overlapping duration is less than 1 second. Thus, assigning two speakers (for the overlap detection case) and multiple speakers (for the speaker count case) for all such long windows is less accurate. With this intuition, the smaller window size allows speaker assignment to be closer to the ground-truth, and therefore the better diarization performance is observed as shown in Table I and Table II. The best performance is obtained by the speaker count integrated approach with the oracle speaker change decision: DER is as low as 20.62% for the Pyannote and 18.74% for the UIS-RNN.

V. CONCLUSION

In this paper, we address the problem of efficiently handling overlapping speech in speaker diarization systems. For such purpose, we first introduce a building block, which can be easily integrated into existing diarization workflows, to independently count the number of active speakers in each audio window in order to better label speakers. We then discuss the upper gain offered by an ideal overlap speech detection, a special and simpler case of the speaker count, via a theoretical analysis. Finally, we perform experiments where the speaker count block is integrated into two strong diarization baselines to confirm its potential benefit in a real-world dataset, and to further investigate the effect of the window size. Future work would be devoted to develop and train a DNN-based speaker count model, *e.g.,* motivated from the Countnet [25] or the CRNN approach [17], for a practical diarization application.

ACKNOWLEDGMENT

This work was majorly funded by Aimenext Joint Stock Company (Aimenext JSC). Thanh T. H. Duong has been

supported by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 07/2020/STS01 and Hanoi University of Mining and Geology under grant number 97/QD-MDC.

REFERENCES

[1] W. X. A. Z. A. Nagrani, J. S. Chung. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech Language*, 2019.

[2] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, 2012.

[3] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] S. Arberet, R. Gribonval, and F. Bimbot. A robust method to count and locate audio sources in a multichannel underdetermined mixture. *IEEE Transactions on Signal Processing*, 58(1):121–133.

[5] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland. Overlapped speech detection for improved speaker diarization in multiparty meetings. In *Proc. IEEE Int. Conf. on Audio, Speech, and Signal Processing (ICASSP)*, page 4353–4356, 2008.

[6] H. Bredin. pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, August 2017.

[7] H. Bredin. pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems. In *18th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Stockholm, Sweden, August 2017.

[8] 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 *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7124–7128, 2020.

[9] P.-A. Broux, F. Desnous, A. Larcher, S. Petitrenaud, J. Carrive, and S. Meignier. S4D: Speaker Diarization Toolkit in Python. In *Interspeech 2018*, pages 1368–1372. ISCA, Sept. 2018.

[10] L. Bullock, H. Bredin, and L. P. García-Perera. Overlap-aware diarization: resegmentation using neural end-to-end overlapped speech detection. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7114–7118, 2020.

[11] J. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation*, 41(2), 2007.

[12] J. Carletta. Unleashing the killer corpus: Experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190, 2007.

[13] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *Proc. INTERSPEECH*, 2018.

[14] G. S. et al. Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural DIHARD challenge. In *Proc. INTERSPEECH*, page 2808–2812. ISCA, 2018.

[15] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe. End-to-end neural speaker diarization with permutation-free objectives, 2019.

[16] G. Gelly and J.-L. Gauvain. Optimization of RNN-based speech activity detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, (3):646–656, 2018.

[17] P. A. Grumiaux, S. Kitíc, L. Girin, and A. Guérin. High-resolution speaker counting in reverberant rooms using CRNN with ambisonics features. In *European signal processing conference (EUSIPCO)*, 2020.

[18] Z. Huang, S. Watanabe, Y. Fujita, P. García, Y. Shao, D. Povey, and S. Khudanpur. Speaker diarization with region proposal network. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6514–6518, 2020.

[19] M. Huijbregts, D. Leeuwen, and F. Jong. Speech overlap detection in a two-pass speaker diarization system. In *Proc. INTERSPEECH*, 2009.

[20] M. Lavechin, M.-P. Gill, R. Bousbib, H. Bredin, and L. P. Garcia-Perera. End-to-end Domain-Adversarial Voice Activity Detection. 2020.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The kaldi speech recognition toolkit. 2011. IEEE Catalog No.: CFP11SRW-USB.

[22] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman. Second DIHARD challenge evaluation plan. In *Tech. Rep*, 2019.

[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, 2021.

[24] H. Sayoud and S. Ouamour. Proposal of a new confidence parameter estimating the number of speakers – An experimental investigation. *Journal of Information Hiding and Multimedia Signal Processing*, 1(2):101–109.

[25] F.-R. Stoter, S. Chakrabarty, B. Edler, and E. A. P. Habets. Countnet: Estimating the number of concurrent speakers using supervised learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(2):268–282, Feb 2019.

[26] J. S. C. A. Z. W. Xie, A. Nagrani. Utterance-level aggregation for speaker recognition in the wild. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP*, pages 5791–5795, 2019.

[27] R. Yin, H. Bredin, and C. Barras. Speaker change detection in broadcast TV using bidirectional long short-term memory networks. In *Proc. INTERSPEECH*. ISCA, 2017.

[28] R. Yin, H. Bredin, and C. Barras. Neural speech turn segmentation and affinity propagation for speaker diarization. In *Proc. Interspeech 2018*, pages 1393–1397, 2018.

[29] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang. Fully supervised speaker diarization. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6301–6305, 2019.