# Improving Speaker Assignment in Speaker-Attributed ASR for Real Meeting Applications

Can Cui, Imran Ahamad Sheikh, Mostafa Sadeghi, Emmanuel Vincent

# Improving Speaker Assignment in Speaker-Attributed ASR for Real Meeting Applications

Can Cui[1,2], Imran Sheikh[2], Mostafa Sadeghi[1], Emmanuel Vincent[1]

[1]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2]Vivoka, Metz, France

[1]firstname.lastname@inria.fr,[2]firstname.lastname@vivoka.com

## Abstract

Past studies on end-to-end meeting transcription have focused on model architecture and have mostly been evaluated on simulated meeting data. We present a novel study aiming to optimize the use of a Speaker-Attributed ASR (SA-ASR) system in real-life scenarios, such as the AMI meeting corpus, for improved speaker assignment of speech segments. First, we propose a pipeline tailored to real-life applications involving Voice Activity Detection (VAD), Speaker Diarization (SD), and SA-ASR. Second, we advocate using VAD output segments to fine-tune the SA-ASR model, considering that it is also applied to VAD segments during test, and show that this results in a relative reduction of Speaker Error Rate (SER) up to 28%. Finally, we explore strategies to enhance the extraction of the speaker embedding templates used as inputs by the SA-ASR system. We show that extracting them from SD output rather than annotated speaker segments results in a relative SER reduction up to 20%.

## 1. Introduction

Multi-speaker meeting transcription has recently become a vibrant and active research topic [1–4]. This task poses various challenges, including distant microphones, overlapped speech, and ambient noise. The earliest works on end-to-end (E2E) models for this task [5, 6] had demonstrated the advantages of an E2E model over a traditional system wherein different components address different subtasks. Recent research primarily concentrates on enhancing the E2E model architectures [7–9]. The work in [7] introduced a Transformer-based E2E Speaker-Attributed ASR (SA-ASR) system for simultaneous recognition of speech and speaker identities. Thereafter, SA-ASR was extended to accommodate diverse scenarios [10], address an unlimited number of speakers [11], and cater to streaming applications [12]. Moreover, non-autoregressive models for multi-speaker ASR [13] have been proposed.

While the model architecture is undoubtedly crucial, the application of the E2E model to real-world data has not received much attention. Due to the lack of significant amounts of real meeting-style training data, training of multi-speaker ASR models relies on simulated multi-speaker overlapped distant speech data. To improve performance on real test data, these models are adapted or fine-tuned on real training data. A few studies [14–16] have discussed training and evaluation on real recorded multi-speaker datasets, such as the AMI meeting corpus [17] which still remains a challenging multi-speaker speech transcription task.

Training and inference on real long-length multi-speaker audio recordings is a non-trivial issue due to computational and memory requirements. One approach to address this issue is to segment the long recording at silence positions into disjoint segments [14]. Another approach is to segment it into fixed-sized chunks and adjust the start/end times to non-overlapped word boundaries [15]. Both approaches rely on the ground-truth utterance or word boundaries. In real-life applications, this information is not available and a Voice Activity Detection (VAD) system becomes essential for obtaining speech segments.

A second problem with training and inference on real multi-speaker audio is how to obtain the speaker embedding templates used as inputs by the SA-ASR system. In real-life applications, front-end Speaker Diarization (SD) becomes essential for this purpose. While studies on VAD [18, 19] and SD [20, 21] exist independently, there is little discussion on how to optimally integrate them into an SA-ASR pipeline to create a ready-to-use meeting transcription system. Speaker assignment in SA-ASR is typically performed using an x-vector-based speaker embedding model [22]. Preparation of a representative template for each speaker typically involves averaging the embeddings of candidate segments [7, 8, 15]. However, the selection of segments for better representing each speaker remains a question.

The contributions of this paper include: (a) a VAD-SD-SA-ASR pipeline for real meeting transcription; (b) fine-tuning of SA-ASR on VAD output segments instead of ground-truth or fixed-sized segments to better fit the test conditions; (c) a discussion on strategies to improve speaker assignment in SA-ASR by leveraging various attributes related to VAD and SD, such as the number and length of segments used to obtain speaker embedding templates. Experiments show that our proposed fine-tuning method reduces the SER by up to 28% relative and that extracting speaker embedding templates from SD output rather than annotated speaker segments results further reduces it by up to 20% relative. To the best of our knowledge, this is the first study that investigates how attributes such as the number and the length of segments impact the accuracy of the speaker embedding templates and the resulting SA-ASR performance.

The rest of the paper is organized as follows. Section 2 presents the individual modules in our pipeline, including the segmentation methods for row meeting audio. Section 3 describes our pipeline approach and the preparation of the training, development, and test data. Section 4 presents our experimental settings, followed by the discussion of results in Section 5. Finally, Section 6 provides a conclusion.

## 2. Background

### 2.1. Voice Activity Detection: CRDNN

A Voice Activity Detection (VAD) system identifies speech and non-speech segments in an audio signal, which is crucial for applications like speech recognition. Neural network-based

models for VAD often include architectures like Convolutional Neural Networks [23], Recurrent Neural Networks [24], Long Short-Term Memory networks [25], etc. One of the common architectures is CRDNN (Convolutional, Recurrent, and Dense Neural Network) [26, 27]. It combines convolutional layers for frequency variations, recurrent layers for temporal modeling, and dense layers for feature mapping, which have succeeded in various speech-related tasks.

### 2.2. Speaker Diarization: ECAPA-TDNN

Classical Speaker Diarization (SD) systems commonly rely on clustering speaker embeddings [20] such as i-vectors [28], d-vectors [29], and x-vectors [22]. Specifically, ECAPA-TDNN (Emphasized Channel Attention, Propagation, and Aggregation in Time-Delay Neural Network) [30] is a model used for computing x-vectors, which are widely employed for speaker representation in SD tasks. It enhances the traditional Time-Delay Neural Network (TDNN) architecture by integrating channel attention mechanisms, which dynamically emphasize relevant information and suppress noise, resulting in more robust speaker embeddings. ECAPA-TDNN utilizes both forward and backward information propagation to capture long-term dependencies and context in input sequences, ensuring effective modeling of temporal relationships for speaker verification. Additionally, it employs aggregation techniques to combine information across multiple frames, creating a comprehensive representation of speaker characteristics.

### 2.3. End-to-end Speaker-Attributed ASR

A Transformer-based end-to-end Speaker-Attributed ASR (SA-ASR) system was proposed in [7]. Following the Serialized Output Training (SOT) principle [31], the output is the concatenation of all speakers' sentences in first-in-first-out order, where each token is associated with one speaker ID and distinct speakers are separated by an <sc> token. As shown in Figure 1, the inputs to the model consist of an acoustic feature sequence and a set of reference speaker embeddings (a.k.a. speaker embedding templates) obtained from enrollment data. A Conformer-based ASR Encoder first encodes acoustic information, along with a Speaker Encoder to encode speaker information. Subsequently, the Transformer-based ASR Decoder and Speaker Decoder modules decode text and speaker information, respectively. The ASR block and the speaker block mutually share information through the ASR Decoder and the Speaker Decoder, enabling mutual assistance between text recognition and speaker identification. In addition to the ASR decoder's output, which consists of tokens corresponding to different speakers, the Speaker Decoder computes a speaker representation corresponding to each ASR output token. This representation is used to assign speakers by computing a dot product with the speaker embedding templates.

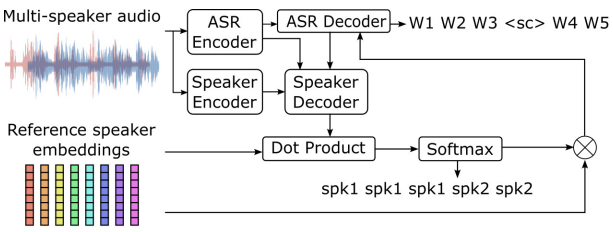


Figure 1: Diagram of SA-ASR [7].

### 2.4. AMI segmentation

#### 2.4.1. *Based on fixed-sized chunks*

The AMI corpus has been recently segmented using fixed chunk and hop sizes [15]. If the start/end time of a segment is within an overlapped speech region, it is modified to be 2 s outside of the overlap region. If the start/end time of a segment falls within a word, it is adjusted to the start/end time of that word using AMI's word boundary annotations. This approach simulates how to process meeting audio in situations where ground-truth silence/non-speech positions between utterances are not available at test time. However, speaker overlap and word boundary annotations are still required.

#### 2.4.2. *Based on ground-truth silence positions*

Multi-speaker ASR systems applied on the AMI corpus typically use ground-truth segment annotations for fine-tuning and test [14, 16]. More specifically, in [14], the concept of an "utterance group" is introduced by segmenting the recording at silence/non-speech positions between utterances. This allows the model to transcribe text and speaker information more effectively using the first-in-first-out principle.

## 3. Proposed methods

### 3.1. Overall system

Our proposed system consists of VAD, SD, and SA-ASR modules. Figure 2 illustrates the overall pipeline, which starts with VAD to divide the entire meeting into speech and non-speech segments. The VAD system takes as input the single-channel signal obtained by delay-and-sum beamforming applied to the AMI 8-channel Multiple Distant Microphone (MDM) signal. Since SA-ASR requires speaker embedding templates as inputs, as mentioned in Section 2.3, we need an SD model to determine the number of speakers in the entire meeting and identify the speech segments for each speaker. We use the results of VAD to perform SD on speech segments. Subsequently, based on the diarization results, we use the non-overlapped portions of speech from each speaker to compute an average speaker embedding for that speaker. Finally, the segments obtained through VAD and the speaker embedding templates obtained through SD serve as input to SA-ASR, allowing us to retrieve the speech content of all different speakers in a first-in-first-out order. The SA-ASR system is single-channel and operates on the first channel of the AMI MDM signal.

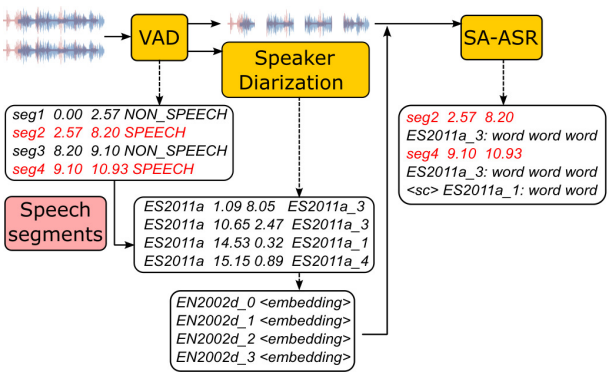


Figure 2: Pipeline of the proposed system.

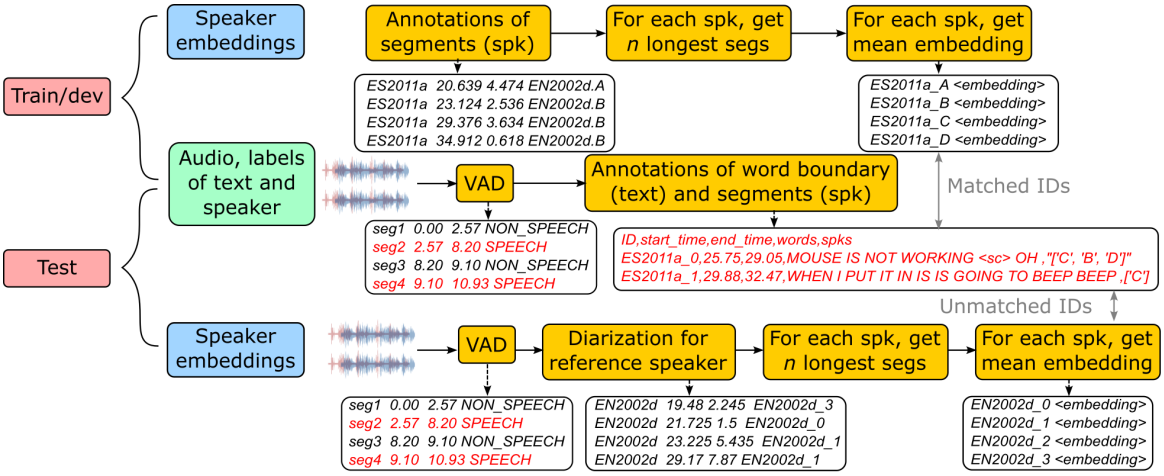


Figure 3: Preparation of the AMI corpus for the training, development, and test sets.

### 3.2. Data preparation

In real-life applications, ground-truth silence positions and utterance/word boundaries are not available at test time, so testing requires the use of VAD for segmentation. In [15], the SA-ASR model fine-tuned on fixed-sized segments and tested on VAD segments was found to yield poor results, which can be attributed to mismatched fine-tuning and test segment lengths. To adapt the model to the length of VAD segments during test, we also utilize VAD segments during the training phase. The following outlines the preparation of training, development, and test data on the AMI corpus.

#### *3.2.1. Preparation of training, development, and test sets*

The process of obtaining speech segments, speaker, and text labels is the same for training, development, and test data. As shown in Figure 3, all meeting data is segmented into variable-sized speech segments using VAD. We merge adjacent speech segments separated by a silence shorter than a given duration threshold, resulting in an average segment length which depends on the specified threshold. Using AMI's annotation files with information about speaker segments and word boundaries, we assign text and speaker sequences to each segment in a first-in-first-out order.

The process of obtaining speaker embedding templates differs for training and development vs. test. For the training and validation sets, we extract all speech segments for each speaker in each meeting based on the ground-truth annotated speaker segments, and select the $n$ longest segments for each speaker. We use a pretrained speaker embedding model to embed all $n$ segments and calculate the average embedding as the template for the associated speaker. The process for the test set is different because, in real test scenarios, the number of speakers is not known a priori. To solve this issue, we apply the SD model to the VAD segments to estimate the number of speakers and the speech segments for each speaker. The speaker embedding templates for all speakers are then derived in the same way as for the training and validation sets.

#### *3.2.2. Remapping of speaker IDs on the test set*

When computing speaker embedding templates for the training and development sets, the speaker IDs align with the labels used for speaker identification, as they both rely on the same speaker segment annotation file. However, during test, the SD results may assign a new ID to the recognized speakers, causing a mismatch with the ground-truth speaker IDs from the original annotation file (Figure 4). This necessitates remapping SD speaker IDs to ground-truth speaker IDs for evaluation purposes.

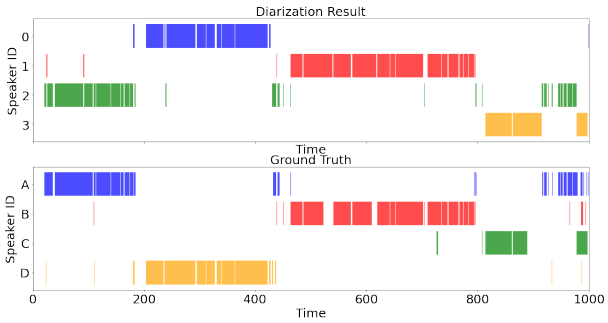


Figure 4: Mismatch of SD and ground-truth speaker IDs on the test set.

As shown in Figure 5, to remap two sets of IDs, assuming SD estimates with $I$ speakers and ground truth with $K$ speakers, we extract a list of non-overlapped speech segments for each speaker $i$ from the SD output, denoted $s_i$, and a list of individual speech segments for each speaker $k$ from the ground truth, denoted $s_k$. For each of the $I$ speakers, we calculate the Intersection over Union (IoU) values between the lists of speech segments for each $s_i$ and $s_k$. IoU is a measure commonly used to evaluate the overlap between two sets. It is defined as the size of the intersection divided by the size of the union of the two sets. For $s_i$ and $s_k$ representing speaker segments, IoU can be applied as follows:

$$\text{IoU}(s_i, s_k) = \frac{|s_i \cap s_k|}{|s_i \cup s_k|}, \tag{1}$$

where $|s_i \cap s_k|$ is the size (i.e., the total duration) of the intersection of lists $s_i$ and $s_k$, and $|s_i \cup s_k|$ is the size of the union of lists $s_i$ and $s_k$. We then assign the IDs from the list with the highest IoU value as the corresponding ID for that speaker.

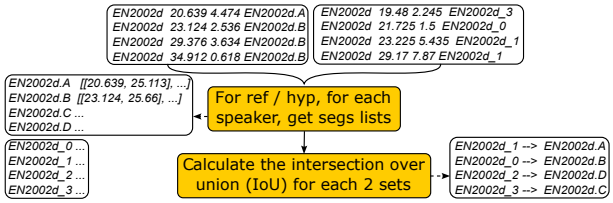


Figure 5: Remapping of SD speaker IDs to ground-truth speaker IDs on the test set.

# 4. Experimental settings

## 4.1. Dataset and metrics

### *4.1.1. Simulated multi-speaker LibriSpeech corpus*

To achieve optimal performance in AMI, it is imperative to undergo pretraining with a larger and cleaner dataset [14–16]. We pretrained the SA-ASR model on distant-microphone multi-speaker data simulated using the LibriSpeech corpus [32]. We utilized the train-960 and dev-clean subsets to construct our training and development datasets, creating a far-field microphone and multi-speaker set with 960 hours for training and 20 hours for development. We generate Room impulse responses (RIRs) using the gpuRIR toolkit [33]. The length, width and the height of the room are randomly drawn in the range of 3 to 8 m and 2.4 to 3 m, respectively. The microphone and speaker positions are randomly sampled with the constraints that the microphone is at most 0.5 m away from the room center, the speakers are at least 0.5 m away from the walls, and their heights are 0.6 to 0.8 m above the middle plane. The RT60 value is randomly drawn in the range of 0.4 to 1 s. A multi-speaker signal is generated by randomly drawing one utterance from each of 1 to 3 speakers, mixing these utterances convolved by RIRs, and concatenating the corresponding transcripts separated by the <sc> token. The start time of each utterance is shifted relative to the previous one, with a delay ranging from 0.5 seconds to the maximum duration of the previous utterance. This realistic choice also guarantees the first-in-first-out principle behind SOT. Each multi-speaker signal is linked to 8 speaker embedding templates, encompassing both the actual speakers in the signal and templates from randomly selected speakers out of the 2,484 LibriSpeech speakers. Speaker embeddings are computed as the average of two random enrollment sentences for each speaker.

### *4.1.2. AMI meeting corpus*

We conducted fine-tuning and testing on the AMI corpus, which consists of approximately 100 h of multiple distant microphone (MDM) recordings with 3 to 5 speakers. In this paper, the VAD module takes all 8 MDM channels as inputs, while the SA-ASR module takes only the first MDM channel. We assessed three alternative segmentation methods. The first method, following [15], involves fixed-sized segmentation with a chunk size of 5 s and a 2 s hop size, followed by adjusting the start/end times to non-overlapped word boundaries as described in Section 2.4.1. The second method, following [14, 16], involves segmenting based on ground-truth silence/non-speech positions between utterances: we extracted segments for all speakers from the ground-truth speaker and utterance boundary annotations, merged overlapped segments, and discarded the resulting segments which were longer than 100 s.

The third segmentation method described in Section 3.2 uses VAD to extract the segments. We fused adjacent segments based on silence duration thresholds of 0.1, 0.3, 0.5, 0.7, and 0.9 s, resulting in datasets with different numbers of segments and different average segment lengths. Table 1 provides the statistics and average lengths for the training set under all segmentation methods.

### *4.1.3. Metrics*

We evaluate the SA-ASR system's ASR and speaker assignment performance using the word error rate (WER) and the token-level speaker error rate (SER)[1], respectively. Furthermore, we compute the speaker counting accuracy [15, 34]. For a test set with a count of $k$ speakers (with $k-1$ speaker change tokens <sc>), we calculate

$$\text{Speaker Counting Accuracy} = \frac{N_k^i}{N_k}, \tag{2}$$

where $N_k^i$ is the number of test signals where the estimate indicates $i$ speakers when the ground truth has $k$ speakers, and $N_k$ represents the number of test signals where the ground truth has $k$ speakers.

## 4.2. Model and training setup

Our experiments were implemented using the SpeechBrain toolkit [35]. For VAD, we used the CRDNN model[2] pretrained on Libriparty [36]. For SD, we use x-vectors as speaker embeddings and the Spectral Clustering [37] method. The SD and SA-ASR systems share a common speaker embedding model, ECAPA-TDNN[3], which is pretrained on the VoxCeleb1 [38] and VoxCeleb2 [39] training datasets, yielding 192-dimensional embeddings. Our text tokenizer is a SentencePiece model [40] with a vocabulary of 5,000 tokens. In SA-ASR, the Conformer-based encoder, the Transformer-based decoder, and the speaker decoder have 12, 6, and 2 layers, respectively. All multi-head attention mechanisms have 4 heads, the model dimension is set to 256, and the size of the feedforward layer is 2,048. The number of parameters of the employed CRDNN, ECAPA-TDNN, and SA-ASR models is 0.1M, 22.15M and 61.1M, respectively.

The ASR module in SA-ASR was pretrained on simulated multi-speaker Librispeech data for 360k iterations using the Adam optimizer with a learning rate of $5 \times 10^{-4}$. The ASR and speaker modules in SA-ASR were then further pretrained for 300k iterations using a learning rate of $2.5 \times 10^{-4}$. Finally, we fine-tuned the model on real AMI single distant microphone (SDM) data for 20k iterations, using the Adam optimizer with a learning rate of $3 \times 10^{-4}$. We tuned only the ASR module for the first 10k iterations and performed joint tuning of the ASR and speaker modules in the last 15k iterations. Those numbers of iterations were fixed in preliminary experiments based on the development set WER.

# 5. Discussion of results

## 5.1. Effectiveness of fine-tuning on VAD segments

In practical applications, it is necessary to divide long continuous meeting audio into smaller-sized segments. As detailed

[1]The token-level SER is calculated from sequences of estimated and ground-truth token-level speaker labels, in the same way as the WER.

[2]Available at `https://huggingface.co/speechbrain/vad-crdnn-libriparty`

[3]Available at `https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb`

Table 1: WER and SER (%) on the AMI test set using different segmentation methods for SA-ASR fine-tuning and testing. The speaker embedding templates are computed by averaging the embeddings obtained from the 3 longest annotated segments.

| Train/dev seg method | Train set statistics | | | Test (matched seg method) | | | | Test (VAD seg with 0.5 s silence threshold) | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1,2,3,4-spk mix | | | | 1-spk | | 2-spk mix | | 3-spk mix | | 1,2,3,4-spk mix | | | |
| | Sil. thresh | # Seg | Avg dur | # Seg | Avg dur | WER | SER | WER | SER | WER | SER | WER | SER | # Seg | Avg dur | WER | SER |
| Fixed-size (5 s) | - | 90,426 | 6.59 s | 10,234 | 6.63 s | 44.54 | 28.54 | 49.41 | 28.92 | 50.91 | 29.78 | 57.48 | 36.29 | 3,586 | 6.96 s | 55.91 | 34.94 |
| Ground-truth | - | 21,917 | 10.19 s | 2,504 | 8.67 s | **43.42** | **21.81** | 44.47 | 23.08 | 50.74 | 28.21 | 56.77 | 31.47 | 3,586 | 6.96 s | 54.00 | 31.02 |
| VAD | 0.1 s | 93,150 | 1.59 s | 10,605 | 1.59 s | 44.62 | 29.03 | **30.53** | 17.12 | 45.67 | 29.15 | 53.04 | 38.07 | | | 47.02 | 30.19 |
| | 0.3 s | 57,589 | 3.40 s | 6,095 | 3.71 s | 44.50 | 28.85 | 31.95 | 16.17 | 43.54 | 28.99 | 52.38 | 35.69 | | | 46.33 | 28.99 |
| | 0.5 s | 35,091 | 6.18 s | 3,586 | 6.96 s | 45.09 | 27.46 | 31.25 | 15.53 | 42.05 | 23.21 | 50.57 | 30.54 | 3,586 | 6.96 s | **45.09** | 27.46 |
| | 0.7 s | 22,027 | 10.33 s | 2,265 | 10.24 s | 45.80 | 24.82 | 32.13 | **13.51** | **41.74** | **22.17** | **49.82** | **27.00** | | | 46.13 | 27.02 |
| | 0.9 s | 14,460 | 15.65 s | 1,519 | 13.15 s | 47.08 | 24.95 | 33.05 | **13.35** | 42.52 | **22.31** | 51.39 | **27.30** | | | 46.27 | **25.04** |

Note: For all the tables, we employed the SCTK toolkit [41] to conduct significance tests, specifically the Matched Pair Sentence Segment test. We highlight in bold the best WER/SER result and the results statistically equivalent to it at a 0.05 significance level.

in Section 4.1.2, our experiments involve three alternative segmentation methods. When fine-tuning the SA-ASR system, we employed different training data segmented using these three distinct methods. To emulate real-world test conditions, the primary focus is on testing all models using VAD output segments. The results presented in Table 1 showcase the results on VAD segments with 0.5 s silence threshold, consistent with the subsequent tables. To gain a more comprehensive understanding of the model's capabilities across various speaker counts, we report the results on subsets comprising 1 to 3 speakers. For comparison, we also report the performance of each model on a matched test segmented in the same way as the training set.

Looking at the last column in Table 1, models fine-tuned on VAD segments consistently exhibit superior performance when tested on VAD segments compared to models fine-tuned on fixed-sized or ground-truth segments. The relative WER reduction can be up to 19% (from 55.91% to 45.09%), and the relative SER reduction can be as high as 28% (from 34.94% to 25.04%). However, this difference is not as pronounced when models are tested on their respective matched test sets. This underscores the significance of training on data that closely aligns with the test data type. For real-life applications relying on VAD segments, it is advisable to fine-tune on VAD segments, even when speaker segment annotations are available in the training set. It is noteworthy to highlight that, even with comparable average segment lengths (10.19 s and 10.33 s), fine-tuning on VAD segments can result in a 15% (from 54% to 46.13%) lower relative WER and a 13% (from 31.02% to 27.02%) lower relative SER compared to the model fine-tuned on ground-truth segments.

Focusing on the models fine-tuned on VAD segments in Table 1, it can be seen that the model fine-tuned on segments with 0.5 s VAD silence threshold exhibits a lower WER, which can be attributed to the matched fine-tuning and test segment lengths. However, the SER shows a different trend. Specifically, as the average fine-tuning segment length becomes longer, the corresponding SER decreases. The SER exhibits a relative reduction of 22% (from 17.12% to 13.35%), 23% (from 29.15% to 22.31%), and 28% (from 38.07% to 27.30%) for 1, 2, and 3 speakers, respectively, when transitioning from an average fine-tuning length of 1.59 s to 15.65 s. This indicates that, as the training segments become longer, SA-ASR demonstrates improved capability in assigning speaker identities.

Table 2 reports the speaker counting accuracy of the three systems. On the 1-speaker test subset, the model fine-tuned on VAD segments exhibits the highest accuracy at 94.76%, compared 80.01% for the model fine-tuned on fixed-sized segments.

Table 2: Speaker counting accuracy (%) on the AMI test set with 0.5 s VAD silence threshold.

| Train/dev seg method | # Speakers | Estimated # speakers | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | >4 |
| Fixed-size (5 s) | 1 | 80.04 | 17.28 | 2.29 | 0.32 | 0.05 |
| | 2 | 14.09 | **69.01** | 16.28 | 0.54 | 0.00 |
| | 3 | 0.73 | 38.93 | **50.04** | 9.18 | 0.91 |
| | 4 | 0.40 | 15.98 | 56.96 | 20.90 | 4.50 |
| Ground-truth | 1 | 89.00 | 9.66 | 1.22 | 0.10 | 0.00 |
| | 2 | 20.41 | 63.53 | 15.28 | 0.54 | 0.10 |
| | 3 | 2.54 | 35.39 | 48.63 | 12.15 | 0.90 |
| | 4 | 0.00 | 11.47 | 54.50 | **29.50** | 3.27 |
| VAD (0.5 s sil. thresh) | 1 | **94.76** | 4.91 | 0.26 | 0.05 | 0.00 |
| | 2 | 21.24 | 67.68 | 10.07 | 0.87 | 0.00 |
| | 3 | 2.72 | 40.29 | 47.00 | 8.89 | 0.72 |
| | 4 | 0.00 | 11.57 | 59.91 | 23.55 | 3.71 |

However, the latter system demonstrates slightly higher accuracy when the number of speakers is 2 (69.01%) or 3 (50.04%). The model fine-tuned on ground-truth segments performs better when the number of speakers is as high as 4 (29.50%). Future studies can explore how to employ different training techniques for datasets with specific numbers of speakers.

### 5.2. Extraction of speaker embedding templates from SD output

So far, we have used annotated speaker segments to extract the speaker embedding templates in the training, development, and test sets. In a real-world situation, using SD segments to extract the speaker embedding templates in the test set is desirable.

Table 3: SER (%) on the AMI test set with 0.5 s VAD silence threshold as a function of the speaker embedding templates used for fine-tuning and testing. Ref: templates extracted from annotated speaker segments. Test: templates extracted from SD output. The WER is unaffected and equal to that in Table 1 (1-spk 31.25%, 2-spk mix 42.05%, 3-spk mix 50.57%, total 45.09%).

| Train/dev | Test | 1-spk | 2-spk mix | 3-spk mix | Total |
|---|---|---|---|---|---|
| Ref | Ref | 15.53 | 23.21 | 30.54 | 27.46 |
| Ref | Test | **12.59** | **21.17** | **27.43** | **26.54** |

Table 4: SER (%) on the AMI test set with 0.5 s VAD silence threshold as a function of the VAD silence threshold and the length of candidate segments used to extract non-overlapped speaker embedding templates. The WER is unaffected and equal to that in Table 1.

| VAD segments | | Using all 2–5 s segments | | | Using all 5–10 s segments | | | Using all 6–50 s segments | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Sil. thresh** | **Avg dur** | **1-spk** | **2-spk mix** | **3-spk mix** | **1-spk** | **2-spk mix** | **3-spk mix** | **1-spk** | **2-spk mix** | **3-spk mix** |
| 0.1 s | 1.59 s | 13.05 | **20.88** | 27.55 | 60.55 † | 54.95 † | † | † | † | † |
| 0.3 s | 3.40 s | 13.22 | 21.16 | **27.24** | 13.98 | 21.80 | 27.93 | 19.09 † | 26.51 † | 32.40 † |
| 0.5 s | 6.18 s | **12.81** | **20.71** | **27.50** | 13.87 | 22.04 | 27.95 | **12.96** | **20.80** | **27.48** |
| 0.7 s | 10.33 s | 14.30 | 22.75 | 27.82 | 14.05 | 23.21 | 27.99 | **12.89** | 21.35 | 28.04 |
| 0.9 s | 15.65 s | 14.67 | 22.41 | 28.13 | **13.03** | 21.80 | 27.62 | 13.86 | 21.68 | 27.82 |

†: The VAD output segments with 0.1 s or 0.3 s silence threshold are inadequate for obtaining enough candidate segments with the specified length, resulting in unavailable or abnormal speaker assignment results.

However, this leads to statistical differences between the fine-tuning and test phases. To understand how this affects the SER, we evaluate both types of templates in the test set. The resulting performance when fine-tuning and testing on VAD output segments with 0.5 s silence threshold is presented in Table 3.

Surprisingly, while the model is fine-tuned on speaker embedding templates extracted from annotated speaker segments, extracting speaker embedding templates from SD segments during test reduces the SER by up to 20% relative (from 15.53% to 12.59%). We attribute this to the fact that human segmentation is partly inaccurate, especially when marking the boundaries of each speaker's speech segments. By contrast, SD can offer more precise speech boundaries, especially for the longest non-overlapped segments which are utilized to compute the speaker embedding templates.

Surprisingly too, these improved speaker embedding templates do not affect the WER, despite the fact that the derived token-level weighted speaker templates are improved and fed back to the ASR Decoder (see Figure 2). This suggests that this feedback mechanism proposed in [7] may not fulfil its goal of enabling the Speaker Decoder to assist the ASR Decoder.

### 5.3. Extraction of speaker embedding templates from different segment lengths

Finally, we explored the extraction of speaker embedding templates from varying segment lengths and with different numbers of candidate segments. To begin with, the impact of using VAD with 0.1 s to 0.9 s silence threshold as input segments to the SD system is analyzed in Table 4. For each SD result, we filtered candidate segments for each speaker by considering different length intervals. The mean embedding for each speaker was then calculated from all segments in that length interval. The results showcased in Table 4 reveal several good solutions with either short or long candidate segment lengths. When candidate segment lengths fall within the range of 2 to 5 s, VAD silence thresholds below 0.5 s are preferable to longer ones, with a relative SER reduction of up to 10% (12.81% compared to 14.67%). On the contrary, when using longer candidate segments ranging from 6 to 50 s, short VAD silence thresholds are inadequate and VAD silence thresholds above 0.5 s are preferable.

We further examined how different configurations, including the presence of overlapping segments, the number of candidate segments, and the length of candidate segments, impact the speaker embedding templates and the resulting SER. Table 5 illustrates the influence of these attributes using VAD segments with 0.5 s silence threshold as inputs to SD. Using all candidate segments instead of the 2 or 10 longest leads to an enhanced speaker representation irrespective of the segment length, as

Table 5: SER (%) on the AMI test set with 0.5 s VAD silence threshold as a function of the number and the length of candidate segments used to extract speaker embedding templates. The WER is unaffected and equal to that in Table 1.

| Candidate segments | | | SER | | |
|---|---|---|---|---|---|
| **Overlap** | **# seg** | **Dur** | **1-spk** | **2-spk mix** | **3-spk mix** |
| no | 2 | 2–5 s | 15.12 | 23.34 | 29.35 |
| no | 10 | 2–5 s | 14.41 | 21.96 | 28.14 |
| no | all | 2–5 s | **12.81** | **20.71** | **27.50** |
| yes | all | 2–5 s | **12.81** | **20.77** | **27.27** |
| no | 2 | 5–10 s | 13.30 | 21.98 | 29.27 |
| no | 10 | 5–10 s | 14.13 | 21.95 | 27.81 |
| no | all | 5–10 s | 13.87 | 22.04 | 27.95 |
| yes | all | 5–10 s | 13.92 | 21.92 | 27.92 |
| no | 2 | 6–50 s | 13.12 | 21.48 | 27.62 |
| no | 10 | 6–50 s | 13.33 | 21.06 | **27.27** |
| no | all | 6–50 s | **12.96** | **20.80** | **27.48** |
| yes | all | 6–50 s | 13.04 | **20.82** | **27.44** |

evidenced by a lower SER up to 15% relative (from 15.12% to 12.81%). Moreover, shorter candidate segments exhibit a greater sensitivity to the number of segments. This can be attributed to the fact that a larger number of segments enables a more robust and comprehensive representation of each speaker. The presence of overlapping segments among the candidate segments does not appear to have a significant impact. Using segment annotations, we compute the number of overlapping clips involving varying numbers of speakers. Subsequently, for each speaker count, we calculate the quantity of overlapping clips across different durations of overlap. Figure 6 presents statistical findings across the complete AMI corpus, revealing a minimal occurrence of 2, 3, and 4 overlapping speaker regions.

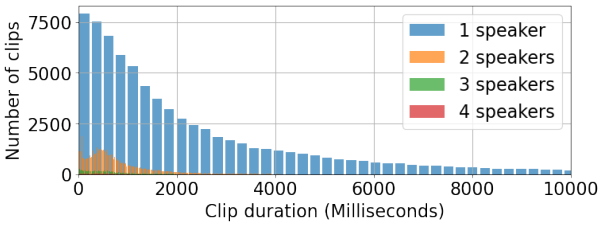


Figure 6: Histograms of speaker overlaps on the AMI corpus. Only clips shorter than 10 s are shown.

Figure 7 depicts the cosine similarity between the speaker embeddings of the candidate segments in one meeting for the three specified segment lengths. Segments in the range of 6 to

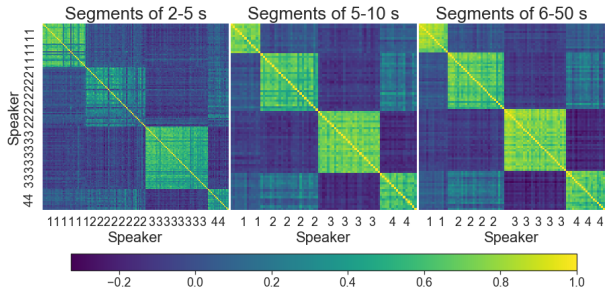


Figure 7: Similarity matrices between the speaker embeddings of the candidate segments in meeting ES2004c. With the three specified segment lengths, the number of candidate segments is 154, 81, and 96, and the resulting SER for that meeting is 19.91%, 19.66%, and 19.64%, respectively.

50 s exhibit a higher similarity compared to those in the range of 2 to 5 s, leading to a slightly lower SER (19.64% vs. 19.91%).

Finally, note that here again the various speaker embedding templates explored in Tables 4 and 5 do not affect the WER.

## 6. Conclusion

We focused on enhancing speaker assignment in SA-ASR for real-life meetings. We proposed a VAD-SD-SA-ASR pipeline and advocated for fine-tuning on VAD segments instead of fixed-size segments. We showed that this can lead to relative SER reduction up to 28%. We then explored strategies for extracting speaker embedding templates by varying the VAD and SD configurations. Our results reveal that extracting them from SD output rather than annotated speaker segments results reduces the SER by up to 20% relative. Moreover, while extracting speaker embedding templates from short or long segments can yield a similar SER, the latter results in improved speaker representation and speaker similarity across segments and is less sensitive to the number of segments averaged to obtain the template. Finally, we have found that improved speaker embedding templates do not affect the WER; this suggests that an effective feeback mechanism enabling the Speaker Decoder to assist the ASR Decoder is still to be found.

## 7. Acknowledgements

Experiments presented in this paper were carried out using (a) the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr), and (b) HPC resources from GENCI-IDRIS (Grant 2023-[AD011013881]).

## 8. References

[1] F. Yu, S. Zhang, Y. Fu, L. Xie, S. Zheng, Z. Du, W. Huang, P. Guo, Z. Yan, B. Ma, et al., "M2met: The ICASSP 2022 multi-channel multi-party meeting transcription challenge," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6167–6171.

[2] T. Yoshioka, I. Abramovski, C. Aksoylar, Z. Chen, M. David, D. Dimitriadis, Y. Gong, I. Gurvich, X. Huang, Y. Huang, et al., "Advances in online audio-visual meeting transcription," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 276–283.

[3] T. Yoshioka, D. Dimitriadis, A. Stolcke, W. Hinthorn, Z. Chen, M. Zeng, and X. Huang, "Meeting transcription using asynchronous distant microphones.," in *Interspeech*, 2019, pp. 2968–2972.

[4] S. Cornell, M. S. Wiesner, S. Watanabe, D. Raj, X. Chang, P. Garcia, Y. Masuyam, Z.-Q. Wang, S. Squartini, and S. Khudanpur, "The CHiME-7 DASR Challenge: Distant meeting transcription with multiple devices in diverse scenarios," in *7th International Workshop on Speech Processing in Everyday Environments (CHiME)*, 2023, pp. 1–6.

[5] X. Chang, W. Zhang, Y. Qian, J. Le Roux, and S. Watanabe, "End-to-end multi-speaker speech recognition with Transformer," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6134–6138.

[6] H. Seki, T. Hori, S. Watanabe, J. Le Roux, and J. R. Hershey, "A purely end-to-end system for multi-speaker speech recognition," in *56th Annual Meeting of the ACL (Volume 1: Long Papers)*, 2018, pp. 2620–2630.

[7] N. Kanda, G. Ye, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "End-to-end speaker-attributed ASR with Transformer," in *Interspeech*, 2021, pp. 4413–4417.

[8] F. Yu, Z. Du, S. Zhang, Y. Lin, and L. Xie, "A comparative study on speaker-attributed automatic speech recognition in multi-party meetings," in *Interspeech*, 2022, pp. 560–564.

[9] X. Chang, N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Hypothesis stitcher for end-to-end speaker-attributed ASR on long-form multi-talker recordings," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6763–6767.

[10] N. Kanda, X. Chang, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Investigation of end-to-end speaker-attributed ASR for continuous multi-talker recordings," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 809–816.

[11] N. Kanda, X. Xiao, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Transcribe-to-diarize: Neural speaker diarization for unlimited number of speakers using end-to-end speaker-attributed ASR," in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 8082–8086.

[12] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, "Streaming speaker-attributed ASR with token-level speaker embeddings," in *Interspeech*, 2022, pp. 521–525.

[13] P. Guo, X. Chang, S. Watanabe, and L. Xie, "Multi-speaker ASR combining non-autoregressive Conformer CTC and conditional speaker chain," in *Interspeech*, 2021, pp. 3720–3724.

[14] N. Kanda, G. Ye, Y. Wu, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Large-scale pre-training of end-to-end multi-talker ASR for meeting transcription with single distant microphone," in *Interspeech*, 2021, pp. 3430–3434.

[15] C. Cui, I. Sheikh, M. Sadeghi, and E. Vincent, "End-to-end multichannel speaker-attributed ASR: Speaker guided

decoder and input feature analysis," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[16] M. Yang, N. Kanda, X. Wang, J. Wu, S. Sivasankaran, Z. Chen, J. Li, and T. Yoshioka, "Simulating realistic speech overlaps improves multi-talker ASR," in *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[17] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, et al., "The AMI meeting corpus: A preannouncement," in *International Workshop on Machine Learning for Multimodal Interaction*, 2005, pp. 28–39.

[18] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, "Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario," in *Interspeech*, 2020, pp. 274–278.

[19] X. Yang, B. Tan, J. Ding, J. Zhang, and J. Gong, "Comparative study on voice activity detection algorithm," in *2010 International Conference on Electrical and Control Engineering*, 2010, pp. 599–602.

[20] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, pp. 101317, 2022.

[21] X. Xiao, N. Kanda, Z. Chen, T. Zhou, T. Yoshioka, S. Chen, Y. Zhao, G. Liu, Y. Wu, J. Wu, et al., "Microsoft speaker diarization system for the VoxCeleb speaker recognition challenge 2020," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5824–5828.

[22] 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)*, 2018, pp. 5329–5333.

[23] M. Asif, M. T. Vinodbhai, S. Mishra, A. Gupta, and U. S. Tiwary, "Emotion recognition in VAD space during emotional events using CNN-GRU hybrid model on EEG signals," in *International Conference on Intelligent Human Computer Interaction*. Springer, 2022, pp. 75–84.

[24] J. Lin, K.-F. Un, W.-H. Yu, R. P. Martins, and P.-I. Mak, "A 47-nW voice activity detector (VAD) featuring a short-time CNN feature extractor and an RNN-Based classifier with a non-volatile CAP-ROM," *IEEE Journal of Solid-State Circuits*, vol. 58, pp. 3020–3029, 2023.

[25] J. Kim, J. Kim, S. Lee, J. Park, and M. Hahn, "Vowel based voice activity detection with LSTM recurrent neural network," in *Proceedings of the 8th International Conference on Signal Processing Systems*, 2016, pp. 134–137.

[26] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4580–4584.

[27] Y. Xiang, T. Tang, T. Su, C. Brach, L. Liu, S. S. Mao, and M. Geimer, "Fast CRDNN: Towards on site training of mobile construction machines," *IEEE Access*, vol. 9, pp. 124253–124267, 2021.

[28] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[29] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 4052–4056.

[30] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Interspeech*, 2020, pp. 3830–3834.

[31] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," in *Interspeech*, 2020, pp. 2797–2801.

[32] 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)*, 2015, pp. 5206–5210.

[33] D. Diaz-Guerra, A. Miguel, and J. R. Beltran, "gpuRIR: A Python library for room impulse response simulation with GPU acceleration," *Multimedia Tools and Applications*, vol. 80, pp. 5653–5671, 2021.

[34] N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers," in *Interspeech*, 2020, pp. 36–40.

[35] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[36] M. Ravanelli, "Libriparty," `https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriParty/generate_dataset`, 2023, GitHub repository.

[37] H. Ning, M. Liu, H. Tang, and T. S. Huang, "A spectral clustering approach to speaker diarization," in *Interspeech*, 2006, pp. 2178–2181.

[38] J. S. Chung, A. Nagrani, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "VoxSRC 2019: The first VoxCeleb speaker recognition challenge," *arXiv preprint arXiv:1912.02522*, 2019.

[39] A. Nagrani, J. Son Chung, J. Huh, A. Brown, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "VoxSRC 2020: The second VoxCeleb speaker recognition challenge," *arXiv e-prints*, pp. arXiv–2012, 2020.

[40] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 66–71.

[41] NIST, "SCTK," `https://github.com/usnistgov/SCTK.git`, 2024, GitHub repository.