MEETING RECOGNITION WITH CONTINUOUS SPEECH SEPARATION AND TRANSCRIPTION-SUPPORTED DIARIZATION

Thilo von Neumann[1], *Christoph Boeddeker*[1], *Tobias Cord-Landwehr*[1], *Marc Delcroix*[2], *Reinhold Haeb-Umbach*[1]

[1]Paderborn University [2]NTT corporation, Japan

ABSTRACT

We propose a modular pipeline for the single-channel separation, recognition, and diarization of meeting-style recordings and evaluate it on the Libri-CSS dataset. Using a Continuous Speech Separation (CSS) system with a TF-GridNet separation architecture, followed by a speaker-agnostic speech recognizer, we achieve state-of-the-art recognition performance in terms of Optimal Reference Combination Word Error Rate (ORC WER). Then, a d-vector-based diarization module is employed to extract speaker embeddings from the enhanced signals and to assign the CSS outputs to the correct speaker. Here, we propose a syntactically informed diarization using sentence- and word-level boundaries of the ASR module to support speaker turn detection. This results in a state-of-the-art Concatenated minimum-Permutation Word Error Rate (cpWER) for the full meeting recognition pipeline.

Index Terms— Speech Separation, Speech Recognition, Diarization, Libri-CSS, Meeting Separation

1. INTRODUCTION

Automatic meeting transcription aims at answering the question “who spoke what and when” [1]. To do so, the tasks of speech separation, diarization and recognition have to be solved. Integrated and modular architectures have been proposed for meeting transcription.

Integrated solutions, such as the Multi-turn Recurrent Neural Network Transducer (MT-RNN-T) [2] and the Token-Level Serialized Output Training (T-SOT) [3] use a transformer transducer model to recognize the speech of all speakers in a multi-talker recording without an explicit separation module. Speaker assignment is then performed on the segmented recognition output. The attractiveness of those integrated approaches is the training under a common objective function, which potentially delivers superior results.

In contrast, modular systems combine modules that explicitly perform speech separation, diarization, and recognition. They have the advantage that the individual modules can be optimized independently and implemented using different tools. However, it is unclear which processing order of the aforementioned three tasks is the most appropriate.

In [1], three variants of the modular approach were introduced. The “CHiME-6 pipeline” first conducts diarization, followed by separation and then Automatic Speech Recognition (ASR) [4]. This pipeline benefits from the wealth of literature on diarization, where in recent years, effective solutions even for partially overlapping speech have been found [5, 6]. The diarization output can guide the source separation [7], or the speaker information can be used to perform target speaker extraction, e.g., using SpeakerBeam [8, 9]. However, poor diarization can negatively impact the source separation and, thus, recognition accuracy.

Therefore, Raj et al. proposed to swap the order of diarization and separation to conduct separation before diarization [1]. This framework employs Continuous Speech Separation (CSS), which is a source separation scheme that maps multi-talker speech with an arbitrary number of speakers to a fixed number of output channels (typically two) such that there is no speaker overlap on any of the output channels [10]. We refer thus to this pipeline as CSS followed by diarization and ASR (CSS-DA). By performing separation first, we can greatly simplify the diarization system since it does not need to handle overlap. For example, (conventional) clustering-based diarization can be sufficient if all speech overlaps are removed. The diarization performance depends, however, on the quality of the source separation. Furthermore, segmentation remains an issue as it is not always possible to reliably detect speaker changes just from the speech activity, especially when speakers take turns quickly.

The third processing order first carries out speech recognition on the CSS output followed by diarization on the ASR output, which we call CSS-AD to emphasize the processing order.[1] It enjoys the same advantages as the CSS-DA pipeline. In addition, the diarization may benefit from word boundary information to, e.g., reduce false alarms if a robust recognizer is employed.

Although the third option appears appealing since diarization does not need to care for overlap and can exploit information from the ASR system, only a few studies have followed this pipeline [11]. In [12], timestamps from ASR were used for diarization. However, they did not integrate it into a full CSS pipeline since they did not use separation and thus could not handle overlapped speech. The full CSS-AD pipeline has been implemented in [11], but it heavily relied on multi-channel and multi-modal inputs to cope with imperfect separation or artifacts introduced by the separation stage.

In this paper, we investigate the potential of the CSS-AD pipeline for processing single-channel meeting recordings. First, we implement a single-channel CSS using TF-GridNet [13, 14]. Although TF-GridNet has been shown to achieve a state-of-the-art separation performance for fully overlapped speech on reverberant mixtures [13, 14], it has not been investigated with partially overlapped speech and CSS-style separation of long recordings. We demonstrate that it indeed has the potential for CSS, but diarization after separation remains challenging because speech activity is not enough to detect speaker turns. We introduce a novel segmentation approach that exploits syntactic information, such as word and sentence boundaries, from the ASR system. In particular, since speaker embeddings benefit from long speech segments, we propose to segment the separated signals at sentence boundaries predicted by the recognizer. Due to imperfect separation, sentences can contain

[1] It was called “CSS pipeline” in [1].

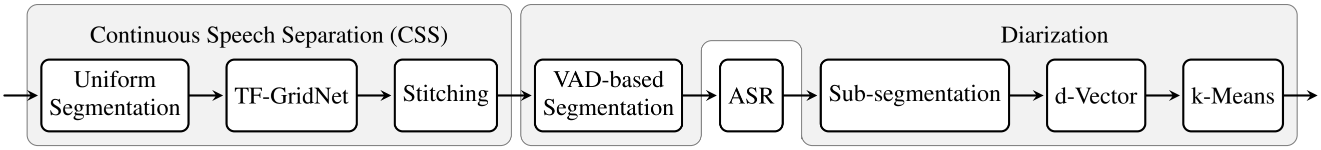

Fig. 1: Proposed processing pipeline for meeting transcription.

speech from multiple speakers. We thus refine these segments by detecting speaker changes by word-level speaker embeddings.

We carry out experiments on the well-known Libri-CSS meeting recognition task [10]. We confirm that the TF-GridNet architecture is well-suited for CSS and achieves state-of-the-art Optimal Reference Combination WER (ORC WER) on the task. In addition, using our proposed transcription-supported diarization, we also achieve state-of-the-art Concatenated Minimum Permutation WER (cpWER), surpassing prior work by a relative WER improvement of 20 %. These strong results suggest the potential for the modular CSS-AD pipeline, which performs diarization after separation and ASR, and may encourage more research in that direction.

2. CONTINUOUS SPEECH SEPARATION

A block diagram of the proposed CSS-AD processing pipeline is shown in Figure 1. It consists of CSS, followed by Voice Activity Detection (VAD)-based segmentation, ASR, and finally clustering-based diarization.

2.1. TF-GridNet

We use TF-GridNet [13, 14] to achieve state-of-the-art separation performance. The TF-GridNet works in the STFT domain and uses spectral mapping instead of masking. It alternates between modeling cross-time dependencies for every frequency, cross-frequency dependencies for each time frame, and global dependencies. Cross-time and cross-frequency dependencies are modeled with Long Short Term Memorys (LSTMs) and global dependencies are modeled with attention.

TF-GridNet has been applied to fully overlapped speech separation and speech enhancement [13], but not to conversational speech, where it is important to separate speech in overlapping regions and not negatively affect single-speaker regions.

To this end, we trained the separator on fully overlapped two-speaker mixtures and single-speaker data that we generated with the MMS-MSG tool [15]. We use the configuration suggested in [13] for reverberant speech separation, i.e., $B = 4$ blocks and a hidden size of $H = 192$.

2.2. Continuous processing with stitching

Extending TF-GridNet from fully overlapping utterances to long conversations that exceed minutes and contain many speakers can be done with the Continuous Speech Separation (CSS) concept proposed in [16], which is also frequently referred to as stitching.

The two-speaker separation system (here TF-GridNet) is applied independently to short, overlapping segments of the recording, here 4 s long with 2 s overlap. This results in an inter-segment speaker-permutation problem. Neighboring segments thus have to be aligned after separation to obtain continuous and consistent output signals from concatenation. The permutation is found based on an MSE criterion computed on the overlapping parts of two adjacent segments.

With this approach, a two-speaker separator can be applied to an arbitrarily long recording containing arbitrarily many speakers. However, given only two output channels, it is constrained to at most two speakers active within a single segment. The outputs of this model are two overlap-free streams, with each stream containing the speech of multiple speakers. Therefore, diarization is not yet solved at this stage.

3. DIARIZATION AND ASR

For diarization, we apply a modular clustering-based diarization pipeline composed of VAD, embedding estimation, and k-Means clustering. This pipeline is then extended with a sub-segmentation module that allows the detection of speaker turns in the VAD segments in order to attribute these regions to the correct speaker. The sub-segmentation is supported by timing information given by the ASR engine. Fig. 2 illustrates these sub-segmentation approaches.

3.1. VAD-based Segmentation

An energy-based Voice Activity Detection (VAD), as shown in Fig. 2-(a), is employed to detect utterance boundaries in the separated signals. The framewise energy is computed in the time-frequency domain and thresholding is applied to detect regions of active speech. Then, the estimate is smoothed over time using morphological closing (dilation followed by erosion). The parameters of the closing operation are chosen such that speech activity is over-estimated by 0.4 s, which improves ASR performance [17].

3.2. Automatic Speech Recognition

We employed Whisper [18] for Automatic Speech Recognition (ASR) because its output is accompanied by word boundary timestamps. This information is used to support the segmentation, which is an essential component of the diarization.

3.3. Sub-segmentation

A VAD-based segmentation is unable to distinguish between speakers and is, therefore, unable to detect speaker changes or quick speaker turns, resulting in activity segments that may contain more than a single speaker. To mitigate this, we sub-segment each continuous speech region found by the initial VAD. For the sub-segmentation, we explore different options.

3.3.1. Uniform segmentation

Uniform segmentation sub-segments the audio into equally sized pieces, as shown in Fig. 2-(b) and is commonly employed in diarization [19]. The choice of the segment length is a trade-off between temporal resolution (smaller segments give better resolution) and the speaker embedding quality (longer segments result in better embeddings while, here, also increasing the probability for mixed-speaker segments). Typically, speaker embeddings are considered to become

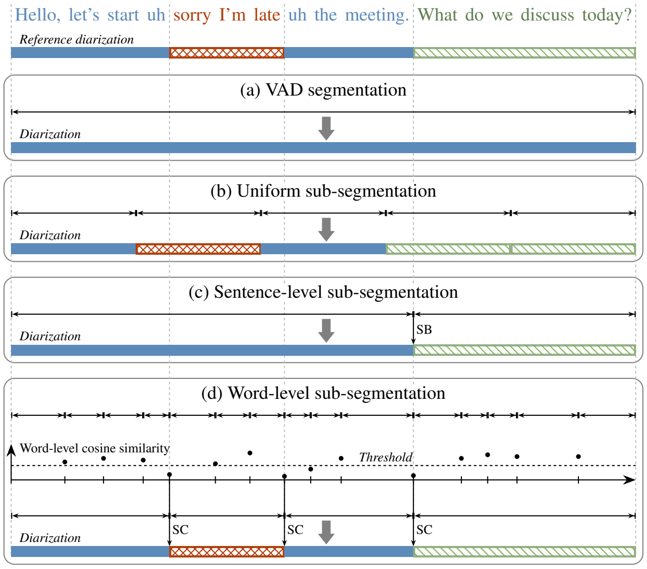

Fig. 2: Comparison of the different segmentation schemes. Here, colors represent speakers. SB and SC stand for sentence boundary and speaker change.

uninformative if computed on segments shorter than 2 s but stop to improve in quality if the length exceeds 4 s of audio [20]. We here perform uniform segmentation after ASR and move words that happen to lie on a sub-segment boundary to the sub-segment with which they most overlap.

3.3.2. Sentence-level segmentation

The uniform segmentation is not guaranteed to segment in positions that are beneficial for speaker clustering. With the assumption that speaker changes are more likely at sentence boundaries than within a sentence, we propose to sub-segment at positions where the ASR system predicts an end-of-sentence tag, e.g., a full stop. This improves the quality of the speaker embeddings extracted from the sub-segments because the detected segments are more likely to contain only a single speaker, while at the same time maximizing the segment length for more robust speaker embeddings. Fig. 2-(c) shows an example of sentence-level sub-segmentation.

3.3.3. Word-level d-vectors for segmentation

Although speaker changes likely happen at the end of sentences, they may also happen in the middle of a sentence, or the ASR model may fail to predict the end-of-sentence tag. We implement the following procedure to find those cases.

The start and end times of words as predicted by the ASR system are used to estimate word-level speaker embeddings, here d-vectors (see Section 3.4), and the positions between words are treated as candidates for speaker-change-points. For each speaker-change-point candidate, we compute a score as the cosine similarity between the average of the six preceding word-level embeddings and the six following word-level embeddings. A candidate with a lower score is more likely to be a true speaker-change-point. We use the average over a certain context because the d-vectors are noisy due to the usually short word durations. We hypothesize a speaker change when a candidate's score is the lowest within a context of six words to the left and right and at the same time below a threshold of 0.2. Note that the word-level d-vectors are only used for sub-segment estimation and not for the follow-up clustering, which is described in the next section. Fig. 2-(d) shows an example of the word-level sub-segmentation.

When sentence-level segmentation is combined with word-level segmentation, the segments are first split using the sentence boundaries. The resulting segments are then subdivided using the word-level d-vector approach.

3.4. Speaker embedding Clustering

We use the speaker embedding extractor from [21] to compute d-vectors. This model is based on the ResNet-34 architecture [22], and is trained on VoxCeleb [23] segments, where noise from the MUSAN database [24] was added and signals were reverberated with simulated room impulse responses for data augmentation. The d-vectors are extracted for each sub-segment and normalized to unit length. Then, they are clustered with k-Means++ [25] to obtain the speaker labels for each sub-segment. The diarization results consist of the combination of the sub-segment boundaries and the associated speaker labels.

4. EXPERIMENTS

4.1. Data

For the experiments, we used Libri-CSS [10], which is a commonly used database for the evaluation of meeting transcription systems. It consists of 10 hours of meeting data that is simulated based on the LibriSpeech `test_clean` set [26] with overlap ratios between 0 % to 40 %. These meetings were then re-recorded in a conference room with distributed loudspeakers in order to mimic real meetings.

4.2. ASR systems

For the recognition, we used Whisper [18] in the "large-v2" configuration. This recognizer ignores speaker changes and provides continuous transcriptions after turn-takings. Additionally, it delivers punctuation symbols, from which the end of sentences can be detected, and the time stamps of the boundaries of each estimated word. Both properties make it well suited for our proposed transcription-supported sub-segmentation.

While Whisper is a strong general ASR model, it falls behind compared to in-domain trained LibriSpeech models: Whisper "large-v2" achieves 5.2% Word Error Rate (WER) on `dev_other` while [27] reports 3.7% on huggingface[2]. To push the pipeline's performance, we hence use the conformer-based ASR system from ESPnet that utilizes WavLM features [28] as a second speech recognizer after diarization. Note that this model does not deliver word boundaries and can thus not be used for transcription-supported diarization.

We use the off-the-shelf ASR systems without fine-tuning them to the artifacts produced by the separation.

4.3. Metrics

We evaluate our meeting transcription pipeline using Optimal Reference Combination WER (ORC WER) [2], Concatenated Minimum Permutation WER (cpWER) [4] using the MeetEval[3]

[2] `https://huggingface.co/espnet/simpleoier_librispeech_asr_train_asr_conformer7_wavlm_large_raw_en_bpe5000_sp`

[3] `https://github.com/fgnt/meeteval`

Table 1: ORC WER on Libri-CSS. $^+$: cpWER with oracle speaker label assignment. *: asclite WER.

Model	ASR	ORC WER
No separation	Whisper	26.5
TF-GridNet	Whisper	6.8
TF-GridNet	ESPnet	6.4
Clean signals	Whisper	3.5
MT-RNN-T [2]	Integrated	23.6
t-SOT TT [3]	Integrated	7.6*
TS-SEP [17]	ESPnet	5.8$^+$

toolkit, and Diarization Error Rate (DER) [29] computed with the `md-eval-22.pl` script without a collar.

The ORC WER assigns each reference utterance to one system output stream such that the WER is minimized. It ignores speaker label errors. In contrast, the cpWER takes the assigned speaker labels into account and evaluates the WER for each speaker. Thus, it is the combination of both metrics that describes the ASR performance and the overall capability of the pipeline.

4.4. Evaluation without diarization

First, we skip the diarization part and evaluate the pipeline without sub-segmentation and clustering in Table 1. Without the speech separation front-end and using only VAD, the Whisper ASR yields an ORC WER of 26.5 %. By performing separation with TF-GridNet in the CSS pipeline, we obtained an ORC WER of 6.8 %, which already beats any ORC WER we found in the literature. With ESPnet and the WavLM features, we got a further improvement of 0.4 %, while Whisper achieved 3.5 % when applied to the clean sources.

Table 1 also contains results from literature. While [2] reported an ORC WER achieved with MT-RNN-T, the other two cited numbers cannot directly be compared because they did not compute ORC WER, but similar quantities, from which it can only be concluded that the ORC WER is in the same ballpark.

4.5. Diarization Evaluation

Table 2 reports cpWERs for the different sub-segmentation schemes discussed in Section 3.3. Using the segments found by the VAD and performing clustering on the d-vectors computed from those segments gave a rather poor performance of 14.8 % cpWER. We evaluated the oracle clustering performance to determine whether this was caused by bad segmentation or bad clustering. The oracle clustering attributes the speaker label that minimizes the cpWER, as done in [17]. Oracle clustering only slightly improved cpWER to 13.9 %, which indicates that the VAD segmentation is the major problem. With uniform segmentation with 4 s and with 2 s we achieved 9.8 % and 12.4 % cpWER, respectively. The comparison with the results achieved with oracle clustering shows that shorter segments are, in principle, better if only reliable d-vectors could be obtained. Our proposed sub-segmentation schemes based on syntactic information significantly improved the cpWER, and using first sentence-level information, and then the word-level d-vectors pushed the performance down to 7.2 % cpWER.

Table 3 compares the DER and cpWER of our proposed method with those of prior works. We selected hyperparameters to optimize recognition performance in terms of WER which does not translate to a good diarization performance. In particular, the morphological closing used for smoothing in the VAD overestimates speech activity for good WER results [17], hence our DER cannot compete with the DER parameters from [17]. Nevertheless, we obtained a solid DER of 9.4 % on Libri-CSS.

Table 2: cpWER on Libri-CSS using TF-GridNet, energy-based VAD, Whisper and different sub-segmentations.

Sub-segmentation	Clustering	
	k-Means	Oracle
–	14.8	13.9
uniform 4s	9.8	7.5
uniform 2s	12.4	6.8
sentence	7.9	7.4
word	7.6	7.1
sentence + word	7.2	6.6

Table 3: Single channel Libri-CSS literature comparison

System	DER	cpWER
SC [30]	11.3	–
RPN [31]	9.5	–
SC + TS-VAD + ESPnet [17]	5.7	9.3
SC + TS-SEP + ESPnet [17]	15.7	7.8
Transcribe-to-Diarize [32]	7.9	11.6
Proposed	9.4	7.2
Proposed + ESPnet	9.4	6.2

Our proposed pipeline outperforms prior works in terms of cpWER. While the Whisper ASR is necessary for the sub-segmentation, a second ASR pass can be applied to the sub-segments after the full pipeline. We used the ESPnet ASR for the second ASR pass since it performs better on LibriSpeech data. With that second ASR pass, we obtained a further improvement of 1 %, which outperforms the best single-channel cpWER on Libri-CSS by a 20 % relative cpWER improvement.

5. CONCLUSIONS

We presented a modular pipeline for transcribing a meeting that consists of a source separation, a recognition, and a diarization module that achieves state-of-the-art cpWER performance on Libri-CSS. Using TF-GridNet as a separator in the CSS module delivers excellent separation results. Since a VAD is unable to detect fast speaker turns, we employed sentence boundary predictions by the ASR for diarization. But even the ASR engine may miss sentence boundaries. The proposed sub-segmentation based on word boundary information to fix this issue further improved the performance, and combining sentence and word information delivered a new state-of-the-art single-channel cpWER of 7.2 % and 6.2 % with Whisper and ESPnet, respectively. Thus, to answer the question posed in the introduction, a modular pipeline with source separation followed by ASR and followed by diarization is highly competitive, even with single-channel input.

6. ACKNOWLEDGEMENTS

Computational Resources were provided by BMBF/NHR/PC2. Christoph Boeddeker was funded by DFG, project no. 448568305.

7. REFERENCES

[1] D. Raj, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M. He, S. Watanabe, J. Du, et al., “Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis,” in *IEEE SLT*, 2021.

[2] I. Sklyar, A. Piunova, X. Zheng, and Y. Liu, “Multi-turn RNN-T for streaming recognition of multi-party speech,” in *IEEE ICASSP*, 2022.

[3] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, “Streaming multi-talker asr with token-level serialized output training,” *arXiv preprint arXiv:2202.00842*, 2022.

[4] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, et al., “CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings,” *arXiv preprint arXiv:2004.09249*, 2020.

[5] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, et al., “Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario,” in *Interspeech*, 2020.

[6] H. Bredin and A. Laurent, “End-To-End Speaker Segmentation for Overlap-Aware Resegmentation,” in *Interspeech*, 2021.

[7] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, “Front-end processing for the chime-5 dinner party scenario,” in *CHiME 2018 Workshop on Speech Processing in Everyday Environments*, 2018.

[8] M. Delcroix, K. Zmolikova, T. Ochiai, K. Kinoshita, and T. Nakatani, “Speaker activity driven neural speech extraction,” in *IEEE ICASSP*, 2021.

[9] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, “Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures,” *IEEE J-STSP*, vol. 13, no. 4, 2019.

[10] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, “Continuous speech separation: Dataset and analysis,” in *IEEE ICASSP*, 2020.

[11] 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 *IEEE ASRU*, 2019.

[12] D. Dimitriadis and P. Fousek, “Developing On-Line Speaker Diarization System,” in *Interspeech 2017*, 2017.

[13] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, “TF-GridNet: Integrating full-and sub-band modeling for speech separation,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[14] Z.-Q. Wang, S. Cornell, S. Choi, Y. Lee, B.-Y. Kim, and S. Watanabe, “TF-GridNet: Making time-frequency domain models great again for monaural speaker separation,” in *IEEE ICASSP*, 2023.

[15] T. Cord-Landwehr, T. Von Neumann, C. Boeddeker, and R. Haeb-Umbach, “MMS-MSG: A multi-purpose multi-speaker mixture signal generator,” in *IEEE IWAENC*, 2022.

[16] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, “Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks,” in *Interspeech*, 2018.

[17] C. Boeddeker, A. S. Subramanian, G. Wichern, R. Haeb-Umbach, and J. L. Roux, “TS-SEP: Joint diarization and separation conditioned on estimated speaker embeddings,” *arXiv preprint arXiv:2303.03849*, 2023.

[18] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International Conference on Machine Learning*. PMLR, 2023.

[19] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, “Speaker diarization using deep neural network embeddings,” in *IEEE ICASSP*, 2017.

[20] T. Zhou, Y. Zhao, and J. Wu, “Resnext and res2net structures for speaker verification,” in *IEEE SLT*, 2021.

[21] T. Cord-Landwehr, C. Boeddeker, C. Zorilă, R. Doddipatla, and R. Haeb-Umbach, “Frame-wise and overlap-robust speaker embeddings for meeting diarization,” in *IEEE ICASSP*, 2023.

[22] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, “Deep speaker: an end-to-end neural speaker embedding system,” *arXiv preprint arXiv:1705.02304*, 2017.

[23] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech & Language*, vol. 60, 2020.

[24] D. Snyder, G. Chen, and D. Povey, “MUSAN: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.

[25] D. Arthur and S. Vassilvitskii, “K-means++ the advantages of careful seeding,” in *Proc. of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *IEEE ICASSP*, 2015.

[27] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe, “End-to-end integration of speech recognition, speech enhancement, and self-supervised learning representation,” *arXiv preprint arXiv:2204.00540*, 2022.

[28] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE J-STSP*, vol. 16, no. 6, 2022.

[29] O. Sadjadi, C. Greenberg, E. Singer, L. Mason, and D. Reynolds, “NIST 2021 Speaker Recognition Evaluation Plan,” 2021-07-12 04:07:00 2021.

[30] D. Raj, Z. Huang, and S. Khudanpur, “Multi-class spectral clustering with overlaps for speaker diarization,” in *IEEE SLT*, 2021.

[31] Z. Huang, M. Delcroix, L. P. Garcia, S. Watanabe, D. Raj, and S. Khudanpur, “Joint speaker diarization and speech recognition based on region proposal networks,” *Computer Speech & Language*, vol. 72, 2022.

[32] 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 *IEEE ICASSP*, 2022.