Interspeech 2024
1-5 September 2024, Kos, Greece

SCDNet: Self-supervised Learning Feature based Speaker Change Detection

*Yue Li, Xinsheng Wang, Li Zhang, Lei Xie**

Audio, Speech and Language Processing Group (ASLP@NPU), School of Software,
Northwestern Polytechnical University, Xi'an, China

yueli77@mail.nwpu.edu.cn, lxie@nwpu.edu.cn

Abstract

Speaker Change Detection (SCD) is to identify boundaries among speakers in a conversation. Motivated by the success of fine-tuning wav2vec 2.0 models for the SCD task, a further investigation of self-supervised learning (SSL) features for SCD is conducted in this work. Specifically, an SCD model, named SCDNet, is proposed. With this model, various state-of-the-art SSL models, including Hubert, wav2vec 2.0, and WavLm are investigated. To discern the most potent layer of SSL models for SCD, a learnable weighting method is employed to analyze the effectiveness of intermediate representations. Additionally, a fine-tuning-based approach is also implemented to further compare the characteristics of SSL models in the SCD task. Furthermore, a contrastive learning method is proposed to mitigate the overfitting tendencies in the training of both the fine-tuning-based method and SCDNet. Experiments showcase the superiority of WavLm in the SCD task and also demonstrate the good design of SCDNet.

Index Terms: speaker change detection, self-supervised models, contrastive learning

1. Introduction

Speaker Diarization (SD), a pivotal method in speech processing, aims to answer the question of 'who speaks when' in scenarios involving multiple speakers [1]. In contrast, Speaker Change Detection (SCD) is to find the speaker turn points in the conversation [2], and thus it can be regarded as a subtask of SD [3], and also with broad applications, e.g., enhancing Automatic Speech Recognition (ASR) accuracy [4] and syncopate captioning [5].

The metric-based approach is a common early method for the SCD task, wherein speaker change points are identified through the comparison of distributions between two consecutive speech windows [6]. Following the emergence of i-vector [7] and DNN-based embeddings [8], uniform segmentation schemes have gained popularity as effective methods [9]. In this approach, the target audio undergoes segmentation into a series of segments with a constant window length and overlap length. Subsequently, speech embeddings from various segments are compared to determine if the speaker has changed. However, due to the fixed window length, a trade-off is inevitable between the efficacy of speech embedding and the accuracy of boundary detection.

To overcome the limitations of segment-based methods, various works have endeavored to predict speaker change points at the frame level through neural networks [10, 1, 11]. In these approaches, the model is generally trained with ground-truth SCD labels to perform a binary classification task. To be specific, in [10], using LSTM as the backbone, the optimizing target is to minimize the distance between the predicted probability signal and linear fuzzy labeling signal.

In addition to label-based methods for frame-level SCD, several works have explored leveraging text transcription for word-level speaker change detection through ASR techniques [12, 13]. For example, in [12], the transcription used to train an ASR model is enhanced by incorporating a distinct token designed to denote speaker turns. Then the augmented transcription is used to train an ASR model that predicts not only regular text tokens but also special speaker turn tokens. While this approach alleviates the necessity for boundary annotations, using the textual transcription can be more intricate, especially in a dialogue scenario characterized by frequent interruptions and insertions, and the prevalence of intonation markers. Additionally, because the predicted boundaries in this method operate at the word level, the precision of boundary predictions may not be as high as those based on frame-level predictions.

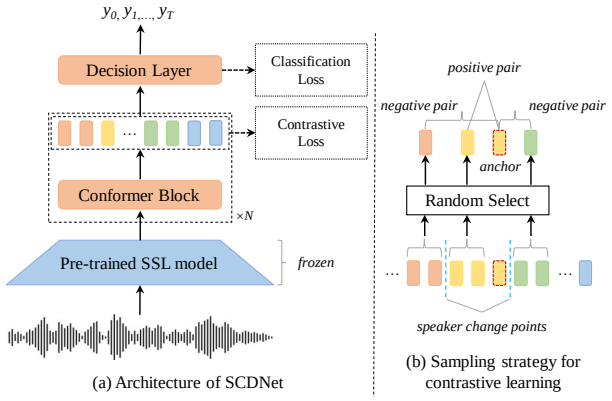
Most recently, Kunešová and Zají [2] explored the effectiveness of one of the most popular SSL models, wav2vec 2.0 [14], on the SCD task. In their research, the pretrained wav2vec 2.0 is fine-tuned in an end-to-end way involving multi-tasks, i.e., SCD, Overlapping Speech Detection (OSD), and Voice Activity Detection (VAD). This wav2vec 2.0 and multitask-based method showcases the remarkable performance, achieving a state-of-the-art (SOTA) level in the SCD task. Inspired by this research, we are conducting a further investigation into SSL-based end-to-end training methods for SCD.

On the one hand, due to the typically large number of parameters in SSL models, directly fine-tuning them requires a certain threshold of data and computational resources. On the other hand, despite efforts by Kunešová and Zají [2] to enhance SCD performance through multitasking, e.g., OSD and VAD, all these tasks are frame-level binary classification tasks, which pose a risk of overfitting when training complex models due to the simplistic learning paradigm. Additionally, besides wav2vec 2.0, other SSL models such as Hubert [15] and WavLm [16] have also gained significant attention in various downstream tasks, such as Hubert-based speech recognition [17] and WavLm-based speech synthesis [18]. However, the performance of these models in SCD has not been explored.

To tackle those issues, we propose an innovative end-to-end SCD model, referred to as SCDNet, based on the Conformer architecture [19]. SCDNet leverages off-the-shelf features as inputs and undergoes end-to-end training to accomplish the SCD task. Additionally, we propose a contrastive learning method for training SCD-oriented models to address the overfitting tendency associated with the frame-level binary classification task. Furthermore, we explore the performance of various SSL features via both SCDNet and fine-tuning-based methods.

* Corresponding author.

10.21437/Interspeech.2024-752



(a) Architecture of SCDNet

(b) Sampling strategy for contrastive learning

Figure 1: *The architecture of the SCDNet (left) and sampling strategy for contrastive learning (right).*

2. Approach

SCDNet is a Conformer-based model to achieve the frame-level binary classification with speech representation as input. In addition to the classification loss, a contrastive loss is proposed to alleviate the overfitting tendency caused by the simplistic binary learning way. This contrastive loss is also used to fine-tune the pre-trained SSL models for the SCD task.

2.1. Problem Formulation

A speaker change point is defined as the point indicating the initiation or conclusion of an individual's speech, regardless of the presence or absence of other speakers. Therefore, the consideration extends beyond transitions between two speakers to encompass voice activity boundaries. Following [2], SCD is treated here as a frame-level classification task. Given a speech feature sequence $X = \{x_0, x_1, ..., x_T\}$ and the corresponding label sequence $Y = \{y_0, y_1, ..., y_T\}$ where T denotes the total number of frames, and $y_i \subseteq \{0, 1\}$. For a model f with learnable parameters θ, the training target of SCD is formulated as:

$$f_\theta = \arg\max_\theta P(Y; X, \theta) \tag{1}$$

2.2. SCDNet

As depicted in Figure 1, the proposed SCDNet primarily comprises three components: the pre-trained SSL model, the Conformer Blocks, and the Decision Layer. During the inference process, the input audio is represented by features extracted from the pre-trained SSL model. Subsequently, these features pass through N-layer Conformer Blocks before producing the final boundary labels through the Decision Layer.

As a frame-level binary classification task, the classification loss, e.g., cross-entropy loss or distance-based loss, is the typical loss function for the training of SCD-related models. However, relying solely on classification loss for training can be challenging due to the limited information provided by binary labels, making it susceptible to overfitting. To address this challenge, a contrastive learning method is proposed for training SCDNet associated with classification loss.

The classification loss is the basic loss function for the binary classification task. Considering the potential errors introduced by manual labeling, the boundaries annotated by humans may exhibit a shift from the actual boundaries. Therefore, following [2], instead of using the original hard label, i.e., 0 or 1, a fuzzy labeling strategy is employed. Specifically, in the original label sequence $Y = \{y_0, y_1, ..., y_T\}$, $y_i = 1$ means the speaker change point, and the points between two change points are all zeros. Here, with the fuzzing strategy, the label value decreases to zero from the change point linearly within 0.2s. Labels that are more than 0.2s away from the nearest change point are set to zero. With updated label y_i, the loss function for the classification is given by:

$$\mathcal{L}_p = \frac{1}{T}\sum_{i=1}^{T} ||\hat{y}_i - y_i||, \tag{2}$$

where $\hat{y}_i$ is predicted value.

Contrastive learning, which yields the contrastive loss, aims to ensure the distinctiveness of representations generated by each Conformer block layer. This serves to mitigate the risk of overfitting during the training of the SCD model. The fundamental concept is to make representations between two change points distinctive from those of adjacent regions. Hence, contrastive learning for SCD aims to enhance the similarity of representations within the same segment while diminishing the similarity with representations in adjacent segments. Here, a segment refers to the region between two speaker change boundaries.

As illustrated on the right side of Figure 1, given a frame-level representation h_i^j as the anchor, where i means the position index of the representation sequence and j means the layer index from N Conformer block layers, the positive sample h_p^j, is randomly chosen from the same segment. Simultaneously, the negative sample h_n^j is randomly selected from one adjacent segment, either on the right one or on the left one, or a randomly sampled vector if no adjacent segment exists.

Based on the anchor h_i^j, the positive sample h_p^j, and the negative sample h_n^j, the contrastive loss is defined as:

$$\begin{aligned}\mathcal{L}_c = -\frac{1}{T \cdot N}\sum_{i=1}^{T}\sum_{j=1}^{N}(&\log S(h_i^j, h_p^j) \\ &+ \log[1 - S(h_i^j, h_n^j)]),\end{aligned} \tag{3}$$

where S is to calculate the cosine similarity between two frame-level features and is given by

$$S(h_i, h_p) = \frac{h_i \cdot h_p}{||h_i|| \cdot ||h_p||} \tag{4}$$

The total loss is calculated by:

$$\mathcal{L} = \mathcal{L}_p + \alpha\mathcal{L}_c \tag{5}$$

where α is a hyper-parameter to balance the weight between $\mathcal{L}_p$ and $\mathcal{L}_c$.

2.3. SSL Features for SCDNet

The intermediate representations from different layers of the same pre-trained SSL model typically exhibit distinct properties [20]. Hence, directly utilizing features from the last layer may not be optimal. To effectively identify the most influential layer for the SCD task, a weighting fusion strategy is employed to assess the contribution of each layer's representation. To be specific, for an SSL model with L layers, the representation from layer l is denoted as X_l, and the fused representation is obtained as follows:

$$X = \sum_{l=1}^{L} \sigma_l X_l \tag{6}$$

where σ_l is a learnable parameter that weights the representation from layer l. Following the completion of training, a larger σ_l suggests a greater contribution from the corresponding layer. This information can be utilized to identify the most influential layer for extracting representations in the SCD task.

2.4. Fine-tuning SSL Models for SCD

In addition to the off-the-shelf representation-based SCDNet, we also assess the performance of various SSL models in the SCD task through fine-tuning. Following the methodology outlined in [2], only the parameters from the transformer layers and the decision layer are updated during the fine-tuning process. However, unlike [2], where the fine-tuning employs a multi-task loss function, in this study, the loss function is based on Eq. 5.

This fine-tuning approach serves a dual purpose: it compares the performance of fine-tuning different SSL models in the SCD task and enables a direct comparison between the multi-task-based loss in [2] and the proposed loss function.

3. Experimental settings

3.1. Dataset and Evaluation

Four real datasets, including AMI [21, 22], AliMeeting [23], AISHELL-4 [24], and DIHARD3 [25], are used to evaluate the proposed method. For the AMI dataset, the "headset mix" recordings are utilized. The far channel 0 and channel 0 of AliMeeting and AISHELL-4 are adopted, respectively. In addition to the above real datasets, an artificial dataset is created from the "train-other-500" subset of LibriSpeech [26] based on the simulation procedure described in [27].

Considering the widespread use of AMI in the SCD task, the comparison with other methods is performed on the AMI dataset, while other datasets are used to further validate the robustness of SCDNet and demonstrate the effectiveness of the contrastive learning method.

Following [2], purity (Pur) and coverage (Cov) scores [28] are adopted as the evaluation metric for the SCD task, and F1 presents the harmonic mean of these two. The Python library *pyannote.metrics*[1] [29] is used to compute the corresponding metric.

3.2. Implementation details

The SCDNet comprises a 3-layer Conformer block ($N = 3$) with a hidden dimension of 384. The parameter α in Eq. 5 is set as 0.05. During inference, a threshold of 0.35 is employed to binarize the predicted probabilities of speaker change points generated by the model.

4. Experimental Results

4.1. SSL Representation Comparison

Various recently popular SSL models, including wav2vec 2.0, Hubert, WavLm, and their different scales are taken into consideration, which can be found in Table 1. Both SCDNet-based and fine-tuning-based methods are employed to explore the effectiveness of these models in the SCD task.

The SCDNet-based SSL exploration is initiated with the weighting fusion strategy to examine which layer's representation from a given SSL model is most influential in the SCD task. Figure 2 illustrates the learnable weighting values (σ_l in Eq. 6) corresponding to different transformer layers (l) of an SSL model. A higher value for a layer indicates that the representation from this particular layer contributes more significantly to the final representation, in the context of the SCD task.

[1]Downloaded from https://pyannote.github.io/

Table 1: *The details of SSL models' parameters and pre-training data.*

Model	Parameters(M)	Data
hubert-base-ls960 [15]	95	LS-960
wav2vec2-base [14]	95	LS-960
wavlm-base [16]	95	LS-960
hubert-large-ll60k [15]	317	LL-60k
wav2vec2-large-xlsr-53 [14]	317	LL-60k
wavlm-large [16]	317	MIX-94k
SCDNet	33	-

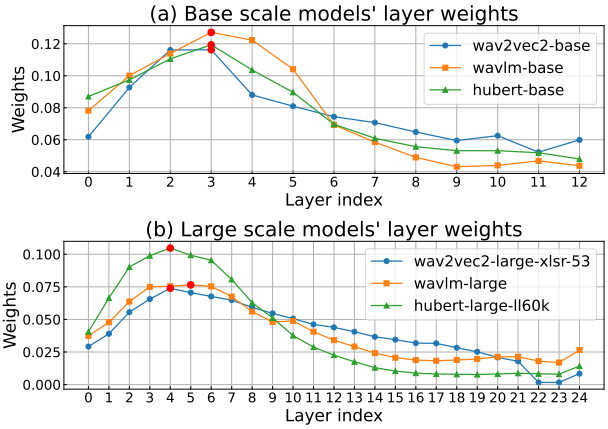


Figure 2: *Weighting values of different layers in the weighted representation fusion method. Experiments are conducted using the AMI dataset.*

As depicted in Figure 2, the weighting values from different models, irrespective of whether they are base or large models, exhibit a similar trend. Specifically, these values increase from the initial layer to a certain layer and then gradually decrease. This trend aligns with the observation in [20], which suggests that the initial layers contain more acoustic information, while the deeper layers contain more semantic information. In the SCD task, both acoustic features and semantic information are valuable. The intermediate layers, striking a balance between acoustic and semantic information, demonstrate more significant contributions than the representations from the two ends.

The performance of representations from the layer with the highest weighting value and the last layer is summarized in Table 2. It is evident that, for each model, the intermediate representation with the highest weighting value outperforms that achieved by the last layer. This underscores the effectiveness of the weighting fusion method in identifying the influential layer, as opposed to directly utilizing the last layer. Comparing all the results, although the representation from layer 3 of Wavlm-base is inferior to the best value achieved by layer 4 of wav2vec 2.0-large, its smaller model scale and less obvious performance disadvantage make it more suitable for SCDNet.

The fine-tuning-based SSL comparison for the SCD is presented in Table 3. As can be seen, WavLm-based methods both with large scale and base scale achieve the best performance compared with other SSL models with a similar scale, indicating that WavLm is particularly well-suited for the SCD task.

Table 2: *SCDNet performance based on various SSL features on AMI dataset.*

Model	Scale	Layer	Cov(%)	Pur(%)	F1(%)
Hubert	base	3	94.46	91.62	93.01
		12	91.97	90.73	91.35
	large	4	94.28	91.68	92.96
		24	96.71	85.84	90.95
wav2vec 2.0	base	3	92.96	92.13	92.55
		12	92.16	91.44	91.80
	large	4	93.69	92.86	**93.27**
		24	94.79	67.18	78.63
WavLm	base	3	93.72	92.35	**93.03**
		12	91.72	90.37	91.04
	large	5	94.58	91.50	93.01
		24	94.22	91.62	92.91

Table 3: *SCD performance by fine-tuning various SSL models on AMI dataset.*

Model	Scale	Cov(%)	Pur(%)	F1(%)
Hubert	base	92.82	93.00	92.91
wav2vec 2.0		92.19	93.48	92.83
WavLm		93.43	93.60	**93.51**
Hubert	large	93.17	93.20	93.18
wav2vec 2.0		91.63	93.34	92.47
WavLm		94.11	94.63	**94.37**

4.2. Comparison with SOTA methods

The comparison of the proposed SCDNet with previous methods is presented in Table 4. In this table, SCDNet refers to the proposed model with the representation from layer 3 of WavLm-base as input, and it will be the default setting hereafter unless specifically mentioned otherwise. It is evident that SCDNet achieves the best performance, with a relative gain of 2.5% in terms of F1 compared to the previous SOTA performance achieved by [2]. This result highlights the effectiveness of the design of SCDNet.

It is noteworthy that the previous SOTA performance in [2] is based on fine-tuning the wav2vec2-base model, the same model as presented in Table 3. However, our results achieved by fine-tuning the wav2vec-base with the proposed contrastive learning method are notably superior to those in [2]. This superiority underscores the effectiveness of the proposed contrastive learning approach. Further evidence of this superiority will be explored in the following ablation study.

Table 4: *Comparison of the proposed scheme with previously reported results for the SCD task on AMI dataset.*

Method	Cov(%)	Pur(%)	F1(%)
Kunešová *et al*. [2]	91.68	89.91	90.79
Su *et al*. [30]	91.75	85.68	88.61
Fan *et al*. [11]	89.81	83.92	86.76
pyannote [31]	84.20	90.40	87.19
SCDNet	93.72	92.35	**93.03**

Table 5: *Results of SCDNet w/wo contrastive learning (CL).*

Dataset	CL	Cov(%)	Pur(%)	F1(%)
AMI	✓	93.72	92.35	**93.03**
	×	89.25	94.03	91.57
AliMeeting	✓	93.57	86.61	**89.95**
	×	94.52	84.12	89.02
AISHELL-4	✓	91.75	91.32	**91.53**
	×	84.78	92.51	88.48
DIHARD3	✓	94.16	90.36	**92.22**
	×	93.86	89.96	91.88

4.3. Ablation Study

To demonstrate the effectiveness of the proposed contrastive learning method, ablation experiments were conducted on more datasets, and the corresponding results are presented in Table 5. All F1 values achieved by the model trained without contrastive learning are lower than those with contrastive learning on the same database. Specifically, a relatively 3.4% higher value is observed when contrastive learning is adopted compared to that without contrastive learning on the AI-SHELL-4 database. These results collectively demonstrate the efficacy of the proposed contrastive learning in enhancing the performance of SCDNet.

4.4. Experiments with artificial database

To further assess the generalization ability of SCDNet and provide additional references for future work, we trained SCDNet based on artificial data and evaluated the model on different datasets. The results are shown in Table 6. As demonstrated, in the four test sets, the performance degradation of SCDNet trained with artificial data is within 10% compared to the model trained directly in the corresponding domain, as shown in Table 5. This showcases that the proposed SCDNet can generalize to unseen domains when trained solely with artificial data.

Table 6: *Results of SCDNet trained with the artificial data.*

Dataset	Cov(%)	Pur(%)	F1(%)
AMI	92.94	82.32	87.31
AliMeeting	90.37	73.76	81.23
AISHELL-4	85.72	87.29	86.50
DIHARD3	96.35	84.27	89.90

5. Conclusions

In this paper, we introduce a self-supervised learning feature-based SCD model called SCDNet. With SCDNet, various SSL models, including Hubert, wav2vec 2.0, and WavLm, are explored. A weighting fusion strategy is employed to assess the effectiveness of representations from different layers in a pre-trained SSL model. This strategy efficiently identifies a better layer compared to the last layer. Results obtained by SCDNet using different representations indicate the suitability of the representation from layer 3 of the WavLm-base model. Additionally, a fine-tuning-based method is employed to evaluate different SSL models for the SCD task, with the results highlighting the strong performance of WavLm, regardless of the scale. Furthermore, both SCDNet and the fine-tuning-based method outperform previous SOTA results, showcasing the efficacy of SCDNet's design and the effectiveness of the proposed contrastive learning approach.

6. References

[1] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 4945–4949.

[2] M. Kunešová and Z. Zajíc, "Multitask detection of speaker changes, overlapping speech and voice activity using wav2vec 2.0," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2023, pp. 1–5.

[3] R. Yin, H. Bredin, and C. Barras, "Neural speech turn segmentation and affinity propagation for speaker diarization," in *INTERSPEECH*. ISCA, 2018, pp. 1393–1397.

[4] L. Sari, M. Hasegawa-Johnson, and S. Thomas, "Auxiliary networks for joint speaker adaptation and speaker change detection," *Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 324–333, 2020.

[5] G. Donabauer, U. Kruschwitz, and D. Corney, "Making sense of subtitles: Sentence boundary detection and speaker change detection in unpunctuated texts," in *Companion Proceedings of the Web Conference 2021*, 2021, pp. 357–362.

[6] T. Kemp, M. Schmidt, M. Westphal, and A. Waibel, "Strategies for automatic segmentation of audio data," in *International Conference on Acoustics, Speech and Signal Processing*, vol. 3. IEEE, 2000, pp. 1423–1426.

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

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2018, pp. 5329–5333.

[9] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge." in *INTERSPEECH*. ISCA, 2018, pp. 2808–2812.

[10] M. Hrúz and M. Hlaváč, "Lstm neural network for speaker change detection in telephone conversations," in *Speech and Computer*. Springer, 2018, pp. 226–233.

[11] Z. Fan, L. Dong, M. Cai, Z. Ma, and B. Xu, "Sequence-level speaker change detection with difference-based continuous integrate-and-fire," *Signal Processing Letters*, vol. 29, pp. 1551–1554, 2022.

[12] W. Xia, H. Lu, Q. Wang, A. Tripathi, Y. Huang, I. L. Moreno, and H. Sak, "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 8077–8081.

[13] G. Zhao, Q. Wang, H. Lu, Y. Huang, and I. L. Moreno, "Augmenting transformer-transducer based speaker change detection with token-level training loss," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2023, pp. 1–5.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[15] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[16] 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," *Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[17] B. Nasersharif and M. Azad, "Speech emotion recognition using transfer learning and self-supervised speech representation learning," in *International Conference on Electrical Engineering*. IEEE, 2023, pp. 684–689.

[18] M. Łajszczak, G. Cámbara, Y. Li, F. Beyhan, A. van Korlaar, F. Yang, A. Joly, Á. Martín-Cortinas, A. Abbas, A. Michalski *et al.*, "Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data," *CoRR*, 2024.

[19] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *INTERSPEECH*, p. 5036–5040, 2020.

[20] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2023, pp. 1–5.

[21] W. Kraaij, T. Hain, M. Lincoln, and W. Post, "The ami meeting corpus," in *International Conference on Methods and Techniques in Behavioral Research*, 2005, pp. 28–39.

[22] 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 pre-announcement," in *Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 28–39.

[23] 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 *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 6167–6171.

[24] Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu *et al.*, "Aishell-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario," *INTERSPEECH*, pp. 3665–3669, 2021.

[25] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third dihard diarization challenge," *INTERSPEECH*, pp. 3570–3574, 2021.

[26] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015, pp. 5206–5210.

[27] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *INTERSPEECH*, pp. 4300–4304, 2019.

[28] R. Yin, H. Bredin, and C. Barras, "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *INTERSPEECH*. ISCA, 2017, pp. 3827–3831.

[29] H. Bredin, "pyannote. metrics: A toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems." in *INTERSPEECH*. ISCA, 2017, pp. 3587–3591.

[30] H. Su, D. Zhao, L. Dang, M. Li, X. Wu, X. Liu, and H. Meng, "A multitask learning framework for speaker change detection with content information from unsupervised speech decomposition," in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 8087–8091.

[31] 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 *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 7124–7128.