SELF-SUPERVISED LEARNING WITH BI-LABEL MASKED SPEECH PREDICTION FOR STREAMING MULTI-TALKER SPEECH RECOGNITION

Zili Huang[1†], *Zhuo Chen*[2], *Naoyuki Kanda*[2], *Jian Wu*[2], *Yiming Wang*[2], *Jinyu Li*[2], *Takuya Yoshioka*[2], *Xiaofei Wang*[2], *Peidong Wang*[2]

[1]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD, USA
[2]Microsoft, One Microsoft Way, Redmond, WA, USA

ABSTRACT

Self-supervised learning (SSL), which utilizes the input data itself for representation learning, has achieved state-of-the-art results for various downstream speech tasks. However, most of the previous studies focused on offline single-talker applications, with limited investigations in multi-talker cases, especially for streaming scenarios. In this paper, we investigate SSL for streaming multi-talker speech recognition, which generates transcriptions of overlapping speakers in a streaming fashion. Firstly, we observe that conventional SSL techniques do not work well on this task due to the poor representation of overlapping speech. We then propose a novel SSL training objective, referred to as *bi-label masked speech prediction*, which explicitly preserves representations of all speakers in overlapping speech. We investigate various aspects of the proposed system, including data configuration and quantizer selection. The proposed SSL setup achieves substantially better word error rates on the LibriSpeechMix dataset.

Index Terms— Self-supervised learning, multi-talker automatic speech recognition

1. INTRODUCTION

Self-supervised learning (SSL), which extracts supervision signals from data itself, is a fast-growing subcategory of unsupervised learning approaches [1]. In SSL pipeline, an upstream model is pre-trained on massive unlabeled data with some pretext tasks derived from the data itself. Then it is adapted for specific downstream tasks with a small amount of labeled data by either using the upstream model as a feature extractor [2, 3] or directly fine-tuning it together with additional task-specific layers [4, 5].

SSL has been widely explored due to its great performance and low adaptation cost. In speech applications, such SSL-based pre-training has achieved remarkable performance for various downstream tasks including speech recognition [4, 5], speaker recognition [6, 7], emotion recognition [8], etc. Since the model learns more generalizable task-agnostic representations in the pre-training stage, it only requires a small amount of labeled data in the fine-tuning stage. For example, wav2vec 2.0 [4] outperforms the previous state-of-the-art automatic speech recognition (ASR) results on the LibriSpeech 100h benchmark with just 1h of labeled speech.

Despite the great achievements in various speech tasks, SSL remains under-explored for streaming multi-talker audio processing tasks. It is known that natural human conversations contain a considerable amount of speech overlaps [9], thus handling overlapping speech in real time is in great demand for many real applications. Nevertheless, most of the existing SSL techniques were explored under single-talker speech conditions for both pre-training and fine-tuning [10, 11, 12, 13, 14, 4, 5]. Recently, WavLM [3] was proposed with the multi-talker data augmentation scheme, called "utterance mixing", which was proven to be effective for several multi-talker tasks such as speech separation and speaker diarization [3]. However, WavLM was designed with an offline model architecture, limiting its usage in streaming scenarios. Moreover, WavLM was pre-trained with a conventional masked speech prediction (MSP) loss, where the model predicts the masked tokens of the primary (i.e. dominant) speaker for augmented multi-talker audios, which could potentially hurt the representations of other speakers. Such a training scheme could be sub-optimal for tasks where every speaker is equally important.

In this paper, we investigate SSL-based pre-training for the streaming multi-talker ASR task, in which we perform real-time speech recognition for all speakers in a conversation containing overlapped speech. We propose a novel *bi-label MSP* objective that forces the model to learn a representation of all speakers in overlapping speech instead of focusing on a single dominant speaker. We also explore several aspects of the proposed bi-label SSL model to further improve its performance, including the ratio of utterance mixing and quantizer type. We conducted our experiment based on the streaming multi-talker ASR with the token-level serialized output training (t-SOT) [15]. Our experimental results on the LibriSpeechMix [16] dataset reveal that, with the proposed bi-label MSP objective, appropriate pre-training data configuration, and quantizer, the streaming multi-talker ASR accuracy can be significantly improved.

2. RELATED WORKS

2.1. HuBERT and WavLM

Our work is based on two SSL models: HuBERT [5] and WavLM [3]. Similar to BERT [17], HuBERT uses MSP as the pretext task: the acoustic embeddings produced by the convolutional neural network encoder are partially masked, and the transformer encoder is trained to predict the pseudo labels of masked regions (Fig. 1 left). The distribution over the pseudo labels is formulated as

$$p(c|o_t) = \frac{\exp(\cos(o_t \cdot W^P, e_c)/\gamma)}{\sum_{c'=1}^{C} \exp(\cos(o_t \cdot W^P, e'_c)/\gamma)}, \quad (1)$$

where W^P is a projection matrix, o_t is the output logit at the time frame t, e_c is the embedding of the pseudo label $c \in \{1, ..., C\}$, $\cos(a, b)$ is the cosine similarity between a and b, and γ is the scale of the logit. The pseudo labels are generated by clustering either

[†]Work performed during an internship at Microsoft.

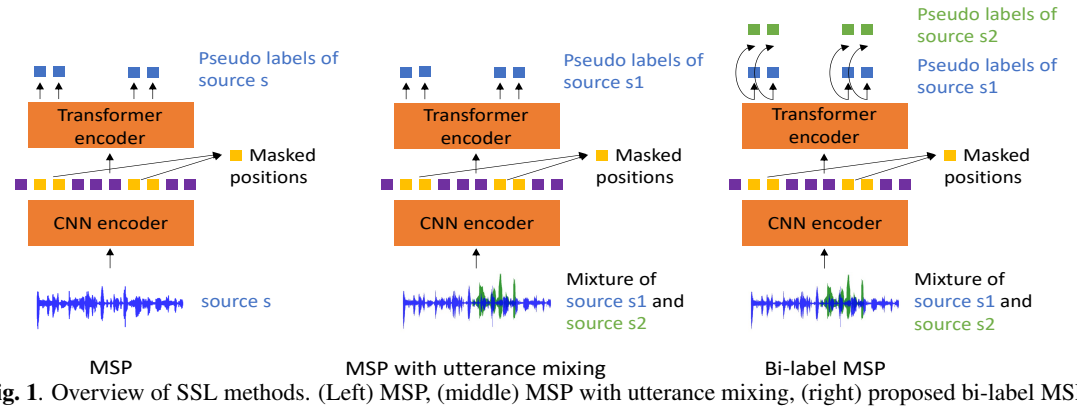

Fig. 1. Overview of SSL methods. (Left) MSP, (middle) MSP with utterance mixing, (right) proposed bi-label MSP

acoustic features (e.g., mel-frequeney cepstrum coefficient or mel-filterbank (FBANK)) or hidden representations from a prior generation of the HuBERT model.

WavLM [3] introduced several modifications to HuBERT to enhance spoken content modeling and speaker identity preservation. Firstly, it employed gated relative position bias for the Transformer structure. Secondly, it introduced data augmentation by mixing the input audio with noise or interfering speech (Fig. 1 middle). Thirdly, it scaled up the training data size and variety to further improve the robustness of learned representations. Among these modifications, WavLM's data augmentation scheme enforces the model to execute a denoising task in addition to the original masked speech prediction task. This significantly improved the performance on speaker-related tasks such as speaker diarization and speech separation.

2.2. Streaming multi-talker ASR based on t-SOT

The token-level serialized output training (t-SOT) [15] is a framework for training streaming multi-talker end-to-end (E2E) ASR models [18] that can generate transcriptions of multiple overlapping speakers with limited latency. The key to t-SOT lies in the serialization of multi-talker transcriptions. Suppose we have time- and speaker-annotated transcriptions of multiple speakers (e.g., "hello how are you" from speaker A and "fine thank you" from speaker B). We first create a single sequence of tokens by simply concatenating the transcriptions of all speakers (e.g., "hello how are you fine thank you"). Next, we reorder the tokens in that sequence based on the end time of each token (e.g., "hello how fine are you thank you"). Finally, we insert a special token ⟨cc⟩ when the adjacent tokens are attributed to different speakers (e.g., "hello how ⟨cc⟩ fine ⟨cc⟩ are you ⟨cc⟩ thank you").

A conventional streaming E2E ASR model, such as transformer transducer (TT) [19, 20], can be trained with overlapping speech annotated with such serialized transcriptions. During inference, the ASR system generates transcriptions including ⟨cc⟩, which is then "deserialized" into two streams of transcriptions based on the estimation of ⟨cc⟩. It was shown that the t-SOT-based multi-talker ASR model achieved better accuracy than prior multi-talker models while keeping the model architecture and computational cost the same as conventional single-talker models.

3. PRE-TRAINING OF STREAMING MULTI-TALKER ASR WITH BI-LABEL MSP

In this section, we introduce our SSL-based pre-training framework for streaming multi-talker E2E ASR. We first describe our proposed bi-label MSP objective in Section 3.1, and then introduce our strategy to pre-train streaming models in Section 3.2. Finally, we explain data configurations for the pre-training in Section 3.3.

3.1. Bi-label MSP objective

The original MSP objective is designed to predict pseudo labels of the masked speech region given the surrounding speech as a context. When combined with utterance mixing, as proposed in WavLM, MSP enforces the model to learn representations that best estimate the masked speech of the primary (i.e. dominant) speaker while ignoring the speech of the secondary speaker. We speculate that this formulation may prevent the model from learning a good representation of the speech from the secondary speaker.

Considering this, we propose a bi-label MSP objective for pre-training, where the model predicts the pseudo labels of both the primary and the secondary speakers. The overview of the bi-label MSP is shown on the right side of Fig. 1. As illustrated in the figure, the transformer encoder has two output nodes, one predicts the pseudo label of the primary speaker while the other predicts the pseudo label of the secondary speaker. If the secondary speaker is not present in the masked regions, a special ⟨blank⟩ token is assigned to that region. The loss function is formulated as

$$\mathcal{L} = \sum_{t \in M} -\log(p(r_t^{pr}|o_t^{pr})) - \log(p(r_t^{sc}|o_t^{sc})), \qquad (2)$$

where r_t^{pr} and r_t^{sc} are the pseudo labels of the primary and the secondary speaker at time frame t, o_t^{pr} and o_t^{sc} represent the output logits for the primary and the secondary speakers at t. M denotes the set of all masked time frames. Note that, in our implementation, the output nodes for the primary and the secondary speakers are predetermined, rather than solving the permutation using permutation invariant training [21, 22, 23]. This is because the utterance mixing algorithm guarantees that the speech duration of the secondary speaker is substantially shorter than that of the primary speaker.

3.2. Attention mask for streaming models

Prior SSL models based on MSP objective [5, 3] utilized a standard self-attention mechanism, in which every computation is executed by accessing the entire input sequence. This property restricts the SSL model to offline scenarios only. In this work, we adopt a masking strategy originally proposed for streaming TT [20], where a specially designed attention mask is applied to constrain the model to see only limited future information for each computation.

The attention mask is defined as a $T \times T$ matrix $\mathbf{S}$, where T is the embedding length, as exemplified in Fig. 2. $\mathbf{S}[i, j] = 1$ indicates that the input at the jth frame can be used to compute the output at the ith frame. The matrix is segmented with fix-sized chunks for both vertical and horizontal directions (the chunk size is three in Fig.

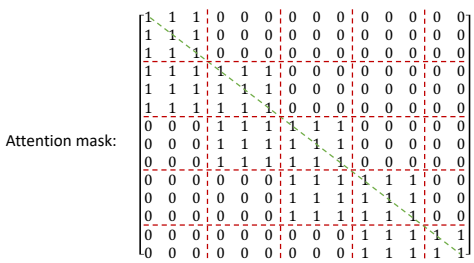

Fig. 2. Attention mask matrix $\mathbf{S}$ for streaming models. If $\mathbf{S}[i, j] = 1$, the jth-frame input can be used for computing ith-frame output.

Table 1. WER (%) on LibriSpeechMix for t-SOT TT-18 with different pre-training configurations. MSP stands for "masked speech prediction". All models have 160 msec of algorithmic latency.

Pre-training		Dev WER (%)		Test WER (%)	
Objective	Quantizer	1spk	2spk	1spk	2spk
-	-	15.42	39.12	15.69	39.52
MSP	FBANK	13.17	36.13	13.20	35.29
Bi-label MSP	FBANK	13.29	25.68	13.90	25.78
MSP	HuBERT	10.77	17.24	11.30	17.25
Bi-label MSP	HuBERT	10.82	15.84	11.19	15.30
MSP	Phoneme	9.80	15.45	9.96	15.13
Bi-label MSP	Phoneme	**9.47**	**13.89**	**9.84**	**13.74**

2). Given the indices $\mathcal{I}_l$ corresponding for l-th chunk, the matrix $\mathbf{S}$ is defined such that $\mathbf{S}[i, j] = 1$ if $(i, j \in \mathcal{I}_l$ for any $l)$ or $(i \in \mathcal{I}_l$ and $j \in \mathcal{I}_{l'}$ for $l - h < l' < l)$, where h is a hyper-parameter to determine how far the history information can be accessed ($h = 2$ in Fig. 2). With this masking strategy, the left receptive field (history) grows with the number of transformer layers, while the right receptive field (future look-ahead) remains the same. The algorithm latency (or the duration of the future look-ahead) of the model is determined by the chunk size. In our work, we use the same masking matrix for SSL-based pre-training and fine-tuning.

3.3. Data configurations for multi-talker ASR pre-training

There are several data configurations that are especially important for multi-talker ASR pre-training. Firstly, data augmentation plays a crucial role in determining the characteristics of extracted representation. In the case of WavLM, the training data was augmented such that random noise was mixed into 10% of the training samples while random secondary speech was mixed into another 10% of the training samples. Such a configuration effectively enforces the model to extract representations of the primary (i.e. dominant) speaker, even from the overlapping speech. However, it has not yet been investigated if this configuration is optimal for multi-talker tasks where the representations of all speakers are equally important. In this work, we thus explore the possibility of drastically increasing the ratio of the utterance mixing for multi-talker modeling.

Secondly, we also explore several quantizers for generating the pseudo labels. The choice of quantizer is crucial in MSP-based pre-training. For instance, it is known that a quantizer with a higher correlation with phonemes generally benefits ASR accuracy [5]. In this study, in addition to clustering on FBANK or HuBERT embedding, we also investigate a phoneme-based quantizer proposed in [24], in which a hybrid ASR system trained with a small amount of transcribed data is used for generating pseudo phoneme labels. We conduct various experiments to investigate the impact of quantizers on our results.

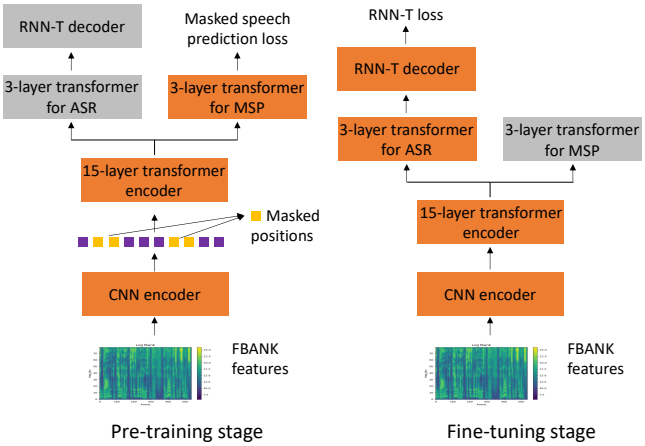

Fig. 3. Pre-training pipeline for transformer transducer based t-SOT model. Orange blocks are updated while grey blocks are frozen.

Table 2. WER (%) on LibriSpeechMix for t-SOT TT-18 with different data augmentation configuration for pre-training. We used the original MSP objective with the HuBERT quantizer. All models have 160 msec of algorithmic latency.

Data augmentation			Dev WER (%)		Test WER (%)	
No aug.	Noise	Speech	1spk	2spk	1spk	2spk
1.0	-	-	12.08	36.62	12.65	35.81
0.8	0.1	0.1	10.80	21.21	11.41	20.79
0.5	-	0.5	**10.77**	**17.24**	**11.30**	**17.25**

4. EXPERIMENTS

4.1. Experimental settings

4.1.1. Data

We evaluated our proposed framework on the LibriSpeechMix [16] evaluation set. This dataset is simulated from LibriSpeech [25] by randomly mixing utterances with random delays. In our experiments, we used the single-speaker and two-speaker-mixed evaluation sets. We adopt the same multi-talker ASR evaluation metric as [15]. Specifically, we considered all possible speaker permutations between the hypotheses and references, and the permutation with the minimum number of errors was chosen to compute the word error rate (WER).

We trained our model using LibriSpeech 960h (LS-960), out of which 100h of speech (train-clean-100, or LS-100) were used as labeled data, while the remaining 860h (consisting of train-clean-360 and train-other-500) were considered unlabeled.

4.1.2. Model configuration

In our experiment, we used TT with the chunk-wise mask [20] as described in Section 3.2. We used the same configuration as "TT-18" in [15]. Specifically, the input to the network was 80-dim FBANK with a 10ms stride, normalized by the mean and variance computed on the entire training data. The encoder consisted of 2 convolution layers that downsampled the acoustic features by a factor of 4, and 18 layers of transformers with relative positional encoding. Each transformer layer contained a 512-dim multi-head attention with 8 heads and a 2048-dim feed-forward layer. The prediction network of TT is a 2-layer LSTM network with 1024 hidden units.

Our training pipeline is depicted in Fig. 3. For pre-training, we optimized the bottom 15 layers of the encoder and three additional transformer layers using MSP or the proposed bi-label MSP

Table 3. WER (%) on LibriSpeechMix test set for t-SOT TT-18 with different algorithmic latency l.

Pre-training		$l = 160$ msec		$l = 640$ msec		$l = 2560$ msec		$l = \infty$ (offline)	
Objective	Quantizer	1spk	2spk	1spk	2spk	1spk	2spk	1spk	2spk
-	-	15.69	39.52	13.71	34.71	12.21	29.56	11.00	24.03
Bi-label MSP	HuBERT	11.19	15.30	8.99	12.41	8.15	10.94	6.78	10.45
Bi-label MSP	Phoneme	9.84	13.74	8.39	11.24	7.20	9.55	6.46	8.51

objective on the LS-960. Once the model was pre-trained, we further fine-tuned the TT with RNN-T loss on simulated mixtures created from LS-100. The simulated mixtures for fine-tuning were generated on-the-fly, ensuring that the ratio between single-speaker and two-speaker mixed samples was 50%:50%. To create the two-speaker mixed samples, we mixed two random utterances from LS-100, adding a random delay ranging from 0 to the duration of the first utterance to the second utterance to simulate partially overlapping speech. We also applied speed and volume perturbation to further increase the variety of the fine-tuning data.

We used the AdamW optimizer and a linear decay learning rate scheduler for both pre-training and fine-tuning. During pre-training, we trained the model on 16 NVIDIA V100 GPUs for 125k updates, with a batch size of 480 seconds per GPU and a peak learning rate of 1.5e-3. During fine-tuning, we fine-tuned the model on 16 NVIDIA V100 GPUs for 35k updates, with a batch size of 60 seconds per GPU and a peak learning rate of 3e-4.

As mentioned in section 3.3, we experimented with three quantizers: FBANK, HuBERT, and phonemes. For the FBANK quantizer, we set the size of the codebook to 500. For the HuBERT quantizer, we extracted hidden representations from the 9th layer of the HuBERT base model [1], and clustered them into 500 groups using the K-means algorithm. For the phoneme quantizer, we first trained a hybrid ASR model on LS-100, and then decoded the LS-960 with it and a 3-gram language model. The hypothesis lattices were rescored, and the phoneme labels were inferred from the 1-best path. In total, there were 347 distinct phonemes.

4.2. Main results

Our main experimental results are shown in Table 1. As a baseline, we first trained a TT-18 model on the LS-100-based multi-talker fine-tuning data without any pre-training. The results are shown in the first row of Table 1. We found that, with such limited training data, the WERs were very poor for both the single-speaker and two-speaker-mixed evaluation sets.

We then performed the pre-training based on the MSP objective with the FBANK quantizer, and presented the results in the second row. During pre-training, we applied utterance mixing with a 50% probability. We observed a large improvement in the single-speaker test set, where the WER reduced by 15.9% relatively (15.69% to 13.20%). However, we only observed a relative WER improvement of 10.7% (from 39.52% to 35.29%) for the two-speaker mixed test set. These results suggest that the MSP objective with utterance mixing was heavily biased towards the primary speaker, and did not perform well for tasks requiring the modeling of all speakers.

After that, we replaced MSP with the proposed bi-label MSP objective, and the results are presented in the third row. Compared to the MSP result, we observed a significant reduction in WER from 35.29% to 25.78% (26.9% relative) on the two-speaker-mixed test set. However, there was a slight WER degradation on the single-speaker test set, increasing from 13.20% to 13.90%.

We finally evaluated the HuBERT-based quantizer and phoneme-based quantizer, with their results listed in the last four rows. In this experiment, we consistently observed improvements in the two-speaker-mixed test sets using the proposed bi-label MSP objective. The phoneme-based quantizer achieved the best WER. With the phoneme-based quantizer, the proposed bi-label MSP objective showed no side effects in the single-speaker test set while remarkably improving the two-speaker-mixed test set.

4.3. Impact of the pre-training data configuration

We present the impact of pre-training data configuration in Table 2. In this experiment, we used the conventional MSP with HuBERT quantizer for pre-training. The first row in the paper shows the configuration without any data augmentation (as used in the original HuBERT), the second row shows the configuration used for WavLM, and the third row shows the configuration used for our experiment as described in the previous section. Pre-training without any data augmentation resulted in a very poor WER, especially for the two-speaker-mixed evaluation sets (first row). The ASR performance significantly improved after adding a small ratio of noise and interference speech (second row). Finally, our configuration, which applied utterance mixing for 50% of the pre-training data, achieved the best WER for both single-speaker and two-speaker-mixed speech.

4.4. Latency variation

We also evaluated the model performance for different latency configurations, and the results are reported in Table 3. We controlled the algorithmic latency by adjusting the chunk size in the self-attention mask, as introduced in section 3.2. As shown in the first row, if no pre-training is applied, the WER for the two-speaker-mixed test set was over 20%, even for the offline model. Our best-performing setup (bi-label MSP objective with the phoneme quantizer) dramatically improved the WER for all latency configurations. The 160ms latency model, pre-trained with the bi-label MSP objective and the phoneme quantizer, achieved a better WER compared to the offline model without pre-training.

5. CONCLUSION

In this paper, we investigated SSL-based pre-training for streaming multi-talker ASR. We proposed the bi-label MSP objective, which enforces the model to learn speech representations of all speakers instead of focusing on the primary speaker. We also explored several aspects of multi-talker ASR pre-training, including the pre-training data configuration and the type of quantizer. Our experimental results on LibriSpeechMix showed that our proposed SSL framework significantly reduced the WER for both non-overlapping and overlapping speech, especially with a prominent improvement on overlapping speech.

[1]The HuBERT base model was pre-trained on LS-960.

6. REFERENCES

[1] Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–34, 2022.

[2] Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, et al., "SUPERB: Speech processing universal performance benchmark," in *Interspeech*, 2021, pp. 1194–1198.

[3] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[4] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.

[5] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[6] Zhiyun Fan, Meng Li, Shiyu Zhou, and Bo Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," in *Interspeech*, 2021, pp. 1509–1513.

[7] Zhengyang Chen, Sanyuan Chen, Yu Wu, Yao Qian, Chengyi Wang, Shujie Liu, Yanmin Qian, and Michael Zeng, "Large-scale self-supervised speech representation learning for automatic speaker verification," in *ICASSP*, 2022, pp. 6147–6151.

[8] Leonardo Pepino, Pablo Riera, and Luciana Ferrer, "Emotion recognition from speech using wav2vec 2.0 embeddings," in *Interspeech*, 2021, pp. 3400–3404.

[9] Özgür Çetin and Elizabeth Shriberg, "Analysis of overlaps in meetings by dialog factors, hot spots, speakers, and collection site: Insights for automatic speech recognition," in *Interspeech*, 2006, pp. 293–296.

[10] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[11] Morgane Riviere, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP*, 2020, pp. 7414–7418.

[12] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," in *Interspeech*, 2019, pp. 146–150.

[13] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019, pp. 3465–3469.

[14] Shaoshi Ling and Yuzong Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[15] Naoyuki Kanda, Jian Wu, Yu Wu, Xiong Xiao, Zhong Meng, Xiaofei Wang, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Takuya Yoshioka, "Streaming multi-talker asr with token-level serialized output training," in *Interspeech*, 2022, pp. 3774–3778.

[16] Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka, "Serialized output training for end-to-end overlapped speech recognition," in *Interspeech*, 2020, pp. 2797–2801.

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[18] Jinyu Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[19] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP*, 2020, pp. 7829–7833.

[20] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *ICASSP*, 2021, pp. 5904–5908.

[21] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *ICASSP*, 2017, pp. 241–245.

[22] Peidong Wang, Zhuo Chen, Xiong Xiao, Zhong Meng, Takuya Yoshioka, Tianyan Zhou, Liang Lu, and Jinyu Li, "Speech separation using speaker inventory," in *ASRU*, 2019, pp. 230–236.

[23] Peidong Wang, Zhuo Chen, DeLiang Wang, Jinyu Li, and Yifan Gong, "Speaker separation using speaker inventories and estimated speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 537–546, 2020.

[24] Chengyi Wang, Yiming Wang, Yu Wu, Sanyuan Chen, Jinyu Li, Shujie Liu, and Furu Wei, "Supervision-guided codebooks for masked prediction in speech pre-training," in *Interspeech*, 2022, pp. 2643–2647.

[25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.