INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Self-supervised Learning Representation based Accent Recognition with Persistent Accent Memory

Rui Li[1*], *Zhiwei Xie*[1*], *Haihua Xu*[2], *Yizhou Peng*[3], *Hexin Liu*[4], *Hao Huang*[1,5], *Eng Siong Chng*[4]

[1]School of Information Science and Engineering, Xinjiang University, Urumqi, China
[2]Bytedance
[3]National University of Singapore, Singapore
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[5]Xinjiang Key Laboratory of Multi-lingual Information Technology, Urumqi, China

huanghao@xju.edu.cn

Abstract

Accent recognition (AR) is challenging due to the lack of training data as well as the accents are entangled with speakers and regional characteristics. This paper aims to improve AR performance from two perspectives. First, to alleviate the data insufficiency problem, we employ the self-supervised learning representations (SSLRs) extracted from a pre-trained model to build the AR models. With the help of SSLRs, it gains significant performance improvement compared with the traditional acoustic features. Secondly, we proposed a persistent accent memory (PAM) as contextual knowledge to bias the AR models. The accent embeddings that are extracted from all training data by the encoder of AR models are clustered to form an accent codebook, i.e. PAM. In addition, we propose diverse attention mechanisms to investigate the optimal utilization of PAM. We observe that the best performance is obtained by selecting the most relevant accent embeddings.

Index Terms: WavLM, Self-supervised learning, representation, accent recognition, persistent accent memory, Conformer

1. Introduction

Accent recognition (AR) or identification is important but challenging. It is vital because the accent not only contains the speaker's personal voice characteristics but also includes regional information, which is potentially crucial for speaker recognition [1–3]and speech recognition [4–12]. Unfortunately, AR is also challenging since large scale accent-labeled data is hard to come by, and therefore it is a low-resource task. As such, to obtain a desirable AR system, one needs to fully exploit both data and modeling efficiency simultaneously.

To achieve state-of-the-art AR performance, [13] employed a series of data augmentations (DAs), as well as novel modeling frameworks. Specifically, the DAs include speed perturbation [14], noise speech corruption [15], and text-to-speech that synthesizes diverse accented speech data, resulting in up to 10 times data increase. Furthermore, it also proposed to employ phone posterior-gram as AR model input features, achieving better performance over the conventional filter-bank features. Recently, to realize improved AR performance, [16] made efforts on novel modeling work and proposed an accent shift approach to the AR task. The method requires acoustic transcripts to estimate linguistic-acoustic bimodal similarity during training and test processes. However, the quality of ASR transcripts can not always be guaranteed when accents are diverse. Likewise, in [17], a joint speech and accent recognition framework is proposed, realizing mutual performance benefits for either task.

In this paper, we attempt to make efforts to boost AR performance from two perspectives. We first attempt to alleviate the data insufficiency issue by exploiting self-supervised learning representations (SSLRs) that are learned from recently proposed WavLM [18] pre-trained with self-supervised learning (SSL) method different from Wav2vec 2.0 [19, 20], as well as HuBert [21]. The advantages of WavLM mainly come from its consideration of speech denoising and yielding much performance improvement on non-speech recognition tasks in addition to ASR task, such as speaker recognition [22], speech separation [23], as well as speech enhancement [24] and speaker diariazation [25], etc. In this paper, we show that employing SSLRs can significantly improve AR performance compared with the systems that are trained from scratch with filter-bank features. More importantly, we propose a novel persistent accent memory (PAM) based attention mechanism for AR in light of efficient modeling efforts. The PAM, namely a codebook of clustered embeddings, is obtained by clustering overall utterance level accent embeddings that are extracted using AR model trained with WavLM SSLRs. It acts as an accent-aware prompter that lets the audio stream attends to the presence of accent context for each time being. To realize better accent information utilization, we try diverse attention mechanisms, and the best performance is achieved with the method that selects the best relevant memories with the pooled utterance level features from the output of the encoder.

The main contributions of this paper include the following aspects: 1) To the best of our knowledge, we are the first to employ WavLM learned representations on AR task and demonstrate its effectiveness with diverse configurations. 2) We propose to use the overall training data to build the persistent accent memory that is compact but representative and empirically effective. 3) To well exploit accent contextual information, we attempt diverse attention mechanisms that make further performance improvement.

2. Related work

Self-supervised learning (SSL) has been drawing increasing attention in speech processing society [26–28]. However, prior SSL-based pre-trained models are mostly focused on ASR performance improvement. AR as a downstream task hasn't been fully attentive. Recently, WavLM [18] has given more attention to the speaker-related tasks, as well as ASR task. As a result, it can achieve improved results on a wide range of speech-processing tasks. Therefore, we propose to use the SSLRs learned from WavLM as input features in this work for the AR recognition. Moreover, we also found SSLRs from different

Hao Huang is the correspondence author. Authors* did this work during their internship at AISG NTU-NUS joint speech lab.

10.21437/Interspeech.2023-1702

layers have different performance for different accents, and a weighting summary of SSLRs from different WavLM layers are also attempted. On model efficiency level work, our persistent accent memory is inspired by the prior works [29] that are aimed for improving ASR recognition. Here, we extend the idea for AR, and more efforts are focused on how to fully exploit the accent context information to boost accent recognition.

2.1. WavLM for SSL Representation

WavLM is a large-scale self-supervised (SS) pretraining framework for full-stack speech processing [27], using a modified Transformer [30] as the backbone. The training objective is to realize masked speech denoising and prediction. Unlike Wav2vec 2.0 [20] which employs contrastive loss [31] as an objective function, WavLM adopts masking prediction to train the model, which is the same as HuBERT [21]. But different from HuBERT, WavLM employs more unlabelled and noisy data. The noisy data include overlapped speech, which implicitly makes the WavLM yield much better performance on speaker-related tasks as mentioned. Therefore, we employ WavLM to generate SSLRs for our AR classifiers in this work.

3. Proposed Method

3.1. Persistent Accent Memory

One of the key components of our AR system is persistent accent memory (PAM). Essentially, PAM is a 256-codeword (clustered embedding) codebook that is clustered from the encoder output of AR model trained with WavLM SSLRs, and the overall embeddings are from the 160-hour accented speech in training dataset [32]. To be concise, the training dataset contains 8 accents. We cluster each accent embeddings into 32-embedding clusters before merging them into a codebook ending up with 256 embeddings which is denoted as PAM. Here, "persistent" indicates that those 256 accent embeddings in the codebook/memory are not updated during training. The entire process of PAM building can be clarified with the following equations:

$$\begin{aligned} e_i^j &= \text{Concat}(\text{mean}(H_i^j), \text{std}(H_i^j)) \\ E^j &= \{E_1^j, \ldots, E_C^j\} = \text{k-means}(e_1^j, \ldots, e_n^j) \\ E_{\text{PAM}} &= \{E^1, \ldots, E^J\} \end{aligned} \tag{1}$$

where the row of "Concat" means a pooling operation; $H_i^j \in T \times \mathbb{R}^d$ refers to the encoder output of AR model trained with SSLRs for the i-th utterance of accent j, C=32, and J=8 respectively. To realize PAM-based AR, the training is performed to minimize the loss as follows:

$$\mathcal{L}_{AR} = \log p(Y|E_{\text{PAM}}, H_{\text{enc}}) \tag{2}$$

where Y is accent label distribution, and $|Y|$=8 in this work; H_{enc} is the output of the accent classifier encoder. During accent inference, we employ the following rule to recognize the accent:

$$Y_j = \underset{Y_j}{\text{argmax}} \log p(Y_j|E_{\text{PAM}}, H_{\text{enc}}) \tag{3}$$

where $0 \leq j < J$, and J=8 as mentioned.

With the training and testing rules defined in Eq. 2 and Eq. 3, the question is now how to sensibly exploit the PAM in practice.

To fully exploit the PAM to boost the AR performance, we propose two categories of methods, where one is the PAM-based attention fusion method, and the other is an N-best persistent accent memory selection method by similarity measure and we can think of it as a kind of attention variant.

3.2. PAM based attention fusion

The attention-based method includes both self-attention and cross-attention, and the attention is performed either on the frame level or utterance level.

3.2.1. Appending PAM for self-attention fusion

Inspired by [29], we use a linear layer to transform the dimensions of E_{PAM} to be consistent with the encoder output H_{enc}, and append the embeddings of E_{PAM} to the encoder output sequence H_{enc}, and conduct self-attention operation on the overall sequence. The entire process can be interpreted with the following expressions:

$$\begin{aligned} Z &= \text{Stack}(H_{\text{enc}}, E_{\text{PAM}}) \\ \bar{Z} &= \text{MHA}(ZW^Q, ZW^K, ZW^V) \\ \hat{Z} &= \bar{Z} + Z \\ \breve{Z} &= \text{Concat}(\text{mean}(\hat{Z}), \text{std}(\hat{Z})) \\ Y &= \text{Softmax}(\breve{Z}) \end{aligned} \tag{4}$$

where MHA refers to multi-head attention operation, and the row of "Concat" also stands for a pooling operation as in Eq. 1. The equation referred to as Eq. 4 suggests that after appending the PAM embeddings E_{PAM} to the encoder output H_{enc}, we perform self-attention on the combined sequence. The motivation is to bias the encoder output by letting the encoder output be aware of the actual accent context, so as to lead to improved AR performance.

3.2.2. Frame level cross-attention fusion

What is proposed in Section 3.2.1 is a little bit implicit in using the PAM embeddings as accent context, here to take the PAM as an explicit accent context, we employ cross attention similar to what is proposed in [29] for context-aware speech recognition, where the attention is performed on the frame level. In practice, the encoder output H_{enc} is a query, and the embeddings of the PAM acts as key and value; the typical part of the process is as follows:

$$\begin{aligned} \bar{Z} &= \text{MHA}(H_{\text{enc}}W^Q, E_{\text{PAM}}W^K, E_{\text{PAM}}W^V) \\ \hat{Z} &= \bar{Z} + H_{\text{enc}} \end{aligned} \tag{5}$$

3.2.3. Utterance level cross-attention fusion

Instead of biasing the encoder on frame level as Section 3.2.2, we can bias the output of the encoder on utterance level by pooling, which could be much simpler. Besides, since the PAM embeddings are also on utterance level, this realizes all attention components, such as query, key, and value, are on utterance level, making the attention have clear semantic meaning. The utterance level cross-attention is performed as follows:

$$\begin{aligned} Z &= \text{Concat}(\text{mean}(H_{\text{enc}}), \text{std}(H_{\text{enc}})) \\ \bar{Z} &= \text{MHA}(ZW^Q, E_{\text{PAM}}W^K, E_{\text{PAM}}W^V) \\ \hat{Z} &= \bar{Z} + Z \end{aligned} \tag{6}$$

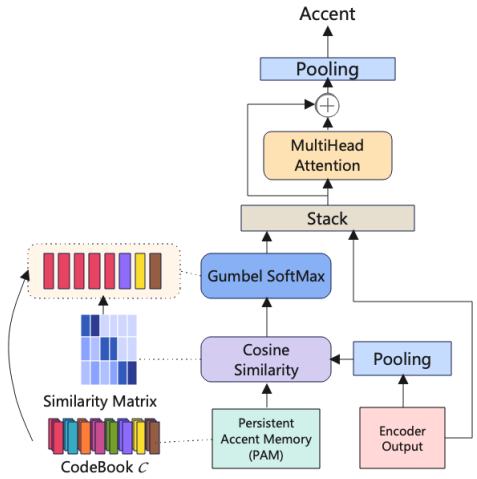

Figure 1: *N-best persistent accent memory selection*

3.3. N-best PAM selection

The similarities of the prior proposed attention methods are simple and straightforward, but there are limitations. For instance, when we attend the PAM, and the overall embeddings in the PAM are equally considered, this would result in redundancy as well as scalable issues as more accents are added. In this section, we propose an N-best based PAM selection method to resolve the problems as mentioned. Here, N denotes the number of embeddings selected from PAM according to the similarity scores between embeddings in PAM and the encoder outputs. The entire architecture of the N-best PAM selection method is illustrated in Figure 1.

For implementation, we first pool encoder output H_{enc} to generate Z on utterance level as in Section 3.2.3, we then calculate the similarity between Z and each embedding in the PAM, and select N-best embeddings according to similarity scores. Finally, we merge such N-best embeddings with the encoder output to perform a self-attention operation as in Section 3.2.1. Specifically, the whole procedure is as follows:

$$\bar{H} = \text{Concat}(\text{mean}(H_{\text{enc}}), \text{std}(H_{\text{enc}}))$$
$$S_i = \text{Cosine-dist}(\bar{H}, E^i_{\text{PAM}}) \quad (7)$$
$$P_i = \text{Gumbel-Softmax}(S_i) \quad (8)$$
$$E^{\text{N-best}}_{\text{PAM}} = \text{topk}(\{P_i\}) * E_{\text{PAM}} \quad (9)$$
$$Z = \text{Stack}(H_{\text{enc}}, E^{\text{N-best}}_{\text{PAM}})$$

where Cosine-dist stands for cosine distance (similarity) estimation, and Gumbel-Softmax refers to Gumbel softmax that is implemented as follows:

$$P_i = \frac{\exp(\log(S_i + g_i)/\tau)}{\sum_{j=1}^{J} \exp(\log(S_j + g_j)/\tau)} \quad (10)$$

where g_i is a random number subjective to 0-1 Gaussian distribution, and τ is a temperature hyper-parameter [33].

The N-best PAM selection method is actually an "attention" variant, since we can also use cross-attention to estimate similarity scores between Z and the embeddings of PAM, and they are attention factors that are estimated with a normal softmax operation. However, we found the conventional attention yields distribution that is much sharper and close to a one-hot vector. It does not facilitate N-best selection by the Gumbel-Softmax.

4. Experiments And Results

4.1. Data Description

In this paper, we conduct all of our experiments on the Accented English Speech Recognition Challenge (AESRC2020) dataset [32], which is also known as a benchmark for the English AR task since 2020. This dataset contains 8 accents of English, which are American(US), British(UK), Chinese(CHN), Indian(IND), Japanese(JPN), Korean(KR), Portuguese(PT), and Russian(RU) accent respectively. The duration of each accent is balanced among the dataset.

4.2. Experimental setup

Our AR models are performed using the Transformer framework based on ESPnet toolkit [34]. We use either 83-dim Fbank-pitch features or 1024-dim representations extracted from WavLM *Large* [18] to build our AR models. Following the work in [17], the AR models are configured with a 12-layer encoder and a 6-layer decoder with 4-head attention in a multi-task way where the ASR task is utilized as an auxiliary task. The weighting factors for ASR and AR tasks are set to 1 and 0.1 respectively. The number of embedding clusters for each accent is 32 and the temperature of Gumbel-softmax is set to 2.0 for the AR task. Besides, SpecAugment [35] is employed at the front-end.

4.3. Results

Table 1 presents the AR accuracy results on the test set using the SSLRs of the WavLM as features. The model trained on SSLRs achieves significant performance improvement compared with using filter-bank (Fbank) features [17], except for the case of which SSLRs are output from the 8th layer of the WavLM encoder, that is, 69.9% being the worst. Besides, we have examined using output either from a different single layer or from the combined layers of the WavLM encoder. The performance varies a little bit for different outputs. For instance, layer-12 yields the best accuracy for US, UK, and KR accents, while layer-20 yields the best performance, i.e., 79.8% on average and the best on accent JPN, i.e., 69.0%, and the final layer, i.e., layer-24, gains the best accuracy on the three accents, that is CHN, IND, and RU. For the combined case, namely, the weighted sum, of which the weight is learned during model training, the performance is moderate.

To validate the efficacy and generality of our proposed methods, we perform 3 categories of experiments: one is "Oracle", the embeddings of each accent are extracted from the corresponding best-performance AR model in Table 1 to build PAM; the other two are based on the final layer output and the overall weighted sum output, denoted as "layer-24" and "layers:1-24" respectively. Particularly, we use the weighted sum "layers:1-24" as the benchmark.

Table 2 reports the AR results of all the proposed methods in Section 3 and the corresponding variants for further clarification. The proposed methods in Section 3.2.1 is denoted as "Append-SA" for PAM appending and mixed self-attention work; likewise, the work in Section 3.2.2 is denoted as "Frame-CA" for frame level based cross-attention; Section 3.2.3 work is named as "Utt-CA" for utterance level based cross-attention; for further clarity, we also perform the combined approach, called as "Utt-CA+Append-SA", that is, we first do utterance-level cross-attention and then perform appending followed by mixed

Table 1: *AR accuracy (%) using SSLRs from WavLM on the test set.*

System	Input Features	-	SSLR source	Average accuracy	US	UK	CHN	IND	JPN	KR	PT	RU
0	Fbank	Baseline	-	73.7	52.4	91.6	72.8	90.9	63.6	77.3	77.6	71.0
1	WavLM SSLR	Single layer	layer-8	69.9	66.4	90.7	64.3	85.7	66.1	64.4	68.1	57.3
2			layer-12	78.3	**71.6**	**95.4**	63.8	92.8	67.3	**82.5**	81.3	76.1
3			layer-16	79.3	67.9	94.4	77.8	94.5	63.7	80.9	**81.9**	78.7
4			layer-20	**79.8**	69.0	93.0	78.5	94.6	**69.0**	81.1	80.5	76.9
5			layer-24	77.3	61.6	94.8	**80.1**	**94.6**	59.5	77.6	76.5	**80.0**
6		Weighted Sum	layers:1-24	77.5	63.3	93.7	78.8	95.0	63.0	82.0	65.0	85.2
7			layers:8,12,16,20,24	78.5	66.3	92.4	79.0	96.1	58.3	77.4	80.2	83.7

Table 2: *AR accuracy results (%) of the proposed methods using persistent accent memory on the test set.*

System	Methods		PAM source scheme	Average accuracy	US	UK	CHN	IND	JPN	KR	PT	RU
6	Weighted Sum (Baseline)		-	77.5	63.3	93.7	78.8	95.0	63.0	82.0	65.0	85.2
8	Frame-CA		Oracle	77.6	65.9	91.8	76.3	95.7	61.8	80.8	72.0	82.3
9	Utt-CA			78.6	58.7	94.1	82.4	96.0	71.3	73.4	78.0	82.1
10	Append-SA			78.9	76.7	91.1	70.0	97.5	69.5	68.8	79.2	81.9
11	Utt-CA+Append-SA			79.9	61.2	94.4	78.6	**96.2**	**71.4**	77.9	79.5	**87.1**
12	N-best PAM selection	AW-similarity	Oracle	79.5	63.0	94.1	73.2	96.4	68.8	81.2	83.1	83.0
13			layer-24	80.9	65.4	93.9	75.9	96.7	74.2	84.1	78.3	85.0
14		Cosine Similarity	layers:1-24	80.9	64.0	94.6	77.2	97.1	72.2	82.4	80.0	86.1
15			Oracle	**81.4**	**66.0**	93.2	82.4	94.5	72.1	83.3	78.4	87.2

Table 3: *Effect of N in the N-best PAM selection method*

N (of N-best)	8	16	32	64	128
Accuracy (%)	81.4	80.9	80.4	**82.0**	81.5

self-attention work. More importantly, the N-best method has two approaches to estimate the similarity: one is based on cosine distance, and the other is the attention-weight-based similarity estimation.

From Table 2, we notice all proposed methods get improved results over the baseline. More interestingly, the methods Utt-CA, Append-SA, and Utt-CA+Append-SA get obviously better results on IND, JPN, and RU accents, compared with what is shown as the best in Table 1. Additionally, the Utt-CA+Append-SA surpasses the best method with the SSLR from layer-20 in terms of on average accuracy across the overall 8 accents. Moreover, the N-best PAM selection methods achieve the best performance, and they are 80.9%, 80.9%, and 81.4% from an average perspective. Under the oracle scenario, the N-best PAM is the best performer, getting 81.4% accuracy. However, naively taking the final layer output (layer-24) and the one with the overall weighted sum setting (layers:1-24), the N-best method still outperforms and is just slightly worse than the oracle method, that is 80.9% versus 81.4% on average. Finally, it is worthwhile to mention that we also attempt the "AW-similarity" method to estimate the similarity between embeddings, and we find the attention weights are sharply distributed (biased to a couple of top embeddings from the PAM), making it harder for us to perform the N-best selection. We hypothesize such a sharp distribution results in a suboptimal update of model parameters and yields worse results than that of cosine distance measure.

In addition, we investigate the effect of different configurations for the N-best select method and present the comparison between different N values in Table 3. It is not surprising that the proposed N-best PAM exhibits higher performance than the baseline approach. The proposed model with $N = 64$ shows the highest accuracy among all model configurations. However, a higher N does not necessarily yield higher performance while it leads to higher computational complexity. This implies that preliminary experiments could be useful to determine an appropriate value of N.

5. Conclusion

In this work, we incorporated self-supervised learning representations (SSLRs) in our proposed persistent accent memory (PAM) method to improve AR. We employed SSLRs extracted from a pre-trained WavLM model to address the data insufficiency problem in the accent recognition task. The use of SSLRs shows significant performance improvement compared to traditional acoustic features, which indicates the efficacy of SSLRs in accent recognition. In addition, we proposed a PAM approach with various attention mechanisms to improve accent recognition. We demonstrated the effectiveness of our proposed method on a public accent benchmark dataset, and the best-performing system that selects the N-best relevant embeddings from the persistent accent memory has achieved further improvements for accent recognition.

6. Acknowledgements

This work was supported by the National Key R&D Program of China (2020AAA0107902).

7. References

[1] S. T. Arasteh, "Generalized LSTM-based end-to-end text-independent speaker verification," *arXiv preprint arXiv:2011.04896*, 2020.

[2] Y. Shi, M. Chen, Q. Huang, and T. Hain, "T-vectors: Weakly supervised speaker identification using hierarchical transformer model," *arXiv preprint arXiv:2010.16071*, 2020.

[3] S. Shon, H. Tang, and J. Glass, "Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model," in *Proc. SLT 2018*. IEEE, 2018, pp. 1007–1013.

[4] G. I. Winata, S. Cahyawijaya, Z. Liu, Z. Lin, A. Madotto, P. Xu, and P. Fung, "Learning fast adaptation on cross-accented speech recognition," *arXiv preprint arXiv:2003.01901*, 2020.

[5] Y. Peng, J. Zhang, H. Xu, H. Huang, and E. S. Chng, "Minimum word error training for non-autoregressive transformer-based code-switching asr," in *Proc. ICASSP 2022*. IEEE, 2022, pp. 7807–7811.

[6] Y. Yang, H. Xu, H. Huang, E. S. Chng, and S. Li, "Speech-text based multi-modal training with bidirectional attention for improved speech recognition," *arXiv preprint arXiv:2211.00325*, 2022.

[7] G. Ma, P. Hu, J. Kang, S. Huang, and H. Huang, "Leveraging phone mask training for phonetic-reduction-robust e2e uyghur speech recognition," *arXiv preprint arXiv:2204.00819*, 2022.

[8] X. Gong, Y. Lu, Z. Zhou, and Y. Qian, "Layer-Wise Fast Adaptation for End-to-End Multi-Accent Speech Recognition," in *Proc. INTERSPEECH 2021*, 2021, pp. 1274–1278.

[9] N. Das, S. Bodapati, M. Sunkara, S. Srinivasan, and D. H. Chau, "Best of Both Worlds: Robust Accented Speech Recognition with Adversarial Transfer Learning," in *Proc. INTERSPEECH 2021*, 2021, pp. 1314–1318.

[10] K. Deng, S. Cao, and L. Ma, "Improving Accent Identification and Accented Speech Recognition Under a Framework of Self-Supervised Learning," in *Proc. INTERSPEECH 2021*, 2021, pp. 1504–1508.

[11] H. Liu, L. P. García-Perera, X. Zhang, J. Dauwels, A. W. Khong, S. Khudanpur, and S. J. Styles, "End-to-end language diarization for bilingual code-switching speech." in *Proc. INTERSPEECH 2021*, 2021.

[12] H. Liu, L. P. Garcia Perera, A. Khong, S. Styles, and S. Khudanpur, "PHO-LID: A Unified Model Incorporating Acoustic-Phonetic and Phonotactic Information for Language Identification," in *Proc. INTERSPEECH 2022*, 2022.

[13] H. Huang, X. Xiang, Y. Yang, R. Ma, and Y. Qian, "Aispeech-sjtu accent identification system for the accented english speech recognition challenge," in *Proc. ICASSP 2021*. IEEE, 2021, pp. 6254–6258.

[14] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[15] B. Raj, M. L. Seltzer, and R. M. Stern, "Reconstruction of missing features for robust speech recognition," *Speech communication*, vol. 43, no. 4, pp. 275–296, 2004.

[16] Q. Shao, J. Yan, J. Kang, P. Guo, X. Shi, P. Hu, and L. Xie, "Linguistic-Acoustic Similarity Based Accent Shift for Accent Recognition," in *Proc. INTERSPEECH 2022*, 2022, pp. 3719–3723.

[17] J. Z., Y. P., V. T. P., H. X., H. H., and E. S. C., "E2E-Based Multi-Task Learning Approach to Joint Speech and Accent Recognition," in *Proc. INTERSPEECH 2021*, 2021, pp. 1519–1523.

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

[19] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[20] 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.

[21] 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," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[22] S. Chen, Y. Wu, C. Wang, S. Liu, Z. Chen, P. Wang, G. Liu, J. Li, J. Wu, X. Yu *et al.*, "Why does self-supervised learning for speech recognition benefit speaker recognition?" *arXiv preprint arXiv:2204.12765*, 2022.

[23] Z. Chen, N. Kanda, J. Wu, Y. Wu, X. Wang, T. Yoshioka, J. Li, S. Sivasankaran, and S. E. Eskimez, "Speech separation with large-scale self-supervised learning," *arXiv preprint arXiv:2211.05172*, 2022.

[24] 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.

[25] A. Wuerkaixi, K. Yan, Y. Zhang, Z. Duan, and C. Zhang, "Dyvise: Dynamic vision-guided speaker embedding for audio-visual speaker diarization," in *Proc. MMSP 2022*. IEEE, 2022, pp. 1–6.

[26] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Proc. INTERSPEECH 2021*, 2021, pp. 2426–2430.

[27] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. INTERSPEECH 2021*, 2021, pp. 1194–1198.

[28] H. Liu, L. P. G. Perera, A. W. Khong, E. S. Chng, S. J. Styles, and S. Khudanpur, "Efficient self-supervised learning representations for spoken language identification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1296–1307, 2022.

[29] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: end-to-end contextual speech recognition," in *Proc. SLT 2018*. IEEE, 2018, pp. 418–425.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[31] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. CVPR 2018*, 2018, pp. 3733–3742.

[32] X. Shi, F. Yu, Y. Lu, Y. Liang, Q. Feng, D. Wang, Y. Qian, and L. Xie, "The accented english speech recognition challenge 2020: open datasets, tracks, baselines, results and methods," in *Proc. ICASSP 2021*. IEEE, 2021, pp. 6918–6922.

[33] E. Jang, S. Gu, and B. Poole, "Categorical reparametrization with gumble-softmax," in *Proc. ICLR 2017*. OpenReview. net, 2017.

[34] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. INTERSPEECH 2018*, 2018, pp. 2207–2211.

[35] D. S. Park, W. C., Y. Z., C.-C. C., B. Z., E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. INTERSPEECH 2019*, 2019, pp. 2613–2617.