



Weakly-Supervised Speech Pre-training: A Case Study on Target Speech Recognition

Wangyou Zhang, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, China

wyz-97@sjtu.edu.cn, yanminqian@sjtu.edu.cn

Abstract

Self-supervised learning (SSL) based speech pre-training has attracted much attention for its capability of extracting rich representations learned from massive unlabeled data. On the other hand, the use of weakly-supervised data is less explored for speech pre-training. To fill this gap, we propose a weakly-supervised speech pre-training method based on speaker-aware speech data. It adopts a similar training procedure to the widely-used masked speech prediction based SSL framework, while incorporating additional target-speaker enrollment information as an auxiliary input. In this way, the learned representation is steered towards the target speaker even in the presence of highly overlapping interference, allowing potential applications to tasks such as target speech recognition. Our experiments on Libri2Mix and WSJ0-2mix datasets show that the proposed model achieves significantly better ASR performance compared to WavLM, the state-of-the-art SSL model with denoising capability.

Index Terms: weakly-supervised learning, self-supervised learning, speech pre-training, target speech recognition

1. Introduction

Recently, self-supervised learning (SSL) based pre-training has greatly advanced research progress in speech processing, showing great potential in a wide range of downstream speech tasks [1, 2]. Existing SSL models can be roughly grouped into three categories based on their pre-training objectives, i.e., generative [3, 4, 5, 6], contrastive [7, 8, 9, 10], and predictive [11, 12, 13, 14] approaches. These SSL models are usually pre-trained on massive *unlabeled* data in an application-agnostic manner [15], and then fine-tuned on downstream speech tasks by updating either the entire network or only a small amount of parameters [16, 17]. The learned representations are found to be versatile for a series of speech tasks such as automatic speech recognition (ASR) [18], text-to-speech (TTS) [19], speaker verification (SV) [20], speech enhancement (SE) [21], and so on.

While existing SSL models are highly effective in extracting rich representations from single-speaker utterances, the capability of eliminating the interference from overlapped speech is still limited [12, 21]. However, this capability is especially important when tackling the well-known cocktail party problem [22, 23], where multiple talkers speak simultaneously in a noisy environment. It is therefore natural to ask whether we can improve the current speech pre-training paradigm to take the above problem into consideration. Recently, there are a few studies working on this direction. Chen *et al.* [12] proposes to augment the input speech by overlapping with background speech or noise to force the SSL model to learn masked

speech denoising and prediction at the same time. Wang *et al.* [17] proposes to explicitly predict multiple labels corresponding to all utterances in the input overlapped speech, enabling the SSL model to learn denoising and separation in parallel with masked speech prediction. These efforts still fall within the self-supervised speech pre-training framework, where no additional information other than the speech itself is exploited.

In this paper, we aim to explore a new speech pre-training direction, namely weakly-supervised speech pre-training, which allows the use of additional information (weak labels) about the data to facilitate better speech pre-training. While similar ideas have been explored in computer vision and natural language processing [24, 25], it has not been well studied in the area of speech processing. Our motivation is that the collected speech data for pre-training may be annotated with meta information such as the relative speaker identity. While SSL methods generally omit such information during pre-training, it can be helpful for the related downstream tasks if we take advantage of it properly during pre-training.

Therefore, in this paper, we make a first attempt at weakly-supervised speech pre-training using speaker-aware data. The only assumption is that all speech samples can be divided into S groups, each corresponding to a different speaker. The speaker division can be obtained from either clustering [26] or the data annotation¹. Based on this assumption, we present a novel speech pre-training model—Target-Speaker HuBERT, or TS-HuBERT for short. It adopts the same masked speech prediction objective as proposed in HuBERT [11], a widely-used SSL model. During pre-training, the main input speech is randomly mixed with speech from a different speaker, and an auxiliary speech sample from the same main speaker is provided to guide the modeling of the main speaker’s speech. Taking both waveforms as input, the proposed pre-training model learns to predict the discrete targets of masked frames while eliminating the interference in the overlapped speech. Furthermore, we investigate the effectiveness of the proposed TS-HuBERT model in the downstream target speech recognition task, which aims to recognize the target speaker’s speech in the overlapped speech. Our experiments show that the proposed model achieves significantly better ASR performance compared to WavLM [12], the state-of-the-art SSL model with denoising capability.

2. Weakly-supervised speech pre-training

2.1. Model design

Figure 1 presents the overview of the proposed TS-HuBERT model. The entire model is built upon the well-established architecture in [9, 11], which consists of a convolutional neural

[†]Corresponding author.

¹In this paper, we adopt the data annotation to simplify the discussion, and leave the former for future investigation.

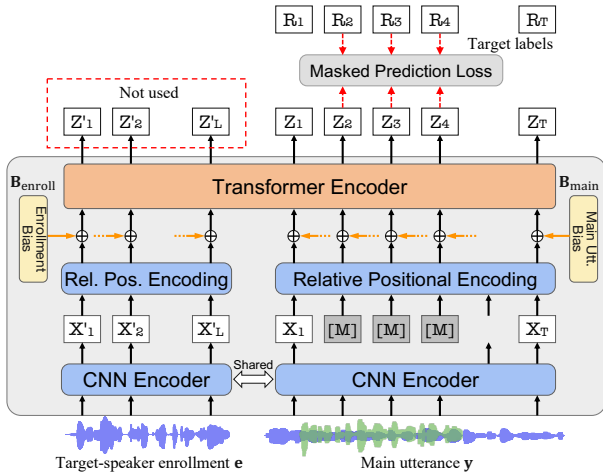


Figure 1: Overview of the proposed weakly-supervised speech pre-training model based on speaker-aware speech data.

network (CNN) encoder followed by a Transformer encoder. In addition, following WavLM [12], the gated relative position bias [27] is also employed in the self-attention mechanism in Transformer encoder layers to boost the performance. The input to the proposed model consists of a main utterance y for masked speech prediction [11] and an auxiliary input e containing information about the target speaker. The details of the pre-training data are presented in Section 2.3.

Now, we consider integrating the target-speaker information with the aforementioned backbone. In the literature, two types of data are often used to provide such information, i.e., an enrollment audio from the target speaker [28] and a speaker embedding vector [29]. While the latter can directly provide a compact and efficient target-speaker representation, it entangles the pre-training model with a specific external speaker embedding model. This inevitably reduces the generalizability and flexibility of the speech pre-training model. Therefore, we opt for the raw enrollment audio as the auxiliary input and rely on the pre-training model to extract the target-speaker information implicitly. Inspired by the success of temporal concatenation-based cross-modal modeling [30, 31, 32, 33], we propose to fuse the two input streams via temporal concatenation. More specifically, we first extract features for the enrollment e and the main utterance y using the same CNN encoder:

$$\mathbf{X} = \text{CNN-Encoder}(\mathbf{y}) \in \mathbb{R}^{T \times D}, \quad (1)$$

$$\mathbf{X}' = \text{CNN-Encoder}(\mathbf{e}) \in \mathbb{R}^{L \times D}, \quad (2)$$

where \mathbf{X} and \mathbf{X}' are the main utterance feature and the enrollment feature, respectively. T and L are the corresponding numbers of feature frames, and D is the feature dimension. Later, we will concatenate the two features along the temporal dimension and leverage the Transformer encoder to learn the implicit correlation between them. However, a simple concatenation of the two streams makes it difficult to distinguish them in the self-attention mechanism. To mitigate this issue, we apply two independent convolution-based relative position encoding (rPE) layers [34, 9] to inject the temporal order information into both streams. In addition, we further enlarge the difference between the two sequences by adding a learnable bias vector to each of them, which is akin to the modality encoding in [32]:

$$\mathbf{X}_{\text{in}} = \text{Conv}_{\text{rel_pos}}(\mathbf{X}) + \mathbf{B}_{\text{main}}, \quad (3)$$

$$\mathbf{X}'_{\text{in}} = \text{Conv}'_{\text{rel_pos}}(\mathbf{X}') + \mathbf{B}_{\text{enroll}}, \quad (4)$$

where $\mathbf{B}_{\text{main}} \in \mathbb{R}^D$ and $\mathbf{B}_{\text{enroll}} \in \mathbb{R}^D$ denote the main utterance

bias vector and the enrollment bias vector, respectively. The updated features \mathbf{X}_{in} and \mathbf{X}'_{in} are finally concatenated and fed into the Transformer encoder to predict the frame-wise labels corresponding to the main utterance. The output labels corresponding to the enrollment frames are discarded via slicing.

2.2. Speaker-aware masked speech prediction

In this subsection, we introduce the training objective of the proposed pre-training method, which follows the design in the HuBERT model [11]. The basic idea of the masked speech prediction objective in HuBERT is to predict the frame-wise discrete labels of the input speech while masking a portion of the feature frames generated by the CNN encoder. The prediction loss (cross-entropy loss) is only calculated for the masked frames, while the labels are generated by one or more iterations of K-means clustering on the feature of the input speech. We refer to the original paper [11] for the details about label generation.

Since our major target is to improve the interference elimination ability of the speech pre-training model via weak supervision, we extend the original objective in HuBERT to fit this goal. Given a main utterance input y and a target-speaker enrollment e , we only apply masking to the main utterance feature \mathbf{X} obtained in Eq. (1) and leave the enrollment feature \mathbf{X}' intact. During pre-training, we also calculate the cross entropy loss only for the main utterance input, whose label is prepared in advance as in HuBERT training. This is to encourage the model to focus on target speaker extraction with the full enrollment information. Furthermore, to take the cocktail party problem into account, we augment the main utterance y by mixing it with a randomly sampled utterance from another speaker. The detailed procedure is described in Section 2.3.

In our implementation, the masks have a fixed length of 10 frames, and the number of masks is proportional to the main utterance length. The maximum percentage of masked frames is 80% and is usually not reached due to mask overlaps. The masked feature frames are simply replaced with zeros.

2.3. Pre-training data preparation

Similar to WavLM [12], we propose a speaker-aware utterance mixing strategy to simulate the overlapped pre-training data on the fly. The core algorithm is illustrated in Algorithm 1. The main differences compared to WavLM are: (1) We sample the interference speech from the entire database instead of the current batch. (2) We set the range of the overlap ratio to $[0, 100\%]$ instead of $[0, 50\%]$. (3) An additional utterance e from the same main speaker is added to provide the target-speaker information.

2.4. Application: target speech recognition

In this subsection, we introduce several fine-tuning methods to apply the proposed TS-HuBERT model to the downstream target speech recognition task. It aims to recognize the target speaker's speech in the overlapped speech, which is a typical task in the cocktail party problem. One straightforward fine-tuning approach is using a linear projection layer on top of the pre-training model to map the feature to the output dimensionality (vocabulary size) [9, 11]. And the connectionist temporal classification (CTC) loss [35] is used for end-to-end training. In this procedure, the entire pre-training model (except for the CNN encoder) will be updated to fit the downstream task. Since TS-HuBERT can extract the target-speaker information from the enrollment, it naturally fits this fine-tuning method.

For SSL models that cannot utilize the enrollment directly, another adaptation-based fine-tuning approach [36] can be used, where lightweight speaker adaptation layers are inserted into the

Algorithm 1 Speaker-aware utterance mixing strategy

Input: an inventory of speaker-aware speech grouped by speaker IDs: \mathbf{G} ;
a list of speaker IDs for all speech samples: \mathbf{Q} ;
a list of pre-training labels for all speech samples: \mathbf{R} ;
batch size: i ;

Output: a batch of pre-training data: \mathbf{U} .

```
1:  $\mathbf{U} = \{\}$ 
2: Sample  $i$  utterances  $\mathbf{U}_{\text{main}}$  uniformly from  $\mathbf{G}$ 
3: for each main utterance  $\mathbf{y} \in \mathbf{U}_{\text{main}}$  do
4:    $q_{\text{main}} \leftarrow \mathbf{Q}[\mathbf{y}]$   $\triangleright$  Find the speaker ID of  $\mathbf{y}$ 
5:   Sample a speaker ID  $q (\neq q_{\text{main}})$  from  $\mathbf{Q}$ 
6:   Sample an utterance  $\mathbf{u}_{\text{interf}}$  from  $\mathbf{G}[q] \triangleright$  Utts of speaker  $q$ 
7:   Sample the mixing energy ratio  $k \sim \mathcal{U}(-5, 5)$ 
8:   Rescale  $\mathbf{u}_{\text{interf}}$  such that  $10 \log_{10} \frac{\|\mathbf{u}_{\text{main}}\|^2}{\|\mathbf{u}_{\text{interf}}\|^2} = k$ 
9:    $M \leftarrow \text{length}(\mathbf{y}); N \leftarrow \text{length}(\mathbf{u}_{\text{interf}})$ 
10:  Sample the mixing length  $l$  from  $\{1, 2, \dots, M\}$ 
11:   $l \leftarrow \min(l, N)$ 
12:  Sample the start position  $m$  from  $\{0, 1, \dots, M-l\}$ 
13:  Sample the start position  $n$  from  $\{0, 1, \dots, N-l\}$ 
14:   $\mathbf{y}[m:m+l] \leftarrow \mathbf{y}[m:m+l] + \mathbf{u}_{\text{interf}}[n:n+l]$ 
15:  Sample an enrollment  $\mathbf{e} (\neq \mathbf{y})$  from  $\mathbf{G}[q_{\text{main}}]$ 
16:   $\mathbf{r} \leftarrow \mathbf{R}[\mathbf{y}]$   $\triangleright$  Find the label of  $\mathbf{y}$ 
17:   $\mathbf{U} \leftarrow \mathbf{U} \cup \{(\mathbf{y}, \mathbf{e}, \mathbf{r})\}$   $\triangleright$  Append a new sample
18: end for
19: return  $\mathbf{U}$ 
```

pre-training model for joint fine-tuning. The newly inserted layers take as input a pre-extracted speaker embedding vector \mathbf{e}_{emb} to steer the intermediate representations in pre-training models towards the target speaker. Here, we evaluate the following three adaptation layers proposed in [36] with TS-HuBERT:

1. **Add:** \mathbf{e}_{emb} is directly added to the CNN encoder output \mathbf{X} through a linear projection to match the hidden dimension.
2. **FiLM:** \mathbf{e}_{emb} is used to estimate a feature-wise linear modulation (FiLM) [37] for the CNN output \mathbf{X} :

$$\mathbf{X} \leftarrow w(\mathbf{e}_{\text{emb}}) \cdot \mathbf{X} + b(\mathbf{e}_{\text{emb}}), \quad (5)$$

where $w(\cdot)$ and $b(\cdot)$ are two linear projection layers.

3. **cLN:** \mathbf{e}_{emb} is used to estimate FiLM transformations for the layer normalizations (LNs) in the first Transformer encoder layer, converting them into conditional LNs (cLNs) [38]:

$$\mathbf{X}_{\text{cLN}} = [w(\mathbf{e}_{\text{emb}}) \cdot \gamma + b(\mathbf{e}_{\text{emb}})] \cdot \frac{\mathbf{X} - \boldsymbol{\mu}}{\boldsymbol{\sigma}} + \boldsymbol{\beta}, \quad (6)$$

where γ and β are the learnable scaling and biasing parameters in standard LNs, respectively. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the mean and standard deviation of the input \mathbf{X} , respectively.

We apply the near-identity initialization [39] to all adaptation layers, which is found crucial in our preliminary experiments.

3. Experiments

3.1. Experimental setup

Our experiments were done using the fairseq toolkit². The TS-HuBERT model follows the same structure as the WavLM Base model, which enhances the HuBERT [11] Transformer encoder by employing the gated relative position bias [27] in the self-attention mechanism. Due to the space limitation, we refer to the WavLM paper [12] for the detailed parameters. As described in Section 2.1, we additionally insert two convolution-based rPE layers and two 768-dimensional learnable bias vectors before the Transformer encoder. Each rPE layer consists of a 16-group convolution layer with kernel size 128 and the

²github.com/facebookresearch/fairseq (313ff05)

Gaussian error linear unit (GELU) activation. These new layers account for 10M new parameters, resulting in 104.37M parameters in total. The TS-HuBERT model is pre-trained for one iteration (400k steps) on the 960-hour Librispeech data [40] using labels generated by 500-centroid K-means clustering of the 9-th Transformer layer output of the HuBERT Base model³. The batch size is at most 312 seconds of audio per GPU. Other hyperparameters are the same as those in HuBERT Base [11]. The enrollment \mathbf{e} is randomly truncated to 48000 samples to avoid massive memory consumption during pre-training.

As for fine-tuning on downstream tasks, we use the character-level CTC loss for all experiments. We evaluate the performance of target speech recognition on two commonly-used datasets: noisy Libri2Mix [41] and WSJ0-2mix [42]. The sample rate of all speech data is 16 kHz. In Libri2Mix, there are 13900, 3000, and 3000 samples in the training⁴, validation, and evaluation sets, respectively. In WSJ0-2mix, the numbers of samples are 20000, 5000, and 3000, respectively. The sets of speakers in training and evaluation sets do not overlap. We adopt the existing enrollment lists for samples in the validation and evaluation sets of Libri2Mix⁵ and WSJ0-2mix⁶, respectively. For the training sets, we randomly select the same-speaker enrollment for each sample to increase diversity. The batch size for fine-tuning is 125 seconds of audio per GPU. The peak learning rate is 2e-5 with 8000 warmup steps. During fine-tuning, we always freeze the CNN encoder in the pre-training model. For adaptation-based fine-tuning, we use a ResNet-34 model [43, 44] pre-trained on the VoxCeleb corpus [45] to extract 256-dimensional speaker embeddings \mathbf{e}_{emb} . During evaluation, we adopt the Viterbi algorithm for decoding, and the word error rate (WER) is calculated as the metric. Unless specifically mentioned, no language model is used. We used 8 RTX 2080 Ti GPUs for all our experiments.⁷

3.2. Evaluation on standard speech recognition

Before applying the proposed pre-training model to target speech recognition, we are firstly interested in the performance of the proposed model in the standard ASR task. To this end, we fine-tune the TS-HuBERT model on three different subsets of Librispeech as in [9, 11, 12] and evaluate it on the `test-clean` and `test-other` subsets. We simply follow the fine-tuning configuration as in [11, 12] and use a batch size of 200s of audio per GPU. Since no enrollment is available in the standard ASR task, we discard the enrollment-related processing and only feed the main utterance feature into the Transformer encoder (denoted as “w/o \mathbf{e} ”). We compare three speech pre-training models in Table 1, where all models are pre-trained on the 960-hour Librispeech data for 400k steps. Interestingly, although our proposed model is pre-trained always with an additional enrollment input, it can still be fine-tuned to achieve comparable performance compared to the two SSL models. This reveals the potential of the proposed TS-HuBERT model to adapt to different tasks.

3.3. Evaluation on target speech recognition

Next, we evaluate the performance of the speech pre-training models on the target speech recognition task. We first conduct

³Available at https://dl.fbaipublicfiles.com/HuBERT/HuBERT_base_ls960.pt

⁴We only use the `train-100` subset to speed up experiments.

⁵<https://github.com/BUTSpeechFIT/speakerbeam>

⁶https://github.com/gemengtju/SpEx_Plus

⁷Code is available at <https://github.com/Emrys365/fairseq/tree/wavlm/examples/tshubert>.

Table 1: WER (%) evaluation of speech pre-training models for standard speech recognition on the LibriSpeech test sets. Note that no language model is used for the listed models.

| Method | test-clean | test-other |
|-------------------------|-------------|-------------|
| 1-hour labeled | | |
| wav2vec 2.0 Base [12] | 24.5 | 29.7 |
| WavLM Base [12] | 24.5 | 29.2 |
| TS-HuBERT (w/o e) | 20.8 | 28.1 |
| 10-hour labeled | | |
| wav2vec 2.0 Base [12] | 11.1 | 17.6 |
| WavLM Base [12] | 9.8 | 16.0 |
| TS-HuBERT (w/o e) | 10.7 | 18.9 |
| 100-hour labeled | | |
| wav2vec 2.0 Base [12] | 6.1 | 13.3 |
| WavLM Base [12] | 5.7 | 12.0 |
| TS-HuBERT (w/o e) | 6.0 | 13.9 |

experiments on the WSJ0-2mix dataset. The results are shown in Table 2. We first compare with two state-of-the-art multi-speaker ASR methods on WSJ0-2mix, i.e., the jointly trained speech separation and ASR models [46] (denoted as “DPRNN-ASR”), and a monolithic multi-speaker ASR model [18] (denoted as “PIT-ASR”). For the SSL-based method, we compare with the WavLM Base model as it is the state-of-the-art SSL model with denoising capability and has a similar pre-training setup to ours. The same adaptation-based fine-tuning approaches as described in Section 2.4 are applied to the WavLM Base model to enable target speech recognition.

From Table 2, we can see that directly fine-tuning TS-HuBERT without adaptation layers can readily achieve very promising results (WER<7%). The adaptation-based fine-tuning methods also work well with both WavLM Base and TS-HuBERT, and increasing the number of fine-tuning steps can slightly improve the performance. In [11, 12], the Transformer encoder in the pre-training model is often frozen for the first 10k fine-tuning steps. Here, we empirically find this trick is harmful to the target speech recognition performance, and it is thus disabled for the rest experiments. It is noteworthy that our proposed TS-HuBERT model obtains the new state-of-the-art performance (WER=6.1%) on WSJ0-2mix, which is significantly better than the WavLM Base model.

Furthermore, we evaluate the models on the Libri2Mix dataset, which is much more difficult due to the presence of noise.⁸ Due to the space limitation, we only present the best results for the adaptation-based fine-tuning methods. In addition, we compare with the monolithic multi-speaker ASR method (denoted as “PIT-ASR”) from ESPnet [47], and enhance it by adding more training data (denoted as “+ train-360”) and by further applying speed perturbation (SP) [48] and a Transformer language model (LM). Although the pre-training-based models only use the train-100 subset for fine-tuning, they obtain substantial performance improvement over the PIT-ASR approach, even when the latter uses much more training data (“+ train-360”). The proposed TS-HuBERT again surpasses the WavLM Base model, with ~10% relative WER reduction. However, their performance still lags behind the doubly enhanced PIT-ASR approach (“++ SP & LM”), implying room for further improvement in the noisy scenario.

⁸Differently, [36] uses the clean version of Libri2Mix.

Table 2: Evaluation of different speech pre-training models for target speech recognition on the WSJ0-2mix test set. “#Param” denotes the number of trainable parameters. “Frz. 10k” denotes whether or not to freeze the pre-training model for the first 10k steps. “50k” and “100k” denote the fine-tuning steps.

| Method | Adapt. Type | #Param | Frz. 10k | WER (%) | |
|------------------------|-------------|----------|----------|------------|------------|
| | | | | 50k | 100k |
| DPRNN-ASR (8 kHz) [46] | - | - | - | 7.1 | |
| PIT-ASR [18] | - | - | - | 12.1 | |
| WavLM Base | Add | 94.75 M | ✓ | 16.6 | 14.6 |
| | | | × | 11.7 | 11.3 |
| | FiLM | 94.95 M | ✓ | 15.8 | 13.9 |
| | | | × | 11.7 | 11.2 |
| | cLN | 95.34 M | ✓ | 13.0 | 12.4 |
| | | | × | 10.9 | 10.4 |
| TS-HuBERT | - | 104.39 M | ✓ | 6.9 | 6.5 |
| | | | × | 6.3 | 6.1 |
| | Add | 104.59 M | ✓ | 7.4 | 6.8 |
| | | | × | 6.2 | 6.1 |
| | FiLM | 104.78 M | ✓ | 7.0 | 6.5 |
| | | | × | 6.3 | 6.1 |
| | cLN | 105.18 M | ✓ | 6.8 | 6.3 |
| | | | × | 6.4 | 6.1 |

Table 3: Evaluation of different speech pre-training models for target speech recognition on the noisy Libri2mix test set. “#Param” denotes the number of trainable parameters. “100k” and “250k” denote the fine-tuning steps.

| Method | Adapt. Type | #Param | WER (%) | |
|-------------|-------------|----------|-------------|------|
| | | | 100k | 250k |
| PIT-ASR | - | 32 M | 50.1 | |
| + train-360 | - | 32 M | 31.0 | |
| ++ SP & LM | - | 32 M | 23.5 | |
| WavLM Base | cLN | 95.34 M | 28.5 | 27.5 |
| TS-HuBERT | - | 104.39 M | 25.5 | 24.9 |
| | cLN | 105.18 M | 25.5 | 24.8 |

4. Conclusions

In this paper, we propose a novel weakly-supervised speech pre-training model—TS-HuBERT, which utilizes an additional enrollment to incorporate the target-speaker information. Pre-trained with the masked speech prediction objective on overlapped speech, TS-HuBERT learns to eliminate the interference by leveraging the enrollment information. Experiments show that TS-HuBERT can work well on both standard ASR and target speech recognition tasks. In future work, we would like to extend TS-HuBERT with clustering-based speaker labels for pre-training, thus making it better fit realistic applications.

5. Acknowledgements

The authors would like to thank Zili Huang from Johns Hopkins University for the helpful discussion on adapting SSL models to target speech recognition. This work was supported in part by China STI 2030-Major Projects under Grant No. 2021ZD0201500, in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

6. References

- [1] J. Shor, A. Jansen, R. Maor, O. Lang *et al.*, “Towards learning a universal non-semantic representation of speech,” in *Proc. Inter-*

- speech, 2020, pp. 140–144.
- [2] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai *et al.*, “SUPERB: Speech Processing Universal PERFORMANCE Benchmark,” in *Proc. Interspeech*, 2021, pp. 1194–1198.
 - [3] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, “An unsupervised autoregressive model for speech representation learning,” in *Proc. Interspeech*, 2019, pp. 146–150.
 - [4] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski *et al.*, “Multi-task self-supervised learning for robust speech recognition,” in *Proc. IEEE ICASSP*, 2020, pp. 6989–6993.
 - [5] A. T. Liu, S.-W. Li, and H.-y. Lee, “TERA: Self-supervised learning of transformer encoder representation for speech,” *IEEE/ACM Trans. ASLP*, vol. 29, pp. 2351–2366, 2021.
 - [6] A. H. Liu, Y.-A. Chung, and J. Glass, “Non-autoregressive predictive coding for learning speech representations from local dependencies,” in *Proc. Interspeech*, 2021, pp. 3730–3734.
 - [7] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint: 1807.03748*, 2018.
 - [8] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised pre-training for speech recognition,” in *Proc. Interspeech*, 2019, pp. 3465–3469.
 - [9] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *Proc. NeurIPS*, vol. 33, 2020, pp. 12 449–12 460.
 - [10] Y. Wang, J. Li, H. Wang, Y. Qian *et al.*, “Wav2vec-Switch: Contrastive learning from original-noisy speech pairs for robust speech recognition,” in *Proc. IEEE ICASSP*, 2022, pp. 7097–7101.
 - [11] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia *et al.*, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Trans. ASLP*, vol. 29, pp. 3451–3460, 2021.
 - [12] S. Chen, C. Wang, Z. Chen, Y. Wu *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.
 - [13] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu *et al.*, “data2vec: A general framework for self-supervised learning in speech, vision and language,” in *Proc. ICML*, 2022, pp. 1298–1312.
 - [14] Z. Ma, Z. Zheng, C. Tang, Y. Wang, and X. Chen, “MT4SSL: Boosting self-supervised speech representation learning by integrating multiple targets,” in *Proc. Interspeech*, 2023.
 - [15] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn *et al.*, “Self-supervised speech representation learning: A review,” *IEEE JSTSP*, vol. 16, no. 6, pp. 1179–1210, 2022.
 - [16] Z.-C. Chen, C.-L. Fu, C.-Y. Liu, S.-W. D. Li, and H.-y. Lee, “Exploring efficient-tuning methods in self-supervised speech models,” in *Proc. IEEE SLT*, 2022, pp. 1120–1127.
 - [17] T. Wang, X. Chen, Z. Chen, S. Yu, and W. Zhu, “An adapter based multi-label pre-training for speech separation and enhancement,” in *Proc. IEEE ICASSP*, 2023.
 - [18] X. Chang, T. Maekaku, P. Guo, J. Shi *et al.*, “An exploration of self-supervised pretrained representations for end-to-end speech recognition,” in *Proc. IEEE ASRU*, 2021, pp. 228–235.
 - [19] C. Du, Y. Guo, X. Chen, and K. Yu, “VQTTs: High-fidelity text-to-speech synthesis with self-supervised VQ acoustic feature,” in *Proc. Interspeech*, 2022, pp. 1596–1600.
 - [20] S. Chen, Y. Wu, C. Wang, S. Liu *et al.*, “Why does self-supervised learning for speech recognition benefit speaker recognition?” in *Proc. Interspeech*, 2022, pp. 3699–3703.
 - [21] Z. Huang, S. Watanabe, S.-w. Yang, P. García, and S. Khudanpur, “Investigating self-supervised learning for speech enhancement and separation,” in *Proc. IEEE ICASSP*, 2022, pp. 6837–6841.
 - [22] E. C. Cherry, “Some experiments on the recognition of speech, with one and with two ears,” *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.
 - [23] Y. Qian, C. Weng, X. Chang, S. Wang, and D. Yu, “Past review, current progress, and challenges ahead on the cocktail party problem,” *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 40–63, 2018.
 - [24] D. Ghadyaram, D. Tran, and D. Mahajan, “Large-scale weakly-supervised pre-training for video action recognition,” in *Proc. CVPR*, 2019, pp. 12 046–12 055.
 - [25] W. Xiong, J. Du, W. Y. Wang, and V. Stoyanov, “Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model,” in *Proc. ICLR*, 2020.
 - [26] B. Han, Z. Chen, and Y. Qian, “Self-supervised speaker verification using dynamic loss-gate and label correction,” in *Proc. Interspeech*, 2022, pp. 4780–4784.
 - [27] Z. Chi, S. Huang, L. Dong, S. Ma *et al.*, “XLM-E: Cross-lingual language model pre-training via ELECTRA,” in *Proc. ACL*, 2022, pp. 6170–6182.
 - [28] M. Delcroix, S. Watanabe, T. Ochiai, K. Kinoshita *et al.*, “End-to-end SpeakerBeam for single channel target speech recognition,” in *Proc. Interspeech*, 2019, pp. 451–455.
 - [29] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar *et al.*, “Voice-Filter: Targeted voice separation by speaker-conditioned spectrogram masking,” in *Proc. Interspeech*, 2019, pp. 2728–2732.
 - [30] J. Lin, A. Yang, Y. Zhang, J. Liu *et al.*, “InterBERT: Vision-and-language interaction for multi-modal pretraining,” *arXiv preprint: 2003.13198*, 2020.
 - [31] X. Pan, P. Chen, Y. Gong, H. Zhou *et al.*, “Leveraging unimodal self-supervised learning for multimodal audio-visual speech recognition,” in *Proc. ACL*, 2022, pp. 4491–4503.
 - [32] A. Rahimi, T. Afouras, and A. Zisserman, “Reading to listen at the cocktail party: Multi-modal speech separation,” in *Proc. CVPR*, 2022, pp. 10 493–10 502.
 - [33] Z. Yang, Y. Fang, C. Zhu, R. Pryzant *et al.*, “i-Code: An integrative and composable multimodal learning framework,” *arXiv preprint: 2205.01818*, 2022.
 - [34] A. Mohamed, D. Okhonko, and L. Zettlemoyer, “Transformers with convolutional context for ASR,” *arXiv preprint: 1904.11660*, 2019.
 - [35] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, 2006, pp. 369–376.
 - [36] Z. Huang, D. Raj, P. García, and S. Khudanpur, “Adapting self-supervised models to multi-talker speech recognition using speaker embeddings,” in *Proc. IEEE ICASSP*, 2023.
 - [37] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *Proc. AAAI*, vol. 32, no. 1, 2018.
 - [38] J. Pilault, A. Elhattami, and C. Pal, “Conditionally adaptive multi-task learning: Improving transfer learning in NLP using fewer parameters & less data,” in *Proc. ICLR*, 2021.
 - [39] R. Karimi Mahabadi, S. Ruder, M. Dehghani, and J. Henderson, “Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks,” in *Proc. ACL-IJCNLP*, 2021, pp. 565–576.
 - [40] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in *Proc. IEEE ICASSP*, 2015, pp. 5206–5210.
 - [41] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, “LibriMix: An open-source dataset for generalizable speech separation,” *arXiv preprint: 2005.11262*, 2020.
 - [42] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *Proc. IEEE ICASSP*, 2016, pp. 31–35.
 - [43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. CVPR*, 2016, pp. 770–778.
 - [44] Z. Chen, B. Han, X. Xiang, H. Huang *et al.*, “Build a SRE challenge system: Lessons from VoxSRC 2022 and CNSRC 2022,” *arXiv preprint: 2211.00815*, 2022.
 - [45] A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” in *Proc. Interspeech*, 2017, pp. 2616–2620.
 - [46] J. Shi, X. Chang, S. Watanabe, and B. Xu, “Train from scratch: Single-stage joint training of speech separation and recognition,” *Computer Speech & Language*, vol. 76, p. 101387, 2022.
 - [47] S. Watanabe, T. Hori, S. Karita, T. Hayashi *et al.*, “ESPnet: End-to-end speech processing toolkit,” in *Proc. Interspeech*, 2018, pp. 2207–2211.
 - [48] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, “Audio augmentation for speech recognition,” in *Proc. Interspeech*, 2015, pp. 3586–3589.