IMPROVING SELF-SUPERVISED LEARNING FOR SPEECH RECOGNITION WITH INTERMEDIATE LAYER SUPERVISION

Chengyi Wang[1,2] *, Yu Wu*[2]*, Sanyuan Chen*[2]*, Shujie Liu*[2]*, Jinyu Li*[2]*, Yao Qian*[2]*, Zhenglu Yang*[1]

[1] NanKai University, China, [2]Microsoft Corporation

ABSTRACT

Recently, pioneer work finds that self-supervised pre-training methods can improve multiple downstream speech tasks, because the model utilizes bottom layers to learn speaker-related information and top layers to encode content-related information. Since the network capacity is limited, we believe the speech recognition performance could be further improved if the model is dedicated to audio content information learning. To this end, we propose **I**ntermediate **L**ayer **S**upervision for **S**elf-**S**upervised **L**earning (ILS-SSL), which forces the model to concentrate on content information as much as possible by adding an additional SSL loss on the intermediate layers. Experiments on LibriSpeech test-other set show that our method outperforms HuBERT significantly, which achieves a 23.5%/11.6% relative word error rate reduction in the w/o language model setting for Base/Large models. Detailed analysis shows the bottom layers of our model have a better correlation with phonetic units, which is consistent with our intuition and explains the success of our method for ASR. We will release our code and model at `https://github.com/microsoft/UniSpeech`.

***Index Terms*—** Self-supervised learning, Automatic speech recognition

1. INTRODUCTION

Automatic speech recognition (ASR) has seen rapid improvement over the last few years due to advances in deep learning. However, neural networks benefit from large quantities of labeled training data, which is much harder to come by than unlabeled data. This drives research in self-supervised learning (SSL), which attempts to learn representations from audio alone with some pretext tasks and then fine-tune the model on the supervised data. Examples of such pretext tasks include generative task [1, 2, 3, 4, 5, 6], discriminative task [7, 8, 9, 10, 11, 12, 13] or multi-task [14, 15, 16, 17].

To perform well on the ASR task, the SSL objectives need to be well designed to learn spoken content information. Some successful SSL methods show that the learnt representations are highly correlated with phonetic units, such as wav2vec 2.0 [11] and HuBERT [13]. However, studies indicate that not all layers have such a high correlation and different layers can learn disentangled aspects of speech. [18] train a supervised phoneme classifier on the top of frozen representations of each of the 24 blocks from a wav2vec 2.0 Large model and find that blocks 10-21 provide much lower phoneme error rate than others. In HuBERT [13], the authors run k-means clustering on representations of each of 12 blocks of a Base model and show that cluster assignments from blocks 5-12 have higher mutual information score with force-aligned phonetic transcripts. Furthermore, [19] collect representations from different HuBERT layers and weighted sum them for various downstream tasks where the weights can be learned. The model tends to assign larger weights to higher layers for phoneme recognition task, while assign larger weights to lower layers for speaker-related tasks. These observations indicate that the middle to top layers are better at learning content knowledge than the lower layers.

Since a network capability is limited, an interesting question is whether we can free up its capacity from learning speaker information or background noise and instead forcing more layers to learn content information. Following this motivation, we propose **I**ntermediate **L**ayer **S**upervision for **S**elf-**S**upervised **L**earning (ILS-SSL). Specifically, instead of computing the self-supervised loss only on the top layer, we also compute the loss on the intermediate layers. In this way, the lower layers must learn more content information to optimize the intermediate SSL loss. In this work, we use the masked prediction loss proposed in HuBERT [13] as our objective since this loss can help the model learn phonetic information and perform well on ASR task.

Our experiments either use the LibriSpeech 960h dataset or the Libri-Light 60k dataset for pre-training. In the BASE model setting (pre-training with LibriSpeech 960h), our models outperform baselines on all fine-tuning subsets (1h, 10h and 100h). Among them, the model fine-tuned with the 100h subset outperforms the competitive HuBERT baseline by a 25.4%/23.5% relative word error rate (WER) reduction on the test-clean/other testsets when decoding without a language model. In the LARGE model setting (pre-training with Libri-Light 60kh), our method shows relative 9.5%/11.6% WER reduction in the w/o language model setting when fine-tuning with 960h LibriSpeech data. When combining with Transformer language model, we reach a WER score of 1.8/3.2. We

Work done during an internship at Microsoft.

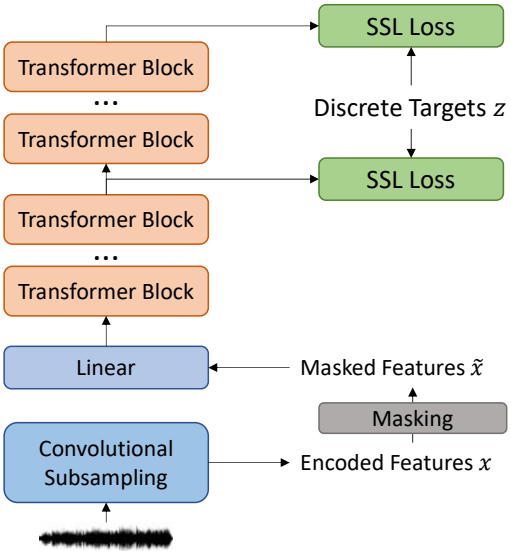

Fig. 1. Model Architecture.

further conduct analysis to explain why the simple method could outperform baselines. We find that ILS-SSL significantly improves the phonetic information learning for the bottom layers of the model. We also evaluate our model on the SUPERB benchmark [19] which includes ten different downstream tasks in four aspects of speech: content, speaker, semantics, and paralinguistics. The results also indicate that ILS-SSL could force the model to concentrate on content related information learning, leading to a better performance on content and semantics tasks.

2. BACKGROUND: HUBERT

HuBERT is an SSL method which benefits from an offline clustering step to provide target labels for a BERT-like prediction loss [20]. The backbone is a Transformer encoder [21]. During pre-training, the Transformer consumes masked acoustic features and the model is optimized to predict the discrete target labels on the masked regions. This forces the model to learn a combined acoustic and language model over the continuous inputs.

HuBERT adopts an iterative re-clustering and re-training process: For the first iteration, the targets are assigned by clustering on the MFCC features of the training data; For the second iteration, a new generation of training targets are created by clustering the latent representations from the first iteration model. The authors measure the correlation between assignments and frame-level force-aligned phonetic units and find that the middle layers obtain higher mutual information scores than others.

3. METHOD

3.1. Model Architecture

Our model architecture design is the same as HuBERT. As shown in Figure 1, it contains a convolutional feature encoder and a Transformer encoder. The convolutional encoder is composed of seven blocks of temporal convolution with channel size 512, strides (5,2,2,2,2,2,2) and kernel widths (10,3,3,3,3,2,2). We corrupt the convolutional output representations $\mathbf{X}$ by replacing part of features with a mask embedding and use this masked $\tilde{\mathbf{X}}$ as Transformer encoder input. We train Transformer in two different configurations: BASE and LARGE. The BASE model contains 12 layers with model dimension 768, inner dimension 3072 and 12 attention heads. The LARGE model contains 24 layers with dimension 1024, inner dimension 4096 and 16 attention heads. The Transformer encoder is equipped with a convolution based relative position embedding layer with kernel size 128 and 16 groups at bottom. We also add a bucket relative position embedding following the implementation in [22] with 320 embeddings shared across all layers. Each embedding is a scalar that is added to the attention logits and corresponds to a range of possible key-query offsets. The range increases logarithmically up to an offset of 800, beyond which we assign all relative offsets to the same embedding.

3.2. Intermediate Layer Supervision

The common practice of SSL is to compute the self-supervised loss on the top layer, such as wav2vec 2.0 and HuBERT. However, the lower layers of such a pre-trained model is shown to have a low correlation with phonetic information. In this work, we propose to apply intermediate layer supervision to encourage lower layers to learn content knowledge. During pre-training, we select a set of layers K as supervised layers and compute the masked prediction loss on the output hidden states $\mathbf{h}^l$, where $l \in K$. The prediction targets are the same for every supervised layer. The final loss is defined as:

$$L = -\sum_{l \in K} \sum_{t \in M} \log p^l(z_t|\mathbf{h}_t^l) \tag{1}$$

where M denotes the set of masked indices in time domain and $\mathbf{h}_t^l$ is the l-th layer output for step t. The distribution over codewords for each selected layer is parameterized with

$$p^l(z_t|\mathbf{h}_t^l) = \frac{\exp(sim(\mathbf{W}^l\mathbf{h}_t^l, \mathbf{e}_{z_t}^l)/\tau)}{\sum_{c'=1}^{C} \exp(sim(\mathbf{W}^l\mathbf{h}_t^l, \mathbf{e}_{c'}^l)/\tau)} \tag{2}$$

where $\mathbf{W}^l$ is a projection matrix, $\mathbf{h}_t$ is the hidden state for step t, $\mathbf{e}_{z_t}$ is the embedding for codeword c, C is the total number of the codewords, sim$(\mathbf{a}, \mathbf{b})$ computes the cosine similarity and $\tau = 0.1$ scales the logit.

As our method will influence the choice of layer to generate the training targets after the first iteration, we only apply intermediate layer supervision on the second iteration for a fair comparison. After pre-training, we remove all the prediction layers $\mathbf{W}^l$ and fine-tune the model with CTC loss. The CTC loss is applied only on the top Transformer layer.

4. EXPERIMENTAL SETUP

4.1. Dataset

For pre-training, we use the 960-hour audio data from LibriSpeech [23] to train the BASE model and the 60,000-hour unlabeled audio from LibriLight [24] to train the LARGE model. For fine-tuning, we consider four different partitions: Libri-light 1 hour, 10 hour splits [24] and LibriSpeech train-clean-100 subset for BASE, and LibriSpeech 960-hour full dataset for LARGE. The models are evaluated on test-clean/other sets.

4.2. Pre-training

Our pre-training process and hyperparameters follow [13]. For the BASE model, we first run k-means with 100 classes on the 39-dimensional MFCC features over 960 hours training set to generate the discrete targets and train the first iteration model. Then we run k-means with 500 classes on the 6-th layer output features to generate the next generation targets. For the LARGE model, we use the 9-th layer of the released HuBERT BASE [1] to generate target labels.

During training, each audio example is cropped to at most 15.6 seconds. The BASE model is trained on 32 GPUs with 87.5 seconds of audio per batch. The first iteration is trained for 250k steps and the second iteration is trained for 400k steps. The LARGE model is trained on 128 GPUs for 600k steps with batch size of 53.75 seconds. Models are optimized with AdamW optimizer with weight decay 0.01 and $\beta = (0.9, 0.98)$. The learning rate ramps up linearly for the first 32k steps and then decays linearly back to 0. The peak learning rates are 5e-4 for BASE and 1.5e-3 for LARGE.

4.3. Fine-tuning

The model is fine-tuned on 8 GPUs with a batch size of 200 seconds of audio for each GPU. The convolutional encoder is always fixed and we freeze the Transformer encoder part during the early training. We optimize with Adam and a tri-stage rate schedule where the learning rate is warmed up for the first 10% of updates, held constant for the next 40% and then linearly decayed for the remainder. For evaluation, we use wav2letter++ [25] beam search decoder with beam size 1500 for a 4-gram language model or beam size 50 for a Transformer language model. The hyperparameter settings mainly follow wav2vec2.0.

5. RESULTS

5.1. Main Results

Table 1 presents results compared with wav2vec2.0 and HuBERT. As HuBERT only reports the WER with LM in their paper, we either fine-tune their released pre-trained model or inference using their released fine-tuned model and report the without LM results. For our method, we set the intermediate layer set K as (4, 12) for BASE setting and as (9, 24) for LARGE setting, which performs best according to our offline experiments. On test-clean set, our method reaches relative 14.4%, 17.8% and 25.4% WER reductions against HuBERT baseline for different fine-tuning settings. The gain is even larger on noisy test-other set, where the relative WER reductions are 16.0%, 19.0% and 23.5%. LM fusion can close the gap to some extent, but the superiority of our method also persists. This indicates that our pre-training method can learn more ASR specific knowledge. On the 960h fine-tuning setup, our model improves the HuBERT baseline by 9.5%/11.6% respectively, indicating the effectiveness of our method for large-scale models. Fusing with Transformer LM can further improve our model to 1.8/3.2 WER score.

Model	LM	test-clean	test-other
BASE ***on 1-hour labeled***			
wav2vec 2.0 [11]	None	24.5	29.7
HuBERT (re-imple)	None	20.9	27.5
wav2vec 2.0 [11]	4-gram	5.5	11.3
HuBERT [13]	4-gram	6.1	11.3
ILS-SSL	None	17.9	23.1
ILS-SSL	4-gram	5.6	11.0
BASE ***on 10-hour labeled***			
wav2vec 2.0 [11]	None	11.1	17.6
HuBERT (re-imple)	None	10.1	16.8
wav2vec 2.0 [11]	4-gram	4.3	9.5
HuBERT [13]	4-gram	4.3	9.4
ILS-SSL	None	8.3	13.6
ILS-SSL	4-gram	3.8	8.1
BASE ***on 100-hour labeled***			
wav2vec 2.0 [11]	None	6.1	13.3
HuBERT (re-imple)	None	6.3	13.2
wav2vec 2.0 [11]	4-gram	3.4	8.0
HuBERT [13]	4-gram	3.4	8.1
ILS-SSL	None	4.7	10.1
ILS-SSL	4-gram	3.0	6.9
LARGE ***on 960-hour labeled***			
wav2vec 2.0 [11]	None	2.2	4.5
HuBERT	None	2.1	4.3
wav2vec 2.0 [11]	4-gram	2.0	3.6
HuBERT	4-gram	2.0	3.7
wav2vec 2.0 [11]	Transf	1.8	3.3
HuBERT [13]	Transf	1.9	3.3
ILS-SSL	None	1.9	3.8
ILS-SSL	4-gram	1.9	3.4
ILS-SSL	Transf	1.8	3.2

Table 1. Results and comparison with the literature.

5.2. Analysis

5.2.1. Cluster Quality Evaluation

To analyze whether our assumption is hold that ISL-SSL can learn more ASR specific information, following HuBERT, we

[1] https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt

Method	#Params	Corpus	Speaker			Content				Semantics			ParaL
			SID	ASV	SD	PR	ASR	KS	QbE	IC	SF		ER
			Acc ↑	EER ↓	DER ↓	PER ↓	WER ↓	Acc ↑	MTWV ↑	Acc ↑	F1 ↑	CER ↓	Acc ↑
FBANK	0	-	8.5E-4	9.56	10.05	82.01	23.18	8.63	0.0058	9.10	69.64	52.94	35.39
HuBERT BASE	94.68M	LS 960 hr	**81.42**	**5.11**	**5.88**	5.41	6.42	96.30	0.0736	98.34	88.53	25.20	64.92
ILS-SSL	94.68M	LS 960 hr	79.29	5.24	6.31	**5.00**	**5.45**	**96.79**	**0.0789**	**98.47**	**89.16**	**24.29**	**65.79**

Table 2. Universal speech representation evaluation on SUPERB benchmark. ParaL denote Paralinguistics aspect of speech.

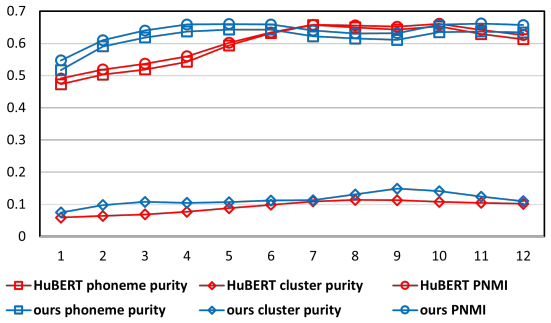

Fig. 2. Quality of the cluster assignments.

measure the correlation between the clustering assignments and the frame-level forced-aligned phonetic transcripts[2]. We extract the representations from each pre-trained layer and run k-means with cluster 500 on a 100 hour subset randomly sampled from training data. The correlation is measured by three metrics based on their co-occurrence: cluster purity, phone purity and phone normalized mutual information (PNMI) [13]. We use the model after the second iteration and evaluate on the joint set of dev-clean and dev-other. Figure 2 shows the comparison. Obviously, our method results in better clustering quality over all three metrics than HuBERT, especially for lower layers. Block 2-12 of our model can achieve a PNMI score over 0.6, while for baseline, only block 5-12 can achieve this. Intermediate layer supervision helps the lower layers learn more phonetic information, thus the burden on higher layers is reduced.

5.2.2. *Evaluation on non-ASR Tasks*

Recent work shows [26] that speech pre-trained models can solve full stack speech processing tasks. We evaluate the proposed method on the SUPERB benchmark [19], which is designed to provide a comprehensive testbed for pre-trained models on various speech tasks. It covers ten tasks, including Speaker Identification (SID), Automatic Speaker Verification (ASV), Speaker Diarization (SD), Phoneme Recognition (PR), Automatic Speech Recognition (ASR), Keyword Spotting (KS), Query by Example Spoken Term Detection (QbE), Intent Classification (IC), Slot Filling (SF), Emotion Recognition (ER). These tasks can be grouped into four aspects of speech: content, speaker, semantics, and paralinguistics.

Table 2 compares ILS-SSL BASE and HuBERT BASE on SUPERB, indicating the ILS-SSL is better than HuBERT on content and semantic related tasks, while the performance degradation is observed for the speaker related tasks. An explanation is that the pseudo-labels are highly correlated with the phone information, which may bias the model to discard the speaker characteristic and focus on the content.

[2]https://github.com/CorentinJ/librispeech-alignments

Model	w/o LM	with LM
HuBERT	16.8	9.4
+ILS-FT (share)	17.9	10.5
+ILS-FT (sep)	18.3	10.8
ILS-SSL	13.1	8.3
+share IL weights	13.5	8.5
+ILS-FT (share)	16.7	10.8
+ILS-FT (sep)	17.1	10.8
-bucket rel pos	14.5	8.8

Table 3. Ablation study results on dev-other set.

5.2.3. *Ablation Study*

Table 3 shows the ablation results conducted on 10 hours labeled data and evaluated on the dev-other set. For LM fusion, we use the 4-gram language model with beam size 50. The baseline HuBERT results are obtained using their released model. First, we share the projection weights $\mathbf{W}^l$ and the codeword embedding $\mathbf{e}^l$ in equation (2) for each supervised layer. The experiment is denoted as "+ share IL weights". The results are slightly worse but the gap is less than 3%. This indicates the gain is come from intermediate layer supervision instead of larger pre-training model size. Next, we apply the intermediate layer supervision in fine-tuning stage (ILS-FT), where we compute CTC loss on every $l \in K$. We use either shared or separated CTC weights. The results show that the ILS method cannot generalized from SSL to supervised task. We further remove the bucket relative position embedding in our model and the performance drops a little bit, indicating the bucket relative position embedding is a supplement to convolutional position embedding.

6. CONCLUSION

In this paper, we introduce ILS-SSL method to learn a speech recognition oriented self-supervised model. The method is simple but effective. It outperforms competitive baselines on LibriSpeech benchmark in different low-resource setting. Analysis shows our method can learn representations with a higher correlation with phonetic units. In the future, we will evaluate our method on larger model and larger datasets and implement it with other objectives.

7. REFERENCES

[1] Yu-An Chung, Wei-Ning Hsu, Hao Tang, et al., "An unsupervised autoregressive model for speech representation learning," in *Interspeech*, 2019.

[2] Andy T. Liu, Shu-Wen Yang, Po-Han Chi, et al., "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP*, 2020.

[3] Andy T. Liu, Shang-Wen Li, and Hung-yi Lee, "TERA: self-supervised learning of transformer encoder representation for speech," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 2351–2366, 2021.

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

[5] Yu-An Chung and James Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP*, 2020, pp. 3497–3501.

[6] Weiran Wang, Qingming Tang, and Karen Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP*, 2020.

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

[8] Steffen Schneider, Alexei Baevski, Ronan Collobert, et al., "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019.

[9] Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, et al., "Data augmenting contrastive learning of speech representations in the time domain," in *Proc. SLT*. IEEE, 2021, pp. 215–222.

[10] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR*, 2020.

[11] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, et al., "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[12] Yu Zhang, James Qin, Daniel S Park, et al., "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[13] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, et al., "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*.

[14] Santiago Pascual, Mirco Ravanelli, Joan Serrà, et al., "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Interspeech*, 2019.

[15] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, et al., "Multi-task self-supervised learning for robust speech recognition," in *ICASSP*, 2020.

[16] Yu-An Chung, Yu Zhang, Wei Han, et al., "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," *arXiv preprint arXiv:2108.06209*, 2021.

[17] Chengyi Wang, Yu Wu, Yao Qian, et al., "Unispeech: Unified speech representation learning with labeled and unlabeled data," in *ICML*, 2021.

[18] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, et al., "Unsupervised speech recognition," *arXiv preprint arXiv:2105.11084*, 2021.

[19] Shu-Wen Yang, Po-Han Chi, Yung-Sung Chuang, et al., "SUPERB: speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al., "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al., "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[22] Colin Raffel, Noam Shazeer, Adam Roberts, et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.

[23] Vassil Panayotov, Guoguo Chen, Daniel Povey, et al., "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[24] Jacob Kahn, Morgane Rivière, Weiyi Zheng, et al., "Libri-light: A benchmark for ASR with limited or no supervision," in *ICASSP*, 2020.

[25] Vineel Pratap, Awni Hannun, Qiantong Xu, et al., "wav2letter++: The fastest open-source speech recognition system," *arXiv preprint arXiv:1812.07625*, 2018.

[26] 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," *arXiv preprint arXiv:2110.13900*, 2021.