Interspeech 2022
18-22 September 2022, Incheon, Korea

Joint Encoder-Decoder Self-Supervised Pre-training for ASR

A Arunkumar, S Umesh

Speech Lab, Indian Institute of Technology Madras
arunkumaras10@gmail.com, umeshs@ee.iitm.ac.in

Abstract

Self-supervised learning (SSL) has shown tremendous success in various speech-related downstream tasks, including Automatic Speech Recognition (ASR). The output embeddings of the SSL model are treated as powerful short-time representations of the speech signal. However, in the ASR task, the main objective is to get the correct sequence of acoustic units, characters, or byte-pair encodings (BPEs). Usually, encoder-decoder architecture works exceptionally well for a sequence-to-sequence task like ASR. Therefore, in this paper, we propose a new paradigm that exploits the power of a decoder during self-supervised learning. We use Hidden Unit BERT (HuBERT) SSL framework to compute the conventional masked prediction loss for the encoder. In addition, we have introduced a decoder in the SSL framework and proposed a target preparation strategy for the decoder. Finally, we use a multi-task SSL setup wherein we jointly optimize both the encoder and decoder losses. We hypothesize that the presence of a decoder in the SSL model helps it learn an acoustic unit-based language model, which might improve the performance of an ASR downstream task. We compare our proposed SSL model with HuBERT and show up to 25% relative improvement in performance on ASR by finetuning on various LibriSpeech subsets.
Index Terms: Self-Supervised Learning, HuBERT, Automatic Speech Recognition, Encoder-Decoder Architecture

1. Introduction

Self-Supervised Learning (SSL) techniques [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] utilize large volumes of unlabeled speech data to learn high-level representations that work well for various speech-related downstream tasks. These representations have also been shown to work well for Automatic Speech Recognition (ASR) task where the objective is to get a sequence of acoustic units or their manifestation in terms of characters or Byte Pair Encodings (BPEs) [11]. There are several approaches in self-supervised learning in speech domain including those based on autoregressive predictive coding [2], contrastive losses [1, 3, 4, 6], masked prediction [5, 8, 9, 10] and multi-task learning [7]. Most SSL models are encoder-only models. In Hidden Unit BERT (HuBERT) [9], acoustic units are discovered using clustering techniques, and these units are used as targets in computing the masked prediction loss at the encoder output.

In supervised ASR tasks, encoder-decoder architectures [12, 13] are very popular since the decoder brings in the advantage of learning the implicit language model and outputting the desired sequence of acoustic units or characters. To bring the same advantage to SSL, this work proposes to incorporate a decoder in the SSL framework. The additional advantage is that the decoder can now better capture the sequential characteristics of the discovered acoustic units. A recent paper, SpeechT5 [14], proposed a unified-modal framework to learn joint contextual representations for speech and text data through a shared encoder-decoder network that uses unlabeled speech and unlabeled text data for pre-training an encoder-decoder model.

The main contributions of this paper are as follows:

- A transformer decoder [15] is introduced in the SSL framework in addition to the encoder, and we name this SSL technique as Joint Encoder-Decoder Pre-training.
- An unsupervised technique to generate the target sequence for the decoder is proposed
- Self-supervised learning is now done with the combined objective of masked prediction loss similar to conventional HuBERT and the newly introduced sequence loss of the decoder.
- The downstream ASR model is obtained by finetuning the proposed Encoder-Decoder SSL Model with downstream supervised data like the joint CTC-attention model [16] in a multitask learning framework.

The rest of the paper is organized as follows. Section 2 briefly explains the working of existing HuBERT and the architectural changes resulting in the proposed joint encoder-decoder pre-training method. The target sequence preparation for the decoder is also discussed in detail in the same section. Section 3 describes the ASR finetuning setup. Sections 4, 5 and 6 present the results of various experiments conducted. Section 7 draws important conclusions from this work and discusses future work. Pre-trained HuBERT-Base model used in the experiments was downloaded from fairseq [17]. The proposed SSL model was implemented using ESPnet toolkit [18], and all the experiments were conducted using the same.

2. Proposed Joint Encoder-Decoder Pre-training Method

The proposed Joint Encoder-Decoder SSL method introduces a decoder in addition to the encoder in the conventional SSL framework. We have considered HuBERT SSL in this work. The architecture of the conventional HuBERT SSL and the additional blocks that make up the proposed Joint Encoder-Decoder SSL are explained in this section.

2.1. Conventional HuBERT SSL

HuBERT is a non-contrastive SSL model trained on raw speech waveforms, and its architecture is shown in the left block of Figure 1. A convolutional feature extractor transforms the raw speech waveforms into the frame-level features. The feature extractor is frozen for the downstream tasks. A transformer encoder block receives masked feature frames according to a masking logic. The masking logic assigns a selection probability to each frame which was set to 8% in our experiments. Ten consecutive frames are masked for every selected frame, including the selected frame. The masked frames are fed to a Transformer Encoder which transforms them through several

Copyright © 2022 ISCA

10.21437/Interspeech.2022-11338

encoder layers, and each encoder output frame is then fed to a classification layer. Targets for the classification layer are the discovered cluster IDs of the corresponding input frames. The cluster ID corresponding to a frame is obtained by building a K-means clustering model with the MFCC features or HuBERT intermediate layer features extracted for the training data. A frame is assigned to a cluster if its distance to that cluster center is minimal. In this work, we always use the sixth layer features extracted from the existing pre-trained HuBERT-Base model for training the K-means model. Based on the weights assigned to the cluster ID prediction of unmasked frames, the contribution of unmasked frames to the final loss is decided. The prediction loss for masked frames alone was shown to be sufficient in [9], and therefore, we also use only masked prediction loss.

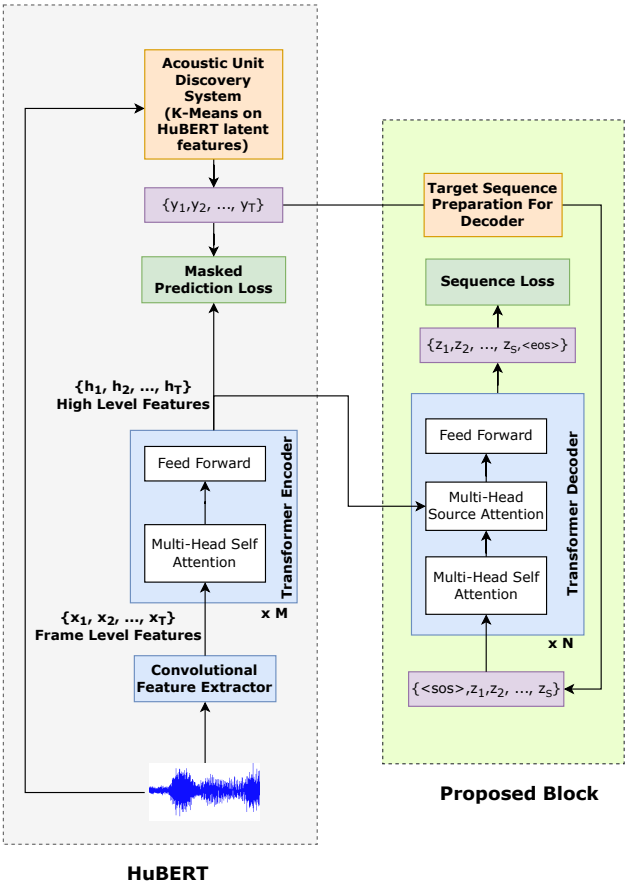

Figure 1: *Proposed Architecture*

2.2. Proposed Joint Encoder-Decoder SSL

2.2.1. Architecture

The proposed SSL method makes architectural changes to the conventional HuBERT SSL model by incorporating a Transformer decoder, as shown in Figure 1. Decoders have the ability to model and generate sequences. In ASR, the decoder acts as an implicit language model, and Transformer encoder-decoder ASR models are very popular. It might be beneficial to incorporate the Transformer decoder during SSL too. The added SSL decoder is made to attend to the SSL encoder outputs through source attention like in any other Transformer based encoder-decoder model.

2.2.2. Decoder Target Preparation

Figure 2 shows the approach that we have adopted to obtain in an unsupervised manner the targets for training the added SSL Transformer decoder. Targets for the added decoder are prepared from the discovered target cluster IDs of all the encoder input frames, both masked and unmasked. Any consecutive repetition in cluster IDs is replaced by a single cluster ID. Hence the decoder targets are also obtained by self-supervision. The core idea behind collapsing repeated target cluster IDs of the encoder frames for the decoder target preparation is to make the decoder learn the sequential characteristics of the acoustic units, and this might be beneficial to the downstream Transformer based Encoder-Decoder ASR.

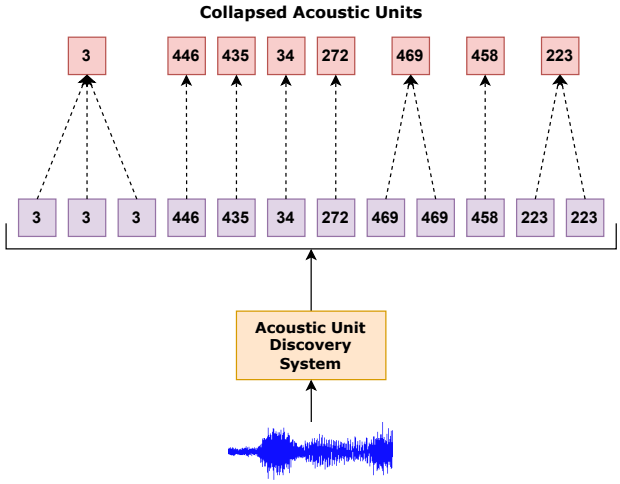

Figure 2: *Target Sequence Preparation*

2.2.3. Loss Function for the Proposed SSL Model

Let L_M be the masked prediction loss from the encoder and L_S be the sequence loss from the decoder. Then the total loss, L will be a weighted sum of both the losses as given by,

$$L = \alpha L_M + (1 - \alpha) L_S \tag{1}$$

where,

$$L_M = \Sigma_{t \in M} \log p(y_t | \hat{X}, t) \tag{2}$$

and

$$L_S = -\log p(Z | \hat{X}) \tag{3}$$

In equations (2) and (3), M is the set of masked indices and $\hat{X}$ represents the masked inputs $X[M]$. Since we wanted to give equal importance to both L_M and L_S, we set α to 0.5. So, while pre-training, the encoder weights are affected by both the losses during backpropogation.

3. Finetuning the SSL Models for ASR

The SSL models are usually evaluated on various downstream tasks by finetuning the SSL models with the task-specific loss functions. In this work, the ASR task is considered. The experimental setups for the ASR task with the conventional and proposed SSL technique are presented in the following subsections.

3.1. Finetuning the HuBERT SSL Model for ASR

For performing the ASR task with HuBERT, it is finetuned by using labeled data from different subsets of Libri-light. A randomly initialized CTC layer is added on top of the HuBERT final layer and finetuned to predict the characters. The loss function is taken as plain CTC. This ASR model serves as the baseline for evaluating the merit of the proposed SSL technique.

3.2. Finetuning the Proposed SSL Model for ASR

The proposed SSL model has both encoder and decoder, as mentioned in Section 2.2.1. For performing ASR in this case, both the encoder and decoder of the proposed SSL model are finetuned. The classification layers on top of the encoder and decoder are randomly initialized as done in HuBERT finetuning and finetuned to predict character outputs. The loss function for training the joint CTC-attention based ASR model is given by,

$$L = \beta L_{\text{CTC}}(Y|X) + (1-\beta)L_{\text{S2S}}(Y|X) \tag{4}$$

In equation (4), X represents the input sequence and Y represents the output sequence. The CTC loss is contributed by the encoder, and sequence to sequence(S2S) loss is contributed by the decoder. β is set to 0.3 since the standard recipes in ESPnet uses that value for training the joint CTC-attention based model . The final loss is a weighted sum of both these losses. This is similar to the standard ASR recipe used in espnet.

4. Results of ASR Finetuning of SSL Models Trained from Scratch

LibriSpeech-360h [19] was used to train the proposed SSL model from scratch for 101 epochs. LibriSpeech-960h was not used owing to compute resource constraints. For a fair comparison, the baseline HuBERT SSL model was also trained from scratch with LibriSpeech-360h since a 360-hour HuBERT pretrained model is not available in the public domain. Both the baseline and proposed SSL model have 12 encoder layers with 8 attention heads and 3072 linear units. The number of decoder layers in the proposed SSL model is 8, with 8 attention heads and 2048 linear units. The SSL models were trained using 8 GPUs with a batch size of at most 117 seconds of audio per GPU and a learning rate of 0.0001. Warmup scheduler with 25000 warmup steps and Adam optimizer were used.

Table 1: *Results of ASR Finetuning of SSL Models Trained from Scratch and without Language Model (LM)*

Model	test clean	test other
1h labeled		
HuBERT-360h (ours)	47.4	62.0
Proposed SSL-360h	**39.4**	**56.2**
10h labeled		
HuBERT-360h (ours)	29.5	47.4
Proposed SSL-360h	**22.6**	**41.0**
100h labeled		
HuBERT-360h (ours)	15.5	36.0
Proposed SSL-360h	**11.5**	**30.6**

For ASR finetuning, 1 GPU with a batch size of at most 200 seconds of audio and learning rate of 0.00002 were used. Warmup scheduler with 8000 warmup steps and Adam optimizer were used. For decoding, we used a beam size of 20, and the CTC weight was set to 0.3, which is the same value as during finetuning. The results of the downstream ASR tasks obtained by finetuning the respective SSL models are presented in Table 1. We have used three subsets of Libri-light to compare our proposed method with HuBERT baseline.

From Table 1, the following observations can be made:

- ASR finetuning of the proposed SSL model is superior to the baseline HuBERT SSL for all the Libri-light subsets taken.
- The relative improvements obtained from the proposed SSL increase with the increase in the amount of labeled data available with a maximum relative improvement of 25.8 % for the LibriSpeech-100h labeled subset.

5. Results of ASR Finetuning of SSL Models Trained by Continued Pre-training

The previous section described the experiments where the SSL models were trained from scratch. Due to resource constraints, LibriSpeech-360h was used instead of the bigger subset, LibriSpeech-960h. To show that our proposed model performs better than the baseline HuBERT SSL in a larger data setting, too, we opted for a continued pre-training approach on the readily available HuBERT-Base SSL model pre-trained with LibriSpeech-960h.

To get the equivalent of the proposed Joint Encoder-Decoder SSL model in the continued pre-training setup, a randomly initialized decoder was added to the already pre-trained 960-hour HuBERT-Base encoder. The pre-training was then *continued* in the same manner as mentioned in the proposed SSL method. The pre-training was continued for 40 epochs.

Table 2: *Results of ASR Finetuning of SSL Models Trained by Continued Pre-training and without LM*

Model	test clean	test other
1h labeled		
HuBERT-Base (Cont. Pre. Train)	32.0	40.6
Proposed SSL with HuBERT-Base Enc. + Rand. Init. Dec. (Cont. Pre. Train)	**24.0**	**31.2**
10h labeled		
HuBERT-Base (Cont. Pre. Train)	12.0	19.4
Proposed SSL with HuBERT-Base Enc.+ Rand. Init. Dec. (Cont. Pre. Train)	**9.8**	**16.4**
100h labeled		
HuBERT-Base (Cont. Pre. Train)	6.7	15.1
Proposed SSL with (HuBERT-Base Enc. + Rand. Init. Dec.) (Cont. Pre. Train)	**5.8**	**13.3**

Note: Enc. denotes Encoder, Rand. Init. Dec. denotes Randomly Initialized Decoder, Cont. Pre. Train denotes Continuosly Pre-trained

For the baseline, instead of directly finetuning the HuBERT-Base model for the CTC-based ASR task, we chose to continue the pre-training of HuBERT-Base for 40 epochs to ensure

fairness. This is because the pre-trained HuBERT-Base model would have used a different mapping for the acoustic units than the one used in our models. For example, an acoustic unit represented by cluster ID 100 in the pre-trained HuBERT-Base model may be represented by cluster ID 25 in our model.
The finetuned ASR results for the above mentioned SSL models with continued pre-training are presented in Table 2. From the Table 2, it can be seen that

- Proposed SSL outperforms the conventional HuBERT base model in large data setting too.
- The results are better than the SSL models trained from scratch, owing to the bigger LibriSpeech-960h data.
- In this case, the encoder from base HuBERT model is already well-trained and then the randomly initialized decoder is added for continued pre-training. When both encoder and decoder are jointly optimized while training from scratch, the relative improvement over conventional HuBERT is even better as seen in Section 4.

6. Results of ASR Finetuning with Randomly Initialized Decoder in the SSL Models

This section presents experiments that reinforce the idea that the added decoder in the proposed SSL holds significant importance in the downstream ASR. The decoder and encoder are jointly pre-trained with combined masked prediction and sequence loss in a self-supervised way. This architecture works best for downstream ASR tasks. It is important to note that the improvements due to the proposed model are not just because of the decoder being finetuned during finetuning for the downstream ASR task. For example, we could take the pre-trained HuBERT-360 hour model and add a randomly initialized decoder to it and do a finetuning step with joint CTC-attention losses. However, as shown in Table 3, this will give worse results than HuBERT-360 because the encoder is well pre-trained while the decoder and cross-attention have to be learned with only finetuning data. A similar observation can be made when we randomized the decoder before finetuning our proposed SSL model. Again the performance degrades when compared to our original SSL model. This proves that pre-training the decoder is really helpful and improves downstream ASR performance. The results show that pre-training the decoder is helpful and improves downstream ASR performance.
The corresponding results from Table 1 are reproduced in this table for comparison with the proposed model.

Table 3: *Results of ASR Finetuning with Randomly Initialized Decoder in the SSL Models and without LM for LibriSpeech-10h*

Model	test clean	test other
HuBERT-360 (ours)	29.5	47.4
Proposed SSL-360h	**22.6**	**41.0**
HuBERT-360h (ours) + Rand. Init. Dec.	32.0	42.0
Enc. of the Proposed SSL-360h + Rand. Init. Dec.	27.8	45.5

Note: Enc. denotes Encoder, Rand. Init. Dec. denotes Randomly Initialized Decoder

7. Conclusions and Future Work

In this paper, a Joint Encoder-Decoder Self-Supervised Pre-training model that jointly optimizes masked prediction loss from encoder and sequence loss from decoder was proposed in place of conventional encoder-only SSL techniques. Through various experiments, the proposed SSL model was proved to be superior to the conventional encoder-only HuBERT SSL model for the downstream ASR task. In the future, we plan to evaluate the proposed model by pre-training with LibriSpeech 960h. We also aim to analyze the proposed incorporation of a decoder in SSL using other state-of-the-art SSL techniques like WavLM. Furthermore, we would like to evaluate the proposed model for other downstream tasks like phoneme recognition.

8. Acknowledgement

We would like to thank Metilda N J for the technical discussion and her help in preparing this paper.

9. References

[1] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[2] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

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

[4] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[5] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6419–6423.

[6] 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.

[7] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, "Multi-task self-supervised learning for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6989–6993.

[8] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[9] 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.

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

[11] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[12] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[14] J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, Y. Wu, S. Liu, T. Ko, Q. Li, Y. Zhang, Z. Wei, Y. Qian, J. Li, and F. Wei, "SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5723–5738. [Online]. Available: https://aclanthology.org/2022.acl-long.393

[15] 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.

[16] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4835–4839.

[17] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[18] 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 *Proceedings of Interspeech*, 2018, pp. 2207–2211.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.