INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

CoBERT: Self-Supervised Speech Representation Learning Through Code Representation Learning

Chutong Meng[1,†], *Junyi Ao*[2,†], *Tom Ko*[1*], *Mingxuan Wang*[1], *Haizhou Li*[2]

[1]ByteDance
[2]Shenzhen Research Institute of Big Data, School of Data Science, The Chinese University of Hong Kong, Shenzhen, China

mengct00@gmail.com, junyiao1@link.cuhk.edu.cn, {tom.ko, wangmingxuan.89}@bytedance.com, haizhouli@cuhk.edu.cn

Abstract

Speech is the surface form of a finite set of phonetic units, which can be represented by discrete codes. We propose the **Co**de **BERT** (CoBERT) approach for self-supervised speech representation learning. The idea is to convert an utterance to a sequence of discrete codes, and perform code representation learning, where we predict the code representations based on a masked view of the original speech input. Unlike the prior self-distillation approaches of which the teacher and the student are of the same modality, our target model predicts representations from a different modality. CoBERT outperforms the most recent state-of-the-art performance on the ASR task and brings significant improvements on the SUPERB speech translation (ST) task. Our code and models are released at https://github.com/mct10/CoBERT.

Index Terms: self-supervised learning, BERT, data2vec

1. Introduction

Self-supervised speech representation learning has become an important research direction after the success of BERT [1] in natural language processing. A key motivation is that effective speech representations simplify the downstream tasks, thus, reducing the required amount of annotated data for supervised fine-tuning. Along this idea, a number of training approaches have been studied [2, 3, 4, 5, 6, 7, 8, 9], that include wav2vec [10, 11, 12], HuBERT [5], and the recent data2vec [7].

While unpaired speech data are exploited in self-supervised learning, unpaired text data [13, 14, 15] are found to be useful as well. The studies on speech-text pre-training have shown promising results, which seek to derive unified representations through joint training of between speech and text. These methods can be seen as an attempt to reduce the gap between representations of speech and text. A common problem for these methods is that it is difficult to align the representations of unpaired speech and text. This problem can be alleviated by using paired speech recognition data [16].

In this paper, we propose the **Co**de **BERT** (CoBERT) approach for self-supervised speech representation learning. Our method is inspired by the recent success of speech-text pre-training approaches and the data2vec approach. First, we believe that the performance gain by speech-text pre-training arises from the improved speech encoder which has captured extra information from the text representations. Although it is difficult to obtain all the transcripts for the unpaired speech data, recent studies [17] show that self-organized codes are closely related to the text of the original speech. Second, data2vec encourages the way of predicting contextualized latent representations in a self-distillation setup. Thus, we examine code representation learning to benefit speech representation learning. The core idea of our approach is to convert the speech to a sequence discrete units (codes), and perform code representation learning, where we predict the code representations based on a masked view of the original speech input. As the codes are generated from the unpaired speech data, the alignment problem encountered by speech-text pre-training approaches can be avoided.

The contributions of this work include, (i) We explore code representation learning using a self-distillation approach to benefit speech representation learning. (ii) CoBERT can successfully predict contextualized latent representations from different and multiple modalities at the same time. (iii) Experiments show that CoBERT relatively reduces the word error rate (WER) by 6.7% on the LibriSpeech subsets [18] and brings significant improvements on the SUPERB ST task [19] by around 1.35 BLEU compared to data2vec.

The rest of this paper is organized as follows. In Section 2, we discuss the related work. Section 3 illustrates our proposed approach, including code generation, code representation learning and speech representation learning. We share the experimental setting, results, and analyses in Section 4. Section 5 concludes the study.

2. Related work

CoBERT is motivated by and most related to vq-wav2vec [10], HuBERT [5] and data2vec [7]. vq-wav2vec is a pioneer work in exploring code representation learning. Further, they show that BERT-based representation learning is more effective in discrete than in continuous signal spaces [11]. HuBERT proposes to generate codes with an offline clustering step. Then it performs a BERT-based pre-training using these codes as target labels. data2vec outperforms BERT-based pre-training methods by promoting a direct prediction of contextualized latent representations instead of modality-specific targets. Our method integrates techniques of these methods.

We follow HuBERT's clustering procedures to obtain the codes so that CoBERT can perform representation learning completely in discrete code space. In comparison to data2vec, we follow their distillation approach to transfer the knowledge learnt from the code to the speech encoder, but our method differs in that CoBERT predicts representations of a different modality. The teacher is a code encoder but the student is a speech encoder. Moreover, CoBERT obtains better results when it predicts two sets of representations, one from the speech and one from the code, at the same time.

Another line of research aims at utilizing unpaired text to

† Equal contribution. Work is done during internship at Bytedance.
∗ Corresponding author.

10.21437/Interspeech.2023-1390

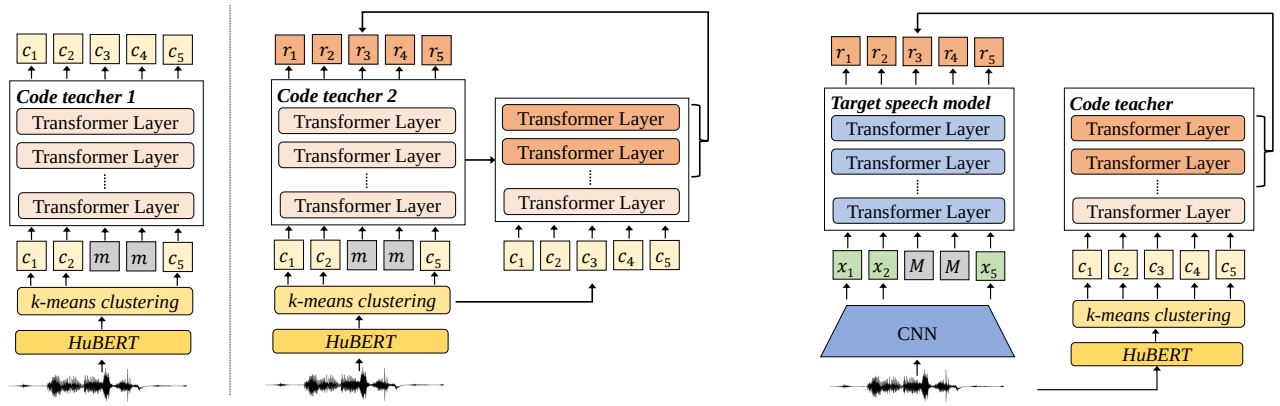

Figure 1: *Illustration of CoBERT. The input codes for the code teachers are obtained by k-means clustering on HuBERT features. (a) We examine two code representation learning methods. The left one is based on BERT and the right one is based on data2vec. (b) The representations produced by the code teacher are regressed by the target speech encoder given a masked view of the speech input.*

benefit speech representation learning [13, 14, 15]. In this paper, we do not compare our results with these methods as we pre-train models only with unpaired speech.

3. Method

Our approach (Fig. 1) can be divided into three steps: code generation, code representation learning and knowledge distillation.

3.1. Code generation

We follow the hidden units discovery steps described in [5] to extract the codes. More precisely, we feed the raw waveform into a HuBERT model, and then apply k-means clustering to the latent features at a certain transformer layer. The generated codes are denoted as $C = [c_1, ..., c_T]$ where T is the number of frames, $c_t \in [K]$, and K is the number of k-means classes.

We obtain codes from the 6$^{\text{th}}$ layer features of HuBERT BASE-it1 as well as 9$^{\text{th}}$ layer features of HuBERT BASE-it2. We train HuBERT BASE-it1 ourselves, with the raw waveform as inputs and k-means clustering results on Mel-frequency cepstral coefficients (MFCC) features as targets. We use the BASE model released by Fairseq [1] as the HuBERT BASE-it2 model. As shown in Table 2, codes from HuBERT BASE it2-L9 have better quality, so we use them in the subsequent steps.

3.2. Code representation learning

We examine code representation learning in two different ways (Fig. 1(a)).

Code teacher 1 is based on mask language modeling (MLM) similar to BERT [1]. Let Z denote the indices where the elements are to be masked. Also let $\widetilde{C} = mask(C, Z)$, which replaces the codes in C whose indices lie in Z with a special code M. Then the model is trained to predict the masked codes. We only compute the loss over the masked positions, which is

$$L_{mlm} = -\sum_{t \in Z} \log p(c_t | \widetilde{C}, t). \quad (1)$$

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

We use the same HuBERT BASE model as in [5], except that we replace the waveform encoder with a word embedding layer.

Code teacher 2 is based on the self-distillation approach similar to data2vec [7]. The teacher model takes C as input and produces target representations $R = [r_1, ..., r_T] = \frac{1}{L}\sum_{l=N-L+1}^{N} \hat{a}^l$, where N is the number of transformer layers and features from each of the top L layers will be normalized to $\hat{a}^l$ then summed to targets. The student model takes in $\widetilde{C}$ and regresses R.

The parameters of the teacher model are exponentially moving average (EMA) of those of the student model: $\theta_t \leftarrow \tau\theta_t + (1-\tau)\theta_s$, where θ_t is the parameters of the teacher model and θ_s is the parameters of the student model. Only the student model is updated via backpropagation. The objective is an $L2$ loss,

$$L_{sd} = \frac{1}{2}(\hat{R} - R)^2. \quad (2)$$

where $\hat{R}$ is the predictions of the student model and sd denotes self-distillation.

For code teacher 2, we apply the data2vec BASE model with its waveform encoder replaced with a word embedding layer. We use the features from the top 8 layers to generate targets, i.e., $L = 8$. We adopt slightly different masking strategies for the two teachers. Let p denote the probability of an input element chosen as the start position of a mask span. We adopt $p = 8\%$ for MLM based training and $p = 6.5\%$ for self-distillation based training. For both models, we set their mask spans to 10.

3.3. Knowledge distillation

The target speech encoder, CoBERT, is trained by predicting the representations of the code input given a masked view of the speech input (Fig. 1(b)). In other words, we distil the code model into a speech model.

In this paper, the architecture of the CoBERT model follows exactly the speech data2vec BASE model [7]. It takes raw waveform as input. Let $X = [x_1, \ldots, x_T]$ denote the speech hidden states after the waveform is downsampled by the same waveform encoder as the one described in [7]. We mask X with $p = 6.5\%$ and mask span 10 to generate a masked view $\widetilde{X} = mask(X, Z)$. The pre-trained code teacher produces

Table 1: *Main results on ASR and ST tasks. The WER scores are evaluated on the test-clean and test-other sets of LibriSpeech when using the 10 hours or 100 hours subsets as the training data and the BLEU scores are evaluated on the test set of the CoVoST2 En → De dataset. The WER scores of HuBERT and data2vec without LM and the BLEU score of data2vec are obtained by fine-tuning the released model as they are not reported in the corresponding papers.*

Model	10 hours subset				100 hours subset				SUPERB ASR	SUPERB ST
	No LM		4-gram		No LM		4-gram			
	clean	other	clean	other	clean	other	clean	other		
Baselines										
DiscreteBERT [11]	-	-	5.9	14.1	-	-	4.5	12.1	-	-
wav2vec 2.0 BASE [12]	11.1	17.6	4.3	9.5	6.1	13.3	3.4	8.0	6.43	14.81
HuBERT BASE [5]	10.1	16.8	4.3	9.4	5.6	12.7	3.4	8.1	6.42	15.53
WavLM BASE [6]	9.8	16.0	4.3	9.2	5.7	12.0	3.4	7.7	6.21	-
data2vec BASE [7]	7.2	12.3	3.9	8.1	4.2	9.7	**2.8**	6.8	4.94	17.56
Our Methods										
code teacher 1 (BERT)	8.8	14.5	5.1	9.5	5.0	10.8	3.6	8.1	-	17.11
code teacher 2 (data2vec)	7.9	12.9	4.4	8.8	4.6	9.9	3.4	7.5	-	17.75
CoBERT										
- distilling code teacher 1	7.4	12.7	3.7	7.9	4.3	9.8	2.9	6.8	-	17.81
+ self-distillation	7.2	12.2	3.7	7.7	4.3	9.6	3.0	6.6	-	18.12
- distilling code teacher 2	7.5	12.3	4.0	8.0	4.2	9.3	3.0	6.7	-	18.73
+ self-distillation	**6.8**	**11.4**	**3.6**	**7.4**	**4.0**	**8.9**	**2.8**	**6.4**	**4.74**	**19.10**

code representations R_{code} based on the aligned code sequence corresponding to X. The objective is to regress the code representations:

$$L_{code} = \frac{1}{2}(R_{Co} - R_{code})^2 \tag{3}$$

where R_{Co} is the representations generated by the CoBERT model.

Additionally, to fully utilize the speech information, our distillation process can be combined with the self-distillation setup used in data2vec. The teacher model, which averages the parameters of the CoBERT model exponentially, takes X as input and produces speech representations R_{speech} from its top 8 layers. We use a separate linear layer on top of CoBERT model, which generates $R_{Co'}$, to regress R_{speech}. The loss from the self-distillation teacher is therefore

$$L_{speech} = \frac{1}{2}(R_{Co'} - R_{speech})^2. \tag{4}$$

The final loss for CoBERT is a weighted sum of L_{code} and L_{speech}:

$$L_{CoBERT} = \alpha L_{code} + (1-\alpha) L_{speech}. \tag{5}$$

where $0 \leq \alpha \leq 1$. In all of our experiments, we set $\alpha = 0.5$.

4. Experiments

4.1. Experiment Setup

We conduct our experiments using Fairseq[1] [20]. For pre-training, we mainly follow the training process and hyper-parameters in [5, 7]. We use the full 960 hours training set of LibriSpeech [18] without the transcription for pre-training. For code teacher 1, we optimize the model with Adam [21] by warming up the learning rate for the first 8% of updates to a peak of 5×10^{-4}, which is linearly decayed for the following updates. For code teacher 2 and CoBERT experiments, we use Adam [21] to optimize the model with a tri-stage scheduler, which linearly warms up the learning rate for the first 3% of updates to a peak of 5×10^{-4}, holds it for 90% of updates and linearly decays the learning rate for the following updates. We pre-train the model for 400k updates on 16 GPUs with a batch size of around 10k tokens, 11.85k tokens and 237s samples per GPU for code teacher 1, code teacher 2 and CoBERT experiments.

For the ASR fine-tuning, we utilize 10 and 100 hours subsets, and segment the text with the character set. The learning rate is warmed up for the first 10% steps, held as a constant for the following 40% steps, and decayed linearly for the rest steps. For the 10/100 hours subset, we train the model with a learning rate of 5e-5/3e-5 for 20k/80k. For the 10 hours subset, the encoder part is fixed for the first 10k steps.

We evaluate our model on the ASR task by using wav2letter++ [22] beam search decoder with a beam size of 1500 for 4-gram language model-fused decoding, which optimized:

$$\log P_{ctc}(Y|X) + \omega_1 \log P_{LM}(Y) + \omega_2 |Y| \tag{6}$$

where Y is the prediction of a text sequence, $|Y|$ is the sequence length, and ω_1 and ω_2 are the weights of the language model and word score, respectively. The decoding hyperparameters, including language model weight, word score and silence weight, are searched with Ax[1], which is a Bayesian optimization toolkit. We use the official 4-gram language model [2] trained on the LibriSpeech-LM corpus for inference.

For the SUPERB ASR and ST fine-tuning, the training and inference processes follow [19]. We use LibriSpeech and CoVoST2 en→de dataset [23] for evaluation. The downstream model for ST has three transformer layers for both the encoder and decoder, with an additional convolution layer for downsampling the input. The downstream model for ASR has two layers of BLSTM and is optimized by CTC loss on characters level.

[1] https://github.com/pytorch/fairseq

[1] https://github.com/facebook/Ax

[2] https://www.openslr.org/resources/11/4-gram.arpa.gz

4.2. Main Results

Table 1 shows the main results of code teachers and CoBERT on ASR and ST tasks. We evaluate the WER scores on the standard LibriSpeech test-clean and test-other sets and BLEU scores on the test set of the CoVoST2 En $\rightarrow$ De dataset. We compare our method with several works from the literature, including DiscreteBERT [11], wav2vec 2.0 [12], HuBERT [5], WavLM [6] and data2vec [7], which are competitive self-supervised approaches. As the WER scores without LM of HuBERT and data2vec and the BLEU scores of data2vec are not reported in the corresponding papers, the results are obtained by fine-tuning the released model.

Experiments show that without LM fusion, code teacher 1 outperforms HuBERT on the ASR task, which relatively brings 12.1% and 14.2% reductions on the test-clean and test-other sets, while it performs worse or on par compared to the HuBERT baseline with LM fusion. For the ST task, code teacher 1 achieves significant improvement by around 1.58 BLEU. This indicates the code representations of code teacher 1 may carry extra semantic information. In consideration with a similar observation in [11], we hypothesize that BERT-based representation learning may be more effective in discrete than in continuous signal spaces.

Without self-distillation, CoBERT models outperform DiscreteBERT, HuBERT and wav2vec 2.0 by a large margin for both the ASR and ST tasks, while performing better than data2vec on the ST task. With self-distillation, CoBERT reaches a relatively 6.7% WER reduction on the averages of all ASR sets and improves the BLEU score from 17.56 to 19.10 compared to the data2vec baseline.

4.3. Analysis

4.3.1. Code quality across different features

As code quality may decide the final performance, we investigate the clustering quality across different features, including features of the 6th layer of HuBERT BASE-it1, the 9th layer of HuBERT BASE-it2, the 9th layer of data2vec, and the average top 8 layers of data2vec. We generate frame-level phonetic transcripts using Montreal Forced Aligner [24] with the default pre-trained English acoustic model [25] and dictionary [26]. Then we follow [5] to compute the code quality metrics using the codes and the phonemes. The results are shown in Table 2. Although the codes from it2-L9 are better than those from it1-L6, they achieve similar performance on the test-other subset of LibriSpeech. Therefore, it is fair for us to compare the performance of CoBERT and HuBERT. We also explore whether we can derive better-quality codes from the data2vec model since it has a better ASR performance. However, it turns out that the qualities of codes from data2vec are much worse than HuBERT's.

Table 2: *Quality of the clustering assignments and the effect on the ASR task with respect to different features.*

model	feature	phone purity	cluster purity	PNMI	WER
HuBERT	it1-L6	0.668	0.110	0.657	12.74
	it2-L9	0.674	0.107	0.666	12.73
data2vec	it1-L9	0.583	0.100	0.594	-
	Top8-avg	0.536	0.111	0.540	-

4.3.2. Effect of different teachers and model input

In this section, we verify the importance of the code teachers by using different teachers for the distillation. In table 3, the parameters of the HuBERT model and code teachers are fixed throughout the distillation. We intentionally not to further distil the data2vec checkpoint in table 1 as it is already trained with self-distillation.

Firstly, we want to show that distilling code models into speech models is better than distilling speech models. In the experiments of using one teacher, distilling HuBERT performs better than HuBERT itself and distilling any of the code teachers achieves a significant improvement compared to it. This indicates code teachers may contain extra language information compared to the conventional self-supervised speech models.

Secondly, we observe that distilling code teachers complements self-distillation and can achieve further improvement. When self-distillation is applied, distilling code teacher 1 outperforms distilling HuBERT by 0.2/0.8 WER reduction on the ASR task and 0.98 BLEU improvement on the ST task. This proves that improvements achieved by our method can not be achieved by simply distilling speech models.

Table 3: *Comparison of using different teacher models in terms of BLEU and WER scores. For ASR, the models are fine-tuned with 100 hours subset of LibriSpeech.*

Teacher model	**BLEU** $\uparrow$	**WER** $\downarrow$	
	SUPERB ST	test-clean	test-other
W/o self-distillation			
HuBERT	17.19	5.0	10.6
code teacher 1	17.81	4.3	9.8
code teacher 2	18.73	4.2	9.3
With self-distillation			
HuBERT	17.14	4.5	10.4
code teacher 1	18.12	4.3	9.6
code teacher 2	19.10	4.0	8.9

5. Conclusions

We present CoBERT, a self-supervised speech representation learning approach which enables a speech student encoder to learn contextualized latent representations from both speech and code modalities. First of all, we pre-train code models with HuBERT codes to benefit from training in discrete space. Then, we distil the code model into a speech model, which aims at performing a better learning across modalities. The significant improvement on the ST task indicates the representation of CoBERT may carry more language information compared to prior work.

In this paper, only codes generated from the unpaired speech are used to pre-train the code teacher and the amount of them is far less than the data scale to train a state-of-the-art text encoder. Future work may investigate the way to exploit text data to compensate for the paired code scarcity. We would also like to evaluate CoBERT on more spoken language understanding tasks and pre-train a multilingual CoBERT for multilingual speech translation task.

6. Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No. 62271432), Internal Project Fund from Shenzhen Research Institute of Big Data (Grant No. T00120220002), and Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen (Grant No. B10120210117-KP02).

7. References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*. Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

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

[3] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-Training for Speech Recognition," in *Proc. Interspeech*, 2019, pp. 3465–3469.

[4] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *Proc. ICASSP*. IEEE, 2020, pp. 3497–3501.

[5] 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.

[6] 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*, 2022.

[7] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proc. ICML*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 1298–1312. [Online]. Available: https://proceedings.mlr.press/v162/baevski22a.html

[8] X. Cheng, Q. Dong, F. Yue, T. Ko, M. Wang, and Y. Zou, "M3st: Mix at three levels for speech translation," *arXiv preprint arXiv:2212.03657*, 2022.

[9] Z. Ma, Z. Zheng, C. Tang, Y. Wang, and X. Chen, "MT4SSL: Boosting Self-Supervised Speech Representation Learning by Integrating Multiple Targets," *Proc. of InterSpeech*, 2023.

[10] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. ICLR*, 2020. [Online]. Available: https://openreview.net/forum?id=rylwJxrYDS

[11] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for asr," in *Proc. ICASSP*, 2020, pp. 7694–7698.

[12] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020, pp. 12 449–12 460.

[13] 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 *Proc. ACL*. Association for Computational Linguistics, 2022, pp. 5723–5738. [Online]. Available: https://aclanthology.org/2022.acl-long.393

[14] A. Bapna, Y.-a. Chung, N. Wu, A. Gulati, Y. Jia, J. H. Clark, M. Johnson, J. Riesa, A. Conneau, and Y. Zhang, "Slam: A unified encoder for speech and language modeling via speech-text joint pre-training," *arXiv preprint arXiv:2110.10329*, 2021.

[15] Y. Tang, H. Gong, N. Dong, C. Wang, W.-N. Hsu, J. Gu, A. Baevski, X. Li, A. Mohamed, M. Auli, and J. Pino, "Unified speech-text pre-training for speech translation and recognition," in *Proc. ACL*. Association for Computational Linguistics, May 2022, pp. 1488–1499. [Online]. Available: https://aclanthology.org/2022.acl-long.105

[16] R. Ye, M. Wang, and L. Li, "Cross-modal contrastive learning for speech translation," in *Proc. ACL*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 5099–5113. [Online]. Available: https://aclanthology.org/2022.naacl-main.376

[17] J. Ao, Z. Zhang, L. Zhou, S. Liu, H. Li, T. Ko, L. Dai, J. Li, Y. Qian, and F. Wei, "Pre-Training Transformer Decoder for End-to-End ASR Model with Unpaired Speech Data," in *Proc. Interspeech*, 2022, pp. 2658–2662.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*. IEEE, 2015, pp. 5206–5210.

[19] H.-S. Tsai, H.-J. Chang, W.-C. Huang, Z. Huang, K. Lakhotia, S.-w. Yang, S. Dong, A. Liu, C.-I. Lai, J. Shi, X. Chang, P. Hall, H.-J. Chen, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB-SG: Enhanced speech processing universal PERformance benchmark for semantic and generative capabilities," in *Proc. ACL*. Association for Computational Linguistics, May 2022, pp. 8479–8492. [Online]. Available: https://aclanthology.org/2022.acl-long.580

[20] 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 *Proc. NAACL-HLT*, 2019.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf

[22] V. Pratap, A. Hannun, Q. Xu, J. Cai, J. Kahn, G. Synnaeve, V. Liptchinsky, and R. Collobert, "Wav2letter++: A fast open-source speech recognition system," in *Proc. ICASSP*, 2019, pp. 6460–6464.

[23] C. Wang, A. Wu, and J. Pino, "Covost 2: A massively multilingual speech-to-text translation corpus," 2020.

[24] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech 2017*, 2017, pp. 498–502.

[25] M. McAuliffe and M. Sonderegger, "English (us) arpa acoustic model v2.0.0a," https://mfa-models.readthedocs.io/acoustic/English/English(US)ARPAacousticmodelv2_0_0a.html, Tech. Rep., May 2022.

[26] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," *Canadian Acoustics*, vol. 39, no. 3, pp. 192–193, 2011.