ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-7605-5/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICASSP39728.2021.9414460

HUBERT: HOW MUCH CAN A BAD TEACHER BENEFIT ASR PRE-TRAINING?

Wei-Ning Hsu[1], *Yao-Hung Hubert Tsai*[2], *Benjamin Bolte*[1], *Ruslan Salakhutdinov*[2], *Abdelrahman Mohamed*[1]

[1]Facebook AI Research [2]Carnegie Mellon University

ABSTRACT

Compared to vision and language applications, self-supervised pre-training approaches for ASR are challenged by three unique problems: (1) There are multiple sound units in each input utterance, (2) With audio-only pre-training, there is no lexicon of sound units, and (3) Sound units have variable lengths with no explicit segmentation. In this paper, we propose the Hidden-Unit BERT (HUBERT) model which utilizes a cheap k-means clustering step to provide aligned target labels for pre-training of a BERT model. A key ingredient of our approach is applying the predictive loss over the masked regions only. This allows the pre-training stage to benefit from the consistency of the unsupervised teacher rather that its intrinsic quality. Starting with a simple k-means teacher of 100 cluster, and using two iterations of clustering, the HUBERT model matches the state-of-the-art wav2vec 2.0 performance on the ultra low-resource Libri-light 10h, 1h, 10min supervised subsets.

Index Terms— representation learning, pre-training

1. INTRODUCTION

Given the huge imbalance between available labeled and unlabeled data, self-supervised pre-training is key for good performance on low-resource downstream tasks. Learning representations of discrete input sequences, such as in Natural Language Processing (NLP) applications, uses either masked prediction [1, 2] or auto-regressive generation [3] of input sequences with partial obfuscation. For continuous inputs, such as in Computer Vision (CV) applications, representations are often learned through instance classification, in which each image and its augmentations are treated as a single output class, contrasted against all other samples [4].

Speech signals differ from text and images in that they are *continuous-valued sequences*. Self-supervised learning for the speech recognition domain faces unique challenges from those in CV and NLP. Firstly, the presence of multiple sounds in each input utterance breaks the instance classification assumption used in many CV pre-training approaches. Secondly, during pre-training, there is no prior lexicon of discrete sound units available, as in NLP applications in which words or word pieces are used, hindering the use of predictive losses. Lastly, the boundaries between sound units are not known, which complicates masked prediction training.

To deal with the first problem, Contrastive Predictive Coding (CPC) [5, 6] distinguishes nearby future input features from negatives sampled from elsewhere in the utterance, based on the idea of ”slow features” [7]. Online Vector-Quantization (VQ) was utilized for learning a discrete dictionary of input image, speech, and video features [8]. Combining online VQ and the CPC loss function, vq-wav2vec [9] learns a discrete latent dictionary through a convolutional encoder. Once this encoder is trained and fixed, a BERT transformer model and pre-training loss is applied over latent discrete sequences as a feature extractor for an ASR model [9] or to directly fine-tune the BERT transformer model using a CTC loss for low-resource scenarios [10]. The wav2vec 2.0 model [11] extends these two approaches by jointly learning a BERT transformer model and a latent discrete dictionary of encoded speech input using a contrastive loss, with impressive downstream ASR performance for both high- and low-resource conditions.

A key ingredient of these models is the discrete output latents that are used as noisy target labels during pre-training. Pseudo-labeling [12, 13, 14, 15] can be thought of as a special instance of this approach in which the target sub-network is trained using supervised data, then used to generate pseudo-labels for a larger volume of unlabeled audio recordings. On the other end of the spectrum, DeepCluster [16] generates unsupervised pseudo-labels for input images using batch VQ on learned features during pre-training.

In this paper, we introduce **H**idden **U**nit **BERT** (HUBERT) that benefits from using offline hidden cluster assignments as noisy labels for BERT-like per-training. Concretely, a BERT model consumes aggressively masked continuous speech features to predict pre-determined k-means cluster assignments. The predictive loss is only applied over the masked regions, forcing the model to learn good high-level representations of unmasked inputs in order to correctly infer the targets of masked ones. Intuitively, the HUBERT model is forced to learn both acoustic and language models from continuous inputs using masked prediction. One key insight motivating this work is the importance of consistency in the targets not just their correctness. If the model were to predict the output of the entire utterance, a high-quality target generator would be required.

When the HUBERT model is pre-trained on the standard 960h Librispeech audio [17] and fine-tuned on the 10h Libri-light supervised subset [18], k-means teachers trained on MFCC features get below 12% of WER on the dev-other set. Pre-training with targets from a second k-means iteration provides 30% relative reduction in WER conditioned on the same number of pre-training steps (100k), and matches the state-of-the-art wav2vec 2.0 performance on the Libri-light 10 hour, 1 hour, 10 minute supervised sets.

2. METHOD

2.1. Preliminaries

Self-supervised learning aims to leverage large amounts of unpaired data $\{X_i\}_{i=1}^{N_u}$ to pre-train a model f or learn representations $f(X)$ that can improve the performance on downstream tasks. The method this paper proposes is inspired by *pseudo labeling* and *masked prediction*, which we now describe.

Pseudo-Labeling, also known as self-training, is one of the most successful approaches to semi-supervised learning in ASR [12, 13, 14, 15]. To leverage unlabeled speech, a teacher model g is first trained on small quantities of labeled data $\mathcal{D}_l = \{(X_j, Y_j)\}_{j=1}^{N_l}$ and then used to create pseudo labels for a larger volume of unlabeled data $\{\hat{Y}_i = g(X_i)\}_{i=1}^{N_u}$. A student model can be trained by using both real and pseudo pairs, $\mathcal{D}_l \cup \{(X_i, \hat{Y}_i)\}_{i=1}^{N_u}$ with a supervised objective. **Masked prediction** is typically applied to structural data (e.g., temporal data $X = [x_1, \cdots, x_T]$) where a model is tasked to predict some target labels for artificially corrupted input regions, conditioned on representations of uncorrupted parts of the sequence. These target labels are generally derived from the input sequence itself. When both the input and the label are the same, it is known as masked language modeling in the NLP community [1], and inpainting for the speech and vision communities [19].

We now describe HUBERT, a speech pre-training framework that exploits hidden units obtained from unsupervised teachers for BERT-style masked prediction pre-training.

2.2. Clustering for Unsupervised Pseudo Labeling

For labels to be used with masked prediction, a teacher g is required to produce *frame-level* annotations such that each input frame x_t is assigned a label. In semi-supervised learning, the acoustic model trained on text and speech pairs can be used to provide pseudo-phonetic labels for each frame. However, for self-supervised learning, one only has access to speech data and therefore such a teacher is not available. Nevertheless, simple discrete latent variable models such has k-means and Gaussian mixture models (GMMs) have shown to be capable of inferring hidden units that exhibit non-trivial correlation with the underlying acoustic units [20] (see also Table 2). Inspired by this, we propose to use these models as *bad teachers* to provide frame-level targets. To differentiate automatically determined hidden units from true acoustic units Y, we denote them with $h(X) = Z = [z_1, \cdots, z_T]$, where $z_t \in [C]$ is a C-class categorical variable and h is an unsupervised teacher. This technique is also used in CV for unsupervised labeling [16].

2.3. Pre-Training via Masked Pseudo Label Prediction

Let $M \subset [T]$ denote the set of indices to be masked for a length-T sequence X, and $\tilde{X} = r(X, M)$ denote a corrupted version of X where x_t is replaced with a mask embedding $\tilde{x}$ if $t \in M$. A masked prediction model f takes as input $\tilde{X}$ and predicts a distribution of the target label at each frame $p_f(\cdot \mid \tilde{X}, t)$. There are two decisions to be made for masked prediction: *how to mask* and *where to predict*.

Regarding the first decision, we adopt the same strategies used in SpanBERT [21] and wav2vec 2.0 [11] for mask generation, where $p\%$ of the frames are randomly selected as start indices, and spans of l frames are masked. To address the second decision, we denote the cross entropy loss computed over masked and unmasked frames as L_m and L_u respectively. L_m is defined as:

$$L_m(f; X, M, Z) = \sum_{t \in M} \log p_f(z_t \mid \tilde{X}, t), \tag{1}$$

and L_u is of the same form except that it sums over $t \notin M$. The final loss is computed as a weighted sum of the two terms: $L = \alpha L_m + (1 - \alpha) L_u$. In one extreme where $\alpha = 0$, the loss is only computed on the unmasked frames, which is similar to acoustic modeling in a hybrid ASR system. In the other extreme where $\alpha = 1$, the loss is only computed on masked frames and the model has to predict the labels of the unseen frames from the context, analogous to language modeling. To excel in the former task, the student model simply has to accurately model the teacher's annotations. To excel in the latter task, the student model is required to both model the *teacher's annotations* and *predict the unseen annotations*. We hypothesize that this task is more resilient to the teacher's quality.

2.4. Teacher Ensembling and Iterative Refinement

A simple idea to improve the pseudo-label quality is to introduce multiple teacher models. While each teacher may perform terribly individually, different teachers can provide complementary information to facilitate pre-training. For example, with different numbers of clusters, k-means can create annotations of different granularity, from manner classes (vowel/consonant) to sub-phone states (senones). The proposed framework can be easily extended to utilize multiple teachers. Let $Z^{(k)}$ be the label sequences annotated by the k-th teacher. We can now re-write L_m as:

$$L_m(f; X, \{Z^{(k)}\}_k, M) = \sum_{t \in M} \sum_{k} \log p_f^{(k)}(z_t^{(k)} \mid \tilde{X}, t), \tag{2}$$

and similarly for the unmasked loss L_u. This is analogous to multi-task learning, but with tasks created by unsupervised clustering. Additionally, teacher ensembling is intriguing because it can be used alongside product quantization (PQ) [22], where a feature space is partitioned into multiple subspaces and each of the subspaces is quantized separately. This allows effective Euclidean distance-based quantization such as k-means for high-dimensional features. The effective codebook size with teacher ensembling can be seen as the product of the size of all codebooks.

In addition to taking multiple bad teachers, one can also consider *refining* the teacher over the course of pre-training. Since we expect a pre-trained model to provide better representations than the raw feature such as MFCC, we can create a new generation of teachers by building a discrete latent model with the learned features, and repeating the pre-training process with the new discovered units.

2.5. Implementation

The proposed HUBERT model adopts the wav2vec 2.0 architecture [11], which is composed of one convolutional waveform encoder, a BERT encoder [1], a projection layer and a code embedding layer. Unless otherwise mentioned, the hyperparameters of the waveform encoder and the BERT encoder are identical to the BASE configuration in [11], where the waveform encoder is composed of seven 512-channel layers with strides [5,2,2,2,2,2,2] and kernel widths [10,3,3,3,3,2,2], and the BERT encoder consists of twelve transformer blocks with 768-dimensional input/output and an inner FFN dimension 3072. Following wav2vec 2.0, the audio encoder generates a feature sequence for a waveform at a 20ms framerate, which is then randomly masked as described in Section 2.3. The BERT encoder takes as input the masked sequence and outputs a 768-dimensional feature sequence $[o_1, \cdots, o_T]$. The distribution over codes is parameterized with

$$p_f^{(k)}(c \mid \tilde{X}, t) = \frac{\exp(\text{sim}(A^{(k)} o_t, e_c)/\tau)}{\sum_{c'=1}^{C} \exp(\text{sim}(A^{(k)} o_t, e_{c'})/\tau)}, \tag{3}$$

where $A \in \mathbb{R}^{256 \times 768}$ is the projection matrix, $e_c \in \mathbb{R}^{256}$ is the embedding for code c, $\text{sim}(\cdot, \cdot)$ computes the cosine similarity between two vectors, and τ scales the logit, which is set to 0.1. When a teacher ensemble is used, we consider using both the same or separate projection matrices $A^{(k)}$ for each teacher.

To fine-tune a pre-trained model for ASR, the projection layer(s) is removed and replaced with a randomly initialized softmax layer for end-to-end training using the connectionist temporal classification (CTC) [23] objective. The target vocabulary includes 26 English characters, a space, an apostrophe, and a special CTC blank symbol.

3. RELATED WORK

We discuss recent studies on self-supervised speech representation learning by grouping them by training objective. The earliest line of work relies on generative modeling with latent variables [24, 25, 8, 26, 27, 28]. Training amounts to maximal likelihood estimation and the learned latents are used as the representations. Prediction-based self-supervised learning has gathered increasing interests recently, where a model is tasked to predict the content of the unseen regions [29, 30, 31, 32, 33, 34, 35] or to contrast the target unseen frame with randomly sampled ones [5, 36, 6, 11]. There are also models that combine both the predictive and the contrastive losses [9, 10]. These objective can usually be interpreted as mutual information maximization [37].

This work is related to DiscreteBERT [10], since both predict discrete pseudo labels for the masked regions. However, we use raw waveforms as the input and investigate a broader range of choices for the teacher model, which eventually results in superior performance under a comparable setup. HUBERT is also related to wav2vec 2.0 [11], but the latter employs a contrastive loss that requires sampling negative frames and only explores quantizing the waveform encoder output. Our proposed method adopts a simpler predictive loss and matches the performance of wav2vec 2.0. Moreoever, our experiments in Table 5 reveals that the best pseudo labels may not be from the waveform encoder.

4. EXPERIMENTAL SETUP

We use the full 960 hours of LibriSpeech [17] audio for pre-training. By default, each model is pre-trained for 100k steps on 32 GPUs, with a batch size of at most 87.5 seconds of audio per GPU, which takes about 9.5 hours to complete training. Mask span is set to $l = 10$, and $p = 8\%$ of the waveform encoder output frames are randomly selected as mask start if not otherwise mentioned. Adam [38] optimizer is used with $\beta = (0.9, 0.98)$, and the learning rate ramps up linearly from 0 to 5e-4 for the first 8% of the training steps, and then decays linearly back to zero. The first generation of teachers are generated by running k-means and GMM clustering on 39-D MFCC features (13 coefficients with the first and the second order derivatives) with $\{50, 100, 500\}$ clusters using `scikit-learn` [39]. `MiniBatchKMeans` are used for quantization by default due to its simplicity and efficiency.

After pre-training, a model is fine-tuned for 80k steps on the 10-hour (default), 1-hour, and 10-minute Libri-light splits [18] to simulate a low-resource scenario, which are balanced LibriSpeech subsets of `train-*` containing 50/50 `clean` and `other` utterances. An effective batch size of at most 800 second is used, and training takes roughly 20 hours to complete on a single GPU. The same Adam optimizer is used, and the learning rate ramps up linearly for first 10% of the steps to 2e-5 and then decay exponentially to 5% of the peak learning rate. To avoid overfitting, the entire model except for the newly added softmax layer is frozen for the first 10k steps, and waveform encoder output is randomly masked with $l = 10$ and $p = 7.5\%$, similar to what has been done during pre-training. The system is implemented in PyTorch [40] with Fairseq [41] and Wav2letter++ decoding [42]. We report the WER on the `dev-other` split decoded with a 4-gram language model (LM), where the LM weight and the word insertion penalty are set to 2 and -1 for all models without tuning.

For analysis, we derive frame-level forced-aligned phonetic transcripts using a hybrid ASR system in order to measure the correlation between the pseudo labels generated by an unsupervised teacher and the true phonetic units. Given frame-level phonetic labels y and pseudo labels z, the joint distribution between the two variables $p(y, z)$ can be estimated by counting the occurrences. For each phone class y, we further compute the most likely pseudo label as $z^*_y = \arg\max_z p(z \mid y)$, and for each pseudo class z most likely phone label $y^*_z = \arg\max_y p(y \mid z)$. Three metrics are considered: (1) **phone purity** (Phn Pur.), $\mathbb{E}_{p(z)}[p(y^*_z \mid z)]$, measuring the average phone purity within a pseudo class, (2) **cluster purity** (Cls Pur.): $\mathbb{E}_{p(y)}[p(z^*_y \mid y)]$, measuring the average pseudo label purity within a phone class, and (3) **phone-normalized mutual information** (PNMI), $I(y; z)/H(y)$, measuring the portion of the uncertainty about the phone eliminated when given a pseudo label.

5. RESULTS

5.1. A Bad Teacher Is Useful with the Right Task

We first determine what the right task is by varying the regions where the loss is computed. Three values of mask weights (α=$\{1.0, 0.5, 0.0\}$) are considered, representing loss computation from masked-only, both, and unmasked-only regions, respectively. Results shown in Table 1 indicate that computing loss only from the masked regions achieves the best performance, while inclusion of unmasked loss results in higher WERs.

Table 2 reports the quality and the performance with $\alpha = 1.0$ for the three k-means teachers in Table 1 and three additional teachers of widely different quality. "Random" teacher assigns a random label drawn from a uniform distribution to each frame, representing a random baseline. "GMM" teacher fits the data with a diagonal-covariance Gaussian mixture model to discover clusters, representing a better unsupervised teacher. Finally, "Chenone" teacher uses a character-based hybrid ASR system trained on the full LibriSpeech data to derive forced-aligned chenone transcripts [43], which represents a supervised topline. It can be seen that fine-tuning performance correlates well with the teacher's quality during pre-training, and varying the number of clusters does not affect the performance much when clustering with k-means on MFCC features.

teacher	C	dev-other WER (%)		
		$\alpha = 1.0$	$\alpha = 0.5$	$\alpha = 0.0$
K-means	50	18.68	31.07	94.60
	100	17.86	29.57	96.37
	500	18.40	33.42	97.66

Table 1: Comparison of where to compute prediction losses.

teacher	C	Phn Pur.(%)	Cls Pur.(%)	PNMI	WER (%)
Random (bottom line)	100	17.56	1.08	0.000	100.00
Chenone (top line)	8976	79.19	22.02	0.809	10.38
K-means	50	31.76	15.12	0.227	18.68
	100	33.26	9.08	0.243	17.86
	500	35.27	2.66	0.276	18.40
GMM	100	35.50	11.14	0.303	16.95

Table 2: How teacher quality affects pre-training.

Figure 1 and Table 3 studies how hyperparameters affect pre-training. It is shown that (1) the portion of frames selected as mask start is optimal at p =9%; (2) increasing the batch size can significantly improve the performance; (3) training for longer consistently helps for both k-means teachers with C={50, 100}, and the best model achieves a WER of 11.68%. These findings are also consistent with those from BERT-like models [2]. In addition, we include a comparable result from DiscreteBERT [10] in Table 3 which applies K-means to quantize the same MFCC features into 13.5k units, used as both the output and the *input* to a BERT model. We hypothesize that HUBERT achieves significantly better performance because more appropriate numbers of clusters are used and raw data are taken as input, which does not discard information.

teacher	C	dev-other WER (%)			
		steps=100k	250k	400k	800k
K-means	50	18.68	13.65	12.40	11.82
	100	17.86	12.97	12.32	11.68
[10]	13.5k	26.6			

Table 3: Varying the number of pre-training steps. p is set to 6.5%.

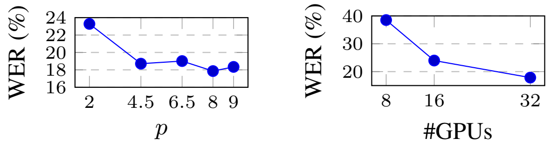

Fig. 1: Varying masking probability p (left) and effective batch size through the number of GPUs (right) with k-means teachers.

5.2. Having Multiple Bad Teachers Can Help

In this section, we demonstrate that using multiple bad teachers can improve the performance from using any single of those alone. Two sets of teachers are considered. The first one is the k-means teachers with different numbers of clusters presented in Table 1, denoted with KM-{50,100,500}. For the second set, we consider spliced MFCC features with a window of three; hence, each frame is represented as a 117 dimensional vector. Furthermore, we apply product quantization on the spliced features, where dimensions are split into the coefficients of the zeroth, first, and second order derivatives and each subspace is quantized with a codebook of 100 entries. We denote these codebooks with PKM-{0,1,2}-100 respectively.

Results are shown in Table 4, where the "untied" column specifies whether a separate projection matrix is used for each teacher when a teacher ensemble is adopted (Sec 2.5). By comparing the numbers from Table 2, it can be observed that using an unsupervised teacher ensemble can lead to better performance than what a single teacher can achieve, and untying projection matrices is sometimes beneficial.

teacher	WER (tied)	WER (untied)
K-means {50,100}	18.17	17.81
K-means {50,100,500}	17.46	17.56
Product K-means-0-100	19.26	N/A
Product K-means-1-100	17.64	N/A
Product K-means-2-100	18.46	N/A
Product K-means-{0,1,2}-100	17.63	16.73

Table 4: Teacher ensemble with k-means and product k-means.

5.3. A Student Can Become a Better Teacher

We take a model pre-trained for 250k steps using the k-means teacher with $C = 100$ and extract features from the BERT encoder at different layers, since the features capturing the most phonetic information may not be the from the last layer [44]. The new features are clustered with k-means to produce a new generation of teachers, the quality of which are shown in Table 5, where L-x denotes using features from the x-th layer.

Each of the new teacher is used to pre-train a model for 100k steps. Compared to the first generation unsupervised teachers in Table 2, the second generation teachers are of much higher quality in all three metrics and can benefit from increasing the number of clusters, especially those from the middle layers. Consequently, models pre-trained with the second generation teachers also achieve much lower WERs (12.05% with just 100k steps) and obtain additional performance gain by increasing the number of clusters to 500.

feature	$C = 100$ / $C = 500$			
	Phn Pur. (%)	Cls Pur. (%)	PNMI	WER (%)
L-12	39.17 / 44.01	14.77 / 6.04	0.338 / 0.402	15.14 / 15.47
L-9	46.20 / 55.11	19.65 / 7.56	0.436 / 0.535	13.73 / 13.50
L-6	53.32 / 63.28	23.75 / 9.95	0.504 / 0.614	12.74 / **12.05**
L-3	43.58 / 48.64	16.70 / 6.62	0.411 / 0.476	14.88 / 13.88
L-0	37.87 / 42.77	14.37 / 4.86	0.338 / 0.406	16.23 / 15.56

Table 5: Iterative training. Each model is pre-trained for 100k steps.

We pre-train the best second generation model highlighted in Table 5 for 400k steps and fine-tune it on the 10h/1h/10m splits to examine the effectiveness in the extremely low resource setup. LM decoding parameters are tuned on the `dev` sets via Bayesian optimization[1]. Table 6 displays the results of HUBERT and two recent studies with the same setup and similar model architectures. The proposed HUBERT model outperforms DiscreteBERT by a large margin and is on par with the state-of-the-art wav2vec 2.0 model. Comparing to wav2vec 2.0, the proposed HUBERT model enjoys the benefit of leveraging multiple weak teachers for simple predictive training and has the potential of further boosting the performance through more rounds of teacher refinement.

D_l	dev-clean / dev-other / test-clean / test-other WER (%)		
	DiscreteBERT [10]	wav2vec 2.0 [11]	HUBERT-it2 (400k)
10m	15.7 / 24.1 / 16.3 / 25.2	8.9 / 15.7 / 9.1 / 15.6	9.1 / 15.0 / 9.7 / 15.3
1h	8.5 / 16.4 / 9.0 / 17.6	5.0 / 10.8 / 5.5 / 11.3	5.6 / 10.9 / 6.1 / 11.3
10h	5.3 / 13.2 / 5.9 / 14.1	3.8 / 9.1 / 4.3 / 9.5	3.9 / 9.0 / 4.3 / 9.4

Table 6: Comparison with previously published results.

6. CONCLUSION

This paper presents HUBERT, a speech pre-training framework that can learn for pseudo labels produced by low quality unsupervised teachers. With the right task, the HUBERT model can achieves a WER below 12% when pre-trained on 960hr of LibriSpeech data and fine-tuned on the 10hr Libri-light split using a simple k-means teacher trained on MFCC features. Furthermore, features extracted from a trained HUBERT model can be used to create better k-means teachers to improve the performance. For future work, we plan to combine many of the findings and train the HUBERT model for more steps. In addition, we would also like to evaluate online VQ to integrate pre-training and teacher refinement more seamlessly.

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

7. REFERENCES

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

[2] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[3] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep contextualized word representations," in *NAACL*, 2018.

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

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

[6] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[7] Laurenz Wiskott and Terrence J Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural computation*, vol. 14, no. 4, pp. 715–770, 2002.

[8] Aaron van den Oord, Oriol Vinyals, et al., "Neural discrete representation learning," in *NeurIPS*, 2017.

[9] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *arXiv preprint arXiv:1910.05453*, 2019.

[10] Alexei Baevski, Michael Auli, and Abdelrahman Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," *arXiv preprint arXiv:1911.03912*, 2019.

[11] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[12] George Zavaliagkos and Thomas Colthurst, "Utilizing untranscribed training data to improve performance," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[13] Jeff Ma, S. Matsoukas, O. Kimball, and Richard Schwartz, "Unsupervised training on large amounts of broadcast news data," in *ICASSP*, 2006.

[14] Jacob Kahn, Ann Lee, and Awni Hannun, "Self-training for end-to-end speech recognition," in *ICASSP*, 2020.

[15] Wei-Ning Hsu, Ann Lee, Gabriel Synnaeve, and Awni Hannun, "Semi-supervised speech recognition via local prior matching," *arXiv preprint arXiv:2002.10336*, 2020.

[16] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018.

[17] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[18] Jacob Kahn et al., "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP*, 2020.

[19] Mikolaj Kegler, Pierre Beckmann, and Milos Cernak, "Deep speech inpainting of time-frequency masks," in *INTERSPEECH*, 2020.

[20] Chia-ying Lee and James Glass, "A nonparametric bayesian approach to acoustic model discovery," in *ACL*, 2012.

[21] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, 2020.

[22] Robert M. Gray and David L. Neuhoff, "Quantization," *IEEE transactions on information theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[23] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[24] Wei-Ning Hsu, Yu Zhang, and James Glass, "Learning latent representations for speech generation and transformation," in *INTERSPEECH*, 2017.

[25] Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[26] Wei-Ning Hsu, Yu Zhang, and James Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *NeurIPS*, 2017.

[27] Janek Ebbers, Jahn Heymann, Lukas Drude, Thomas Glarner, Reinhold Haeb-Umbach, and Bhiksha Raj, "Hidden markov model variational autoencoder for acoustic unit discovery.," in *INTERSPEECH*, 2017.

[28] Sameer Khurana, Antoine Laurent, Wei-Ning Hsu, Jan Chorowski, Adrian Lancucki, Ricard Marxer, and James Glass, "A convolutional deep markov model for unsupervised speech representation learning," *arXiv preprint arXiv:2006.02547*, 2020.

[29] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

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

[31] Yu-An Chung and James Glass, "Improved speech representations with multi-target autoregressive predictive coding," *arXiv preprint arXiv:2004.05274*, 2020.

[32] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *ICASSP*, 2020.

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

[34] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP*, 2020.

[35] Po-Han Chi, Pei-Hung Chung, Tsung-Han Wu, Chun-Cheng Hsieh, Shang-Wen Li, and Hung-yi Lee, "Audio albert: A lite bert for self-supervised learning of audio representation," *arXiv preprint arXiv:2005.08575*, 2020.

[36] Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux, "Data augmenting contrastive learning of speech representations in the time domain," *arXiv preprint arXiv:2007.00991*, 2020.

[37] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency, "Self-supervised learning from a multi-view perspective," *arXiv preprint arXiv:2006.05576*, 2020.

[38] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] Fabian Pedregosa et al., "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, 2011.

[40] Adam Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[41] Myle Ott et al., "fairseq: A fast, extensible toolkit for sequence modeling," in *NAACL*, 2019.

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

[43] Duc Le, Xiaohui Zhang, Weiyi Zheng, Christian Fügen, Geoffrey Zweig, and Michael L Seltzer, "From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition," in *ASRU*, 2019.

[44] Ian Tenney, Dipanjan Das, and Ellie Pavlick, "Bert rediscovers the classical nlp pipeline," *arXiv preprint arXiv:1905.05950*, 2019.