AN EXPLORATION OF HUBERT WITH LARGE NUMBER OF CLUSTER UNITS AND MODEL ASSESSMENT USING BAYESIAN INFORMATION CRITERION

Takashi Maekaku[1], *Xuankai Chang*[2], *Yuya Fujita*[1], *Shinji Watanabe*[2]

[1]Yahoo Japan Corporation, Tokyo, JAPAN
[2]Carnegie Mellon University, PA, USA

ABSTRACT

Self-supervised learning (SSL) has become one of the most important technologies to realize spoken dialogue systems for languages that do not have much audio data and its transcription available. Speech representation models are one of the keys to achieving this, and have been actively studied in recent years. Among them, Hidden-Unit BERT (HuBERT) has shown promising results in automatic speech recognition (ASR) tasks. However, previous studies have investigated with limited iterations and cluster units. We explore HuBERT with larger numbers of clusters and iterations in order to obtain better speech representation. Furthermore, we introduce the Bayesian Information Criterion (BIC) as the performance measure of the model. Experimental results show that our model achieves the best performance in 5 out of 8 scores in the 4 metrics for the Zero Resource Speech 2021 task. It also outperforms the HuBERT `BASE` model trained with 960-hour LibriSpeech (LS) even though our model is only trained with 100-hour LS. In addition, we report that BIC is useful as a clue for determining the appropriate number of clusters to improve performance on phonetic, lexical, and syntactic metrics. Finally, we show that these findings are also effective for the ASR task.

Index Terms— HuBERT, unit-based language model, self-supervised learning, acoustic unit discovery, BIC

1. INTRODUCTION

In recent years, studies on the application of self-supervised learning to speech processing have been proposed to realize a spoken dialogue system from scratch using only sensory information like the auditory information available to an infant [1, 2, 3, 4]. Among them, a language model (LM) trained using only raw speech audio has become one of the most important topics to address for languages that lack textual information. The Zero Resource Speech (ZeroSpeech) Challenge 2021 [2] is designed to address the improvement of such LM training. The baseline system applies contrastive predictive coding (CPC) [5] to extract a feature representation from raw speech. Then it trains a k-means clustering on the outputs of CPC to obtain discrete units for the input of a unit-based LM (uLM). The evaluation is done at 4 linguistic levels: phonetic, lexicon, syntax, and semantic. As an extension of this challenge, Lakhotia *et al.* built a speech synthesizer that takes discrete units as input in addition to uLM and established a set of metrics at the acoustic and linguistic levels to evaluate each component [3].

To realize ZeroSpeech processing, self-supervised learning has become a more promising research direction, and various methods of speech representation models have been proposed, such as autoregressive predictive coding (APC) [6], Mockingjay [7], wav2vec2.0 [8], and Hidden-Unit BERT (HuBERT) [9] in addition to CPC. Among them, HuBERT has been reported to show competitive or better performance than other models on the automatic speech recognition (ASR) and other downstream tasks [9, 10, 11]. HuBERT trains a masked prediction task [12] using discrete units obtained by clustering mel-frequency cepstrum coefficients (MFCC) of input speech as pseudo-labels. This model is able to capture a better representation by updating the units by re-clustering the representation feature obtained from the previously trained model iteratively. Due to the nature of the training method, it is expected that the variety of cluster units would have a big impact on the performance of downstream tasks. However, previous studies [9, 3] have evaluated the linguistic metrics or ASR tasks with limited iterations and cluster units (e.g., up to 2 iterations and 500 units). Considering the variety of lexical units, the number of cluster units would not be sufficient.

This paper proposes to explore HuBERT clustering configurations (mostly number of cluster units and iterations) and model assessment using Bayesian Information Criterion (BIC) [13]. We explore the case of a large increase in the number of clusters up to around 2000 units with 3 iterations. In order to see the effectiveness from a wide range of acoustic and linguistic perspectives, we evaluate our methods on the ZeroSpeech 2021 Challenge task in this work. Although an extensive exploration of the number of cluster units is needed to achieve optimal performance, it is very difficult to conduct such exploration since the uLM training and the ZeroSpeech metrics are computationally expensive. To avoid the expensive evaluation for each number of cluster units, we introduce BIC to efficiently find the optimal number of cluster units before the uLM training. Experimental results show that our system achieves the best performance among the systems submitted to INTERSPEECH 2021 [14, 15, 16, 17] with 5 out of 8 scores in the 4 metrics. It also outperforms the HuBERT `BASE` [1] model which is publicly available and trained on LibriSpeech(LS) 960 hours even though it is only trained on LS 100 hours. We demonstrate that BIC is useful as a clue for determining the appropriate number of clusters to improve performance on phonetic, lexical, and syntactic metrics. Finally, we show that these findings are also effective for the ASR task.

2. ZEROSPEECH 2021 BASELINES BASED ON CPC

2.1. Baseline Systems

The ZeroSpeech 2021 baseline system consists of speech representation learning, clustering, and language models.

2.1.1. Contrastive Predictive Coding

The speech representation model is based on CPC, a self-supervised representation learning method proposed in [5]. The CPC network consists of a convolutional neural network (CNN) encoder g_{enc} and an autoregressive network g_{ar}. The CPC model is trained by

[1] https://github.com/pytorch/fairseq/blob/main/examples/hubert/

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

maximizing the mutual information between the current context vector and future embeddings by minimizing the noise-contrastive estimation-based (NCE) loss [18].

There are two different versions of the CPC model as baseline systems: CPC-small which has a 2-layer long short-term memory (LSTM) as g_{ar} and 256 hidden units, and CPC-big which has a 4-layer LSTM and 512 hidden units.

2.1.2. Clustering and Language Models

To train a uLM, the raw speech signal needs to be mapped to a sequence of discrete units. The pre-trained CPC model first generates a sequence of representations given the raw speech. Then, a subset of these representations are used to apply k-means with 50 clusters.

After training, the clustering model is applied to the speech representation of the training data to produce discrete units. Using these units as pseudo-text, we can train the uLM. In this work, we trained a BERT LM [19]. This model consists of multiple Transformer layers. Note that it is only trained with the masked language model objective, following [20]. Finally, the score of the uLM on the unit sequence is regarded as a pseudo-probability (PP).

2.2. Dataset

The training data is comprised of the audio from the LibriSpeech 960h dataset [21] and the Libri-light dataset [22]. The k-means clustering is performed on the train-clean-100h subset to obtain the centroid coordinates. Then the k-means estimates the unit sequences on LibriSpeech 960h data, which is the training set for the uLM.

Evaluation is performed on its *dev* set, which are specially designed for the corresponding task. Please refer to the challenge description [2, 14] for more details.

2.3. Evaluation Metrics

The performance of the uLM is evaluated using 4 different metrics as follows:

Phonetic. The ABX error [23] discriminates the speech sound between phonetic minimal pairs (e.g. "aba" and "apa"). Given the speech sounds a, x and b, where a and b are from two categories A and B ($A \neq B$), and x belongs to category A respectively, it computes the probability that the two sounds from the same category are closer than the two sounds from different categories. We evaluate it under 2 conditions, "within" (all stimuli a, b and x are uttered by the same speaker) and "across" (a and b are from the same speaker, and x from a different speaker).

Lexicon. The sWUGGY "Spot-the-word" [24] is used to discriminate an existing word from a lexically similar non-word using the uLM (e.g. "brick" and "blick"). The metric measures the percentage of PP of the real words that are higher than that of non-word among all word pairs.

Syntax. sBLIMP acceptability, adapted from BLIMP [25], discriminates a grammatical sentence from an ungrammatical sentence (e.g. "dogs eat meat" and "dogs eats meat"). The metric is calculated in the same way as the sWUGGY metric.

Semantic. sSIMI similarity measures the similarity between the representations of pairs of words and compares the results by human judgment. The metric is computed as the Spearman's rank correlation coefficient ρ between the semantic similarity scores given by the model and the human scores in the dataset. In this work, it is evaluated using synthetic speech data "*synth.*" and a subset of the LibriSpeech corpus "*libri.*".

3. PROPOSED METHODS

3.1. HuBERT with large cluster units

3.1.1. HuBERT

HuBERT [9] is a representation learning method based on masked prediction and it consists of a CNN encoder and a Transformer encoder. During pre-training, k-means clustering is first executed on T frames of MFCC input features $X = [x_1, \cdots, x_T]$ to obtain discrete units as pseudo-labels. These units are denoted as $Z = [z_1, \cdots, z_T]$, where $z_t \in \{1, \cdots, U\}$ and U is the number of cluster units. Let $M \subset [T]$ denote the set of indices to be masked and $\tilde{X}$ denote a masked version of X. $\tilde{X}$ is fed to the masked prediction model f which predicts a distribution over the target indices at each time step $p_f(\cdot|\tilde{X}, t)$. Finally, we compute the cross-entropy loss on only the masked regions as follows:

$$L(f; X, M, Z) = -\sum_{t \in M} \log p_f(z_t|\tilde{X}, t). \tag{1}$$

This model is able to capture a better representation by updating the units by iteratively re-clustering the feature representation obtained from the previously trained model. During inference, we extract outputs from the last layer of the Transformer encoder as a representation feature.

We replace CPC, as discussed in Section 2, with HuBERT in the baseline system. As HuBERT performed better than CPC in previous studies [3, 11], it is expected that HuBERT is superior in terms of the ZeroSpeech Challenge setting.

3.1.2. Large Cluster Units Setting

It is clear that the variety of cluster units would have a big impact on the performance of downstream tasks. However, previous studies [9, 3] have only evaluated the linguistic metrics or ASR tasks up to 2 iterations of clustering with U from 100 to 500. Considering the variety of lexical units, the number of cluster units would not be sufficient. Therefore, we further explore the impact of larger U and iterations. We consider U from 100 to 2100 and train each model up to 3 iterations. Following [9], we extract outputs of the 6th layer of the Transformer encoder at the 1st iteration and extract the outputs of the 9th layer of the encoder at the 2nd iteration to update the units. And we use outputs of the 12th layer of the encoder at the 3rd iteration.

However, this intensive exploration of U is very difficult since the uLM training and the ZeroSpeech metrics in Section 2.3 are computationally expensive. To avoid the expensive evaluation for each U, we introduce the Bayesian Information Criterion (BIC) to efficiently find the optimal U before the uLM training.

3.2. Assessing Quality of uLM Input Using BIC

Before the uLM training, we calculate the BIC score to assess the representation quality in terms of HuBERT feature and k-means centroids used for producing discrete units for uLM[2]. BIC is one of the criteria used for statistical model selection, and it has been widely applied in the field of ASR tasks [27, 28, 29]. It consists of a log-likelihood term and a penalty term as follows:

$$BIC = -2 \log p(\mathcal{D}|\theta) + S \log N, \tag{2}$$

$$S = 2dU + U - 1, \tag{3}$$

[2] As a preliminary experiment, we calculated BIC in terms of MFCC feature and k-means centroids for HuBERT training. However, we could not find any local minimum point. We concluded that it is because the MFCC representation does not follow a Gaussian distribution well.

Table 1. HuBERT model architecture. d^{att}, d^{ff}, p^{LD}, and H denote attention dimension, feed forward network dimension, LayerDrop [26] probability, and attention heads, respectively.

CNN Encoder			**Transformer**					**Projection**
strides	kernel size	channel	layer	d^{att}	d^{ff}	p^{LD}	H	dim.
5, 2, 2, 2, 2, 2, 2	10, 3, 3, 3, 3, 2, 2	512	12	768	3072	0.05	8	256

Table 2. Overall performance of our systems. We evaluate ABX error under 2 conditions, "within" (all stimuli a, b and x are uttered by the same speaker) and "across" (a and b are from the same speaker, and x from a different speaker on Libri-light dev-clean and dev-other. All embeddings are extracted from the final layer of the HuBERT encoder network. sSIMI metric is evaluated using synthetic speech data "*synth.*" and a subset of the LibriSpeech corpus "*libri.*".

System	**Feature for k-means**	**ABX within(%) (↓)**		**ABX across(%) (↓)**		**sWUGGY(%) (↑)**	**sBLIMP(%) (↑)**	**sSIMI (↑)**	
		dev-clean	dev-other	dev-clean	dev-other			synth.	libri.
HuBERT BASE (2iter) [9]	BASE 1iter	3.38	**4.42**	4.16	6.89	70.15	55.42	**4.59**	6.19
100-0iter	MFCC	8.93	12.32	14.20	21.00	55.09	53.89	0.51	5.10
500-0iter	MFCC	6.88	9.78	11.54	18.48	50.92	53.28	0.99	6.85
100-1iter	100-0iter	3.13	4.86	3.84	7.84	64.77	54.08	0.84	1.66
500-1iter	100-0iter	3.30	5.23	3.97	7.99	69.38	55.44	2.62	7.94
1100-1iter	100-0iter	3.28	5.08	3.82	7.84	70.50	55.47	-1.62	9.50
1300-1iter	100-0iter	3.34	5.03	3.80	7.68	70.47	55.65	0.60	6.77
1500-1iter	100-0iter	3.21	4.97	3.76	7.75	69.34	55.34	3.35	9.27
1900-1iter	100-0iter	3.25	5.10	3.78	7.80	70.84	55.39	1.23	8.60
2100-1iter	100-0iter	5.33	7.80	6.87	12.74	67.43	53.47	0.22	5.77
500-2iter	1100-1iter	3.43	4.91	3.91	7.11	72.60	56.34	1.22	7.31
1100-2iter	1100-1iter	3.06	4.63	**3.51**	6.72	73.24	55.86	-0.74	4.30
1700-2iter	1100-1iter	3.11	4.62	3.52	6.60	72.24	**56.80**	0.75	6.09
2100-2iter	1100-1iter	4.81	7.47	6.23	11.35	69.31	54.86	0.86	8.86
500-3iter	1100-2iter	**3.02**	4.63	3.56	**6.47**	72.27	55.95	2.86	**12.53**
1100-3iter	1100-2iter	3.07	4.71	3.53	6.63	**73.96**	55.81	-2.35	10.53
2100-3iter	1100-2iter	5.45	8.05	6.85	12.18	67.51	54.39	-0.71	7.16

where $\mathcal{D}$ is the training data used for k-means clustering to produce discrete units for uLM training, θ is parameters of k-means, S is the number of parameters in θ, d is the feature dimension, N is the data size, and U is the number of clusters introduced in 3.1.1. In general, the lower the value of BIC is, the better the performance of the model can be. Since k-means can be regarded as a special case of Gaussian Mixture Model (GMM), we approximate the likelihood of the clustering model with GMM using the centroid of each cluster as a mean.

The computation can be efficiently done on the CPU, and the cost is much lower than the uLM training and calculation of the 4 metrics, which require the GPU. Therefore, we can efficiently find the optimal U before the uLM training. We also investigate which linguistic metric actually correlates with it.

4. EXPERIMENTS

4.1. Experimental Setup

We applied the same architecture as HuBERT BASE [9] and all our models were trained on LibriSpeech 100h. The summary of architecture is given in Table 1. During the initial model training, we used 39-dimensional MFCC features which are 13 coefficients with the first and the second-order derivatives to produce discrete units. To examine a small amount of data carefully, we used 80% of LibriSpeech 100h data for k-means clustering. When training the model for the second and subsequent times, we extracted hidden features from the HuBERT encoder to update the units as mentioned in 3.1.2.

As for uLM, we adopted the "BERT-small" model trained on LibriSpeech 960h and this is also used for the baseline. See [2] for details of the model architecture. To generate discrete units for uLM training, we extracted the outputs of the 12th layer of the encoder using LibriSpeech 960h. When k-means clustering was executed, we used 7% of them from LibriSpeech 960h to obtain the centroid of each cluster. We also calculated BIC scores based on the 7% of them and the centroids. All models were implemented with fairseq[3], and we calculated BIC by modifying scikit-learn[4].

4.2. Exploration among Large Cluster Models

In Table 2, we present the results for 4 linguistic metrics[5]. We also evaluated the case using the BASE pre-trained model for comparison. In this model, U is set to 100 initially and kept constant at 500 for the 1st and 2nd iterations. Regarding the notation of each model name {a}-{b}iter, {a} and {b} represent U at the last iteration and the number of iterations, respectively.

4.2.1. Performance on 4 Metrics

We can see the effectiveness of increasing U as the performance in ABX "across", sWUGGY, and sBLIMP metrics improve until the 2nd iteration[6] . In terms of ABX "within", the increase in U does not contribute much to the performance comparing the case of 1st iteration models. However, in the 2nd iteration setting, all ABX errors improve drastically when U is 1100 and 1700.

And it is obvious that `*-3iter` using the hidden feature of `1100-2iter` for updating cluster units achieves the best results in 4 out of 8 scores. These results suggest that more iterations lead to better speech representation. Eventually, `500-3iter` outperforms BASE on 6 / 8 scores.

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

[4] https://scikit-learn.org/stable/

[5] We first tried to find U with the best performance in the 1st iteration. In the 2nd and 3rd iterations, we further investigated the points where the performance difference was clearly seen.

[6] The high variance in sSIMI metrics could be due to the fact that the task of calculating semantic similarity using only speech was very challenging.

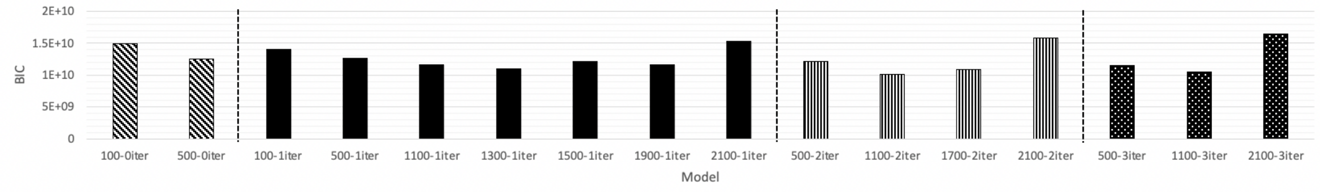

Fig. 1. BIC for all our HuBERT models.

Table 3. Comparison among all submissions for INTERSPEECH 2021. Each model is trained on LibriSpeech (LS) or Libri-light (LL). "Training Data" refers to the dataset for speech representation learning.

System	Training Data	Use of Speaker Info	ABX within(%) (↓)		ABX across(%) (↓)		sWUGGY(%) (↑)	sBLIMP(%) (↑)	sSIMI (↑)	
			dev-clean	dev-other	dev-clean	dev-other			synth.	libri.
Baseline : CPC-small [2]	LS-100h		6.24	8.48	8.17	13.55	65.79	52.88	-0.09	9.23
Baseline : CPC-big [2]	LL-6kh		3.41	4.85	4.18	7.64	65.81	52.91	3.88	5.56
van Niekerk *et al.* [15]	LL-6kh	✓	5.38	8.80	6.56	12.79	72	54	4.29	7.69
Chorowski *et al.* [16] System 1	LS-100h	✓	**2.94**	**4.47**	3.58	7.02	73.92	53	-7.75	4.60
Chorowski *et al.* [16] System 2	LS-100h	✓	3	5	4	8	72	52	**5.90**	10.20
Liu *et al.*	LS-100h		17	20	25	30	52	52	3.16	1.79
Maekaku *et al.* [17] System 1	LL-6kh		3.11	4.96	3.98	7.92	62.64	54.06	-1.65	4.81
Maekaku *et al.* [17] System 2	LL-6kh		3.28	4.96	4.14	8.28	66.01	54.15	-0.81	5.45
Ours: 1100-3iter	LS-100h		3.07	4.71	**3.53**	**6.63**	**73.96**	**55.81**	-2.35	**10.53**

Table 4. ASR evaluation on WSJ dataset. Comparison between log filter-bank features (FBANK), HuBERT BASE model and our HuBERT models. Word error rates are shown for dev/test sets.

Model	dev93 WER (%)	eval92 WER (%)
FBANK	9.1	5.7
100-0iter	7.9	5.7
500-3iter	7.3	5.4
1100-2iter	7.3	5.1
1100-3iter	7.3	**4.9**

4.2.2. Considerations in BIC

Fig 1 compares the BIC scores with all our HuBERT models. Comparing between 100-0iter and 500-0iter, which both use MFCC for updating units, we find that the BIC score decreases as ABX error improves, while sWUGGY and sBLIMP metrics degrade. Therefore BIC is correlated with the performance of the ABX error in this case.

On the other hand, as for other models which use a hidden feature of HuBERT, BIC is strongly correlated with the sWUGGY metric in addition to ABX error. Especially in the case of the 2nd and 3rd iterations, BIC decreases as each sWUGGY metric improves when U is increased from 500 to 1100. And BIC increases as each sWUGGY metric degrades when U is increased from 1100 to 2100. Moreover, the same trend can be seen for the results of ABX error on the dev-clean set in the case of the 2nd iteration, and ABX "across" on the dev-clean set in the case of the 3rd iteration. In addition, as for sBLIMP metric, there is a tendency for U to have a maximum value between 1100 and 1700 in the case of the 1st and 2nd iterations. It indicates that we can determine the optimal range of U per iteration on ABX, sWUGGY, and sBLIMP metrics using BIC without training the uLM and checking the final results.

4.3. Comparison among all submissions

In Table 3, we choose 1100-3iter, which has the lowest value of BIC in ⋆-3iter and compare it to the best models of all the participants in the challenge[7] for INTERSPEECH 2021 [14, 15, 16, 17][8]. Niekerk *et al.* [15], Chorowski *et al.* [16], and Maekaku *et al.* [17] used CPC while Liu *et al.* used Mockingjay for the speech representation model. Note that while 2 systems [15, 16] use speaker information in addition to speech, we do not use any of it. We can see that our system achieves the best performance in 5 out of 8 scores even though we use only LibriSpeech 100h for speech representation learning. Interestingly, we find that our model has a high rate of improvement in terms of ABX "across". It indicates that the HuBERT model is robust to differences in features between speakers.

4.4. ASR evaluation

Finally, to further verify the performance of learned speech representation of our models, we trained end-to-end (E2E) ASR systems based on some of the HuBERT models, using the Wall Stret Journal (WSJ) corpus for the purpose of fast adaption. We adopted the conformer-based E2E ASR model architecture in [30]. We extracted weighted-averaging hidden states from the HuBERT models as features to replace the commonly used 80-dimensional log filterbank. We mapped the HuBERT features to 80-dimensions before feeding them to the encoder. We show the performance in Table 4. It is observed that the performance improves as U and the number of iterations increase, and this shows a similar trend to the evaluation results of the ZeroSpeech task.

5. CONCLUSION

This paper proposed a HuBERT based uLM with a large number of cluster units and iterations and the model assessment using BIC. We demonstrated the importance of increasing cluster size and the number of iterations in HuBERT through the ZeroSpeech task. Moreover, we introduced BIC as a measure of model selection. We showed that our models achieved the best performance in 5 out of 8 scores in the 4 metrics for the ZeroSpeech task. We found that BIC could be useful as a clue for determining the appropriate number of clusters to improve the phonetic, lexical, and syntactic performances without training uLM beforehand. We also evaluated the model on an ASR task and showed its effectiveness.

[7] In the challenge leaderboard, the number of significant digits is displayed as 1 digit for ABX error and 2 digits for sWUGGY and sBLIMP metrics. Therefore the number of significant digits in each score differs depending on the information provided in the participant's paper.

[8] Please note that ZeroSpeech Challenge2021 was held twice, once for INTERSPEECH 2021 and once for AAAI 2022 workshop.

6. REFERENCES

[1] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2020: Discovering discrete subword and word units," in *Proc. Interspeech*, 2020.

[2] T. A. Nguyen, M. d. Seyssel, P. Rozé, M. Rivière, E. Kharitonov, A. Baevski, E. Dunbar, and E. Dupoux, "The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," in *NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*, 2020.

[3] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, et al., "On generative spoken language modeling from raw audio," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1336–1354, 2021.

[4] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," in *Advances in NeurIPS*, 2021, vol. 34.

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

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

[7] 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 *Proc. ICASSP*, 2020, pp. 6419–6423.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in NeurIPS*, 2020, vol. 33.

[9] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "HuBERT: How much can a bad teacher benefit ASR pre-training," in *NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*, 2020.

[10] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, et al., "SUPERB: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[11] X. Chang, T. Maekaku, P. Guo, J. Shi, Y.-J. Lu, A. S. Subramanian, T. Wang, S. w. Yang, Y. Tsao, H. y. Lee, and S. Watanabe, "An exploration of self-supervised pretrained representations for end-to-end speech recognition," in *Proc. ASRU*, 2021.

[12] J. D. M.-W. C. Kenton and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[13] G. Schwarz, "Estimating the dimension of a model," *The annals of statistics*, pp. 461–464, 1978.

[14] E. Dunbar, M. Bernard, N. Hamilakis, T. A. Nguyen, M. d. Seyssel, P. Rozé, M. Rivière, E. Kharitonov, and E. Dupoux, "The zero resource speech challenge 2021: Spoken language modelling," in *Proc. Interspeech*, 2021, pp. 1574–1578.

[15] B. v. Niekerk, L. Nortje, M. Baas, and H. Kamper, "Analyzing speaker information in self-supervised models to improve zero-resource speech processing," in *Proc. Interspeech*, 2021, pp. 1554–1558.

[16] J. Chorowski, G. Ciesielski, J. Dzikowski, A. Łańcucki, R. Marxer, M. Opala, P. Pusz, P. Rychlikowski, and M. Stypułkowski, "Information retrieval for zerospeech 2021: The submission by university of wroclaw," in *Proc. Interspeech*, 2021, pp. 971–975.

[17] T. Maekaku, X. Chang, Y. Fujita, L.-W. Chen, S. Watanabe, and A. Rudnicky, "Speech representation learning combining conformer CPC with deep cluster for the zerospeech challenge 2021," in *Proc. Interspeech*, 2021, pp. 1564–1568.

[18] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. AISTATS*, 2010, pp. 297–304.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in NeurIPS*, 2017, pp. 6000–6010.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

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

[22] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, et al., "Libri-Light: A benchmark for ASR with limited or no supervision," in *Proc. ICASSP*, 2020, pp. 7669–7673.

[23] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair ABX task: Analysis of the classical MFC/PLP pipeline," in *Proc. Interspeech*, 2013, pp. 1006–1010.

[24] G. Le Godais, T. Linzen, and E. Dupoux, "Comparing character-level neural language models using a lexical decision task," in *Proc. EACL*, 2017, pp. 125–130.

[25] A. Warstadt, A. Parrish, H. Liu, A. Mohananey, W. Peng, S.-F. Wang, and S. Bowman, "BLiMP: The benchmark of linguistic minimal pairs for english," *Transactions of the ACL*, vol. 8, pp. 377–392, 2020.

[26] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," *arXiv preprint arXiv:1909.11556*, 2019.

[27] W. Chou and W. Reichl, "Decision tree state tying based on penalized bayesian information criterion," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1999, vol. 1, pp. 345–348.

[28] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *Acoustical Science and Technology*, vol. 21, no. 2, pp. 79–86, 2000.

[29] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*, Cambridge University Press, 2015.

[30] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, et al., "Recent developments on ESPnet toolkit boosted by conformer," *arXiv preprint arXiv:2010.13956*, 2020.