ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747259

SELF SUPERVISED REPRESENTATION LEARNING WITH DEEP CLUSTERING FOR ACOUSTIC UNIT DISCOVERY FROM RAW SPEECH

Varun Krishna, Sriram Ganapathy

Learning and Extraction of Acoustic Patterns (LEAP) Lab, Indian Institute of Science, Bangalore.
varunkrishna@iisc.ac.in, sriramg@iisc.ac.in

ABSTRACT

The automatic discovery of acoustic sub-word units from raw speech, without any text or labels, is a growing field of research. The key challenge is to derive representations of speech that can be categorized into a small number of phoneme-like units which are speaker invariant and can broadly capture the content variability of speech. In this work, we propose a novel neural network paradigm that uses the deep clustering loss along with the autoregressive contrastive predictive coding (CPC) loss. Both the loss functions, the CPC and the clustering loss, are self-supervised. The clustering cost involves the loss function using the phoneme-like labels generated with an iterative k-means algorithm. The inclusion of this loss ensures that the model representations can be categorized into a small number of automatic speech units. We experiment with several sub-tasks described as part of the Zerospeech 2021 challenge to illustrate the effectiveness of the framework. In these experiments, we show that proposed representation learning approach improves significantly over the previous self-supervision based models as well as the wav2vec family of models on a range of word-level similarity tasks and language modeling tasks.

Index Terms— Self-supervised learning, Representation learning, Contrastive Predictive Coding, Deep clustering, ZeroSpeech challenge.

1. INTRODUCTION

The area of textless natural language processing (NLP) involves using raw speech data, without any text or labels, for various information extraction tasks [1] like spoken language modeling, speech recognition and speech synthesis. At the core of these modeling methods, is the sub-problem of automatic sub-word unit discovery of speech [2]. This problem is the identification of fundamental units that allow the modeling of the wide variety of spoken content.

The early approaches to automatic unit discovery used dynamic time warping (DTW) based cluster templates, proposed by Wilpon et. al. [3]. Further, following the trends in speech recognition, acoustic unit discovery based on hidden Markov model (HMM) was investigated by Lee et. al. [4] and Varadarajan et. al. [5]. The zerospeech challenges [6, 7, 8, 9] have propelled this area of research to derive automatic units of speech. The performance metrics used in these challenges, include a variety of zero shot metrics for probing the quality of the learned models at the acoustic level and the linguistic level. The phonetic quality of representations are measured using ABX similarity metrics, while the semantic quality is measured as the correlation between the human and model scores [9]. The lexical and syntactic abilities of the model are measured in-terms of unnormalized probability scores output by the language models.

The approaches pursued in zero speech challenges include Gaussian mixture modeling by Heck et. al. [10], HMMs by Ansari et. al. [11], and deep learning methods [12, 13]. In the recent years, self-supervised learning methods, a method of generating pseudo labels from raw data itself [14], have been increasingly used for automatic unit discovery. These efforts focus on the construction of suitable model architectures, like convolutional networks [15], recurrent networks [16], transformer models [17], and conformer models [18], as well as the choice of suitable cost functions, like contrastive [19], vector quantization [20], clustering [18] and autoregressive predictive coding [21].

The wav2vec family of models [22, 23, 24] use self-supervision for unit discovery applied to various downstream tasks like speech recognition and synthesis. The first model, wav2vec 1.0 [22], is similar to the contrastive predictive coding [25], except that the 1-D convolution layers are used instead of long short term memory (LSTM) network layers. The subsequent model, using vector-quantization, wav2vec-vq [23], is inspired by the use of quantization module proposed by Oord et. al. [26]. The output of the quantization module can be then leveraged to train language models like bidirectional encoder representations from transformer (BERT) [27]. The recent extension, the wav2vec 2.0 [24], improves the quantization module with learnable codebooks along with the introduction of the diversity loss. Most of these approaches use a separate clustering model (like k-means) on the embeddings derived from the representation learning network for language modeling tasks which is not optimized with the representation learning model.

In this paper, we explore a novel approach for joint self-supervised representation learning and unit discovery inspired by the deep clustering framework [28]. The deep clustering module jointly learns the parameters of the representation learning neural network and the cluster assignments on the learned representations. Our proposed approach starts with the initial embeddings from the CPC model, which are trained using the CPC loss. The unsupervised cluster assignments on these embeddings act as pseudo-labels for subsequent self-supervised learning algorithm. This clustering loss can also be combined with the contrastive loss to derive rich representations that are categorical. Using the data from the Zerospeech 2021 challenge, we show that the proposed approach of deep clustering and self-supervision improves the performance metrics of the phonetics (ABX), syntactic and the semantic modeling. Further, the visualization of the embeddings highlights the ability of the proposed model to succinctly capture the speech characteristics.

2. BACKGROUND

The prior works related to the proposed approach can be divided into two broad directions, one based on autoregressive predictive

Authorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on June 13,2022 at 04:49:36 UTC from IEEE Xplore. Restrictions apply.

coding [21] and the wav2vec models which contain a masked representation learning in the discrete latent space [24]. Both these paradigms learn using the contrastive loss.

2.1. Contrastive predictive coding

The CPC model [25] consists mainly of 2 layers. The first layer performs a non-linear encoding of the input features. This is based on convolutional networks with 1-D kernels. Next, an autoregressive model summarizes all the encoded outputs in the latent space and produces a context representation. The model predicts the density ratio, which preserves the mutual information between the context vector and the future input time-series.

The raw audio samples $x_\lambda \in \mathcal{X}$, are forward passed through convolutional layer $f : \boldsymbol{\mathcal{X}} \rightarrow \boldsymbol{\mathcal{Z}}$, where $\boldsymbol{\mathcal{Z}}$ represents the the learnable features. This block outputs the low frequency representation $\boldsymbol{z}_t \in \boldsymbol{\mathcal{Z}}$ which is sampled at 30 ms with a stride of 10 ms. The features $\boldsymbol{z}_t$ are then fed to the LSTM layer $\mathcal{G}$. For each time step t , LSTM layers aggregate $(\boldsymbol{z}_t, \boldsymbol{z}_{t-1}, ..., \boldsymbol{z}_{t-v})$ to generate the context vector $\boldsymbol{c}_t$ with a receptive field v. The CPC model loss is based on the prediction of the K future embeddings $\{\boldsymbol{z}_{t+k}\}_{1\leq k\leq K}$ by minimizing the following contrastive loss [26],

$$\mathcal{L}_t = -\frac{1}{K}\sum_{k=1}^{K} log\left(\frac{exp(\boldsymbol{z}_{t+k}^T \boldsymbol{W}_k \boldsymbol{c}_t)}{\sum_{\tilde{\boldsymbol{z}}\in\mathcal{N}_t} exp(\tilde{\boldsymbol{z}}^T \boldsymbol{W}_k \boldsymbol{c}_t)}\right) \quad (1)$$

Here, $\mathcal{N}_t$ is a random subset of negative embedding samples, and W_k is a linear classifier used to predict the future k-step observations.

2.2. wav2vec models

The wav2vec models [22, 23, 24] also have an initial 1-D convolutional network based encoder $f : \mathcal{X} \mapsto \mathcal{Z}$. The sampling of the representations $\boldsymbol{z}_t$ vary according to the model. The wav2vec [24] and wav2vec-vq [23] output $\boldsymbol{z}_t$ at 100Hz, while the wav2vec 2.0 outputs representations at 49Hz. These models differ in the way they handle the representations $\boldsymbol{z}_t$ subsequently.

2.2.1. wav2vec

The representations $\boldsymbol{z}_t$, obtained from the encoder, are fed to another multi-layered convolutional neural network, called the aggregator $\mathcal{G}$, that combines the encoder outputs of the multiple time steps into a new representation $\boldsymbol{c}_t$, for each time step t. Given an aggregated representation $\boldsymbol{c}_t$, the model is trained to distinguish a sample $\boldsymbol{z}_{t+k}$ from distractor samples $\tilde{z}$, by minimizing the contrastive loss,

$$\mathcal{L}_k = -\sum_{t=1}^{T-k} log(\sigma(\boldsymbol{z}_{t+k} h_k(\boldsymbol{c}_t) + \lambda \sum_{\tilde{\boldsymbol{z}}\in\mathcal{N}_t} log(\sigma(\tilde{\boldsymbol{z}}^T h_k(\boldsymbol{c}_t)) \quad (2)$$

Here, $\mathcal{N}_t$ is the set of distractor samples, h_k defines an affine transform and λ is a hyper-parameter.

2.2.2. VQ-wav2vec

This model learns vector quantized representations from raw audio using future prediction task. Architecturally it is same as wav2vec with two convolutional networks $f : \mathcal{X} \mapsto \mathcal{Z}$ and $g : \tilde{\mathcal{Z}} \mapsto \mathcal{C}$ for feature extraction and aggregation. However, the model contains additional quantization module $q : \mathcal{Z} \mapsto \tilde{\mathcal{Z}}$

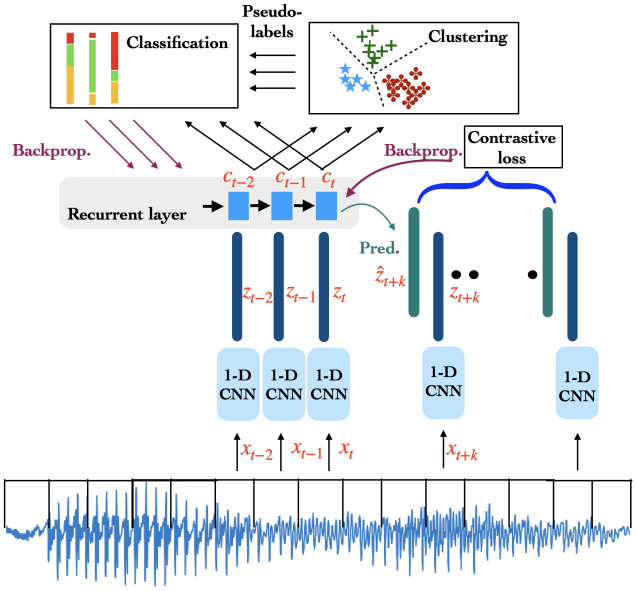

Fig. 1. Schematic of the proposed approach for automatic unit discovery. The final cluster indices are used as the acoustic sub-word unit representations of the speech.

2.2.3. wav2vec 2.0

In this model, the output of the feature encoder $\boldsymbol{z}_t$ is fed to a context network which follows the transformer architecture [24]. For self-supervised training, discretized outputs are obtained using the product quantization [29] of the feature encoder. The product quantization amounts to using multiple codebooks and concatenating the quantized representations from all of them. While training, masking a proportion of feature encoder output is done before feeding them to the context networks. The model was trained to minimize the combination of contrastive loss and the diversity loss that helps to avoid the mode collapse problem.

3. OUR FRAMEWORK

The schematic of the proposed framework is shown in Figure 1. The initial processing steps are inspired by the contrastive predictive coding model [25]. The convolutional layers generate features $\boldsymbol{z}_t$ and recurrent layers generate context vectors $\boldsymbol{c}_t$. The context vectors are clustered to generate the psuedo-labels for self-supervision.

3.1. Self-Supervised learning using pseudo labels

Our work is inspired by the self supervised learning technique described in [28]. In this prior work, the output of the convolution layer was clustered using k-means. Subsequently, these cluster assignments were used as pseudo-labels. The convolutional network then updates the weights by minimizing the classification loss. The cluster updates and the weight updates happen in an alternating fashion. In our work, we explore a similar algorithm where the representations are clustered and the pseudo labels are used in discriminative learning framework. The rationale behind this approach is to exploit the weak supervision labels to bootstrap the discriminative power of the representation learning network and to encourage the representations to form succinct cluster categories. We found this approach

Authorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on June 13,2022 at 04:49:36 UTC from IEEE Xplore. Restrictions apply.

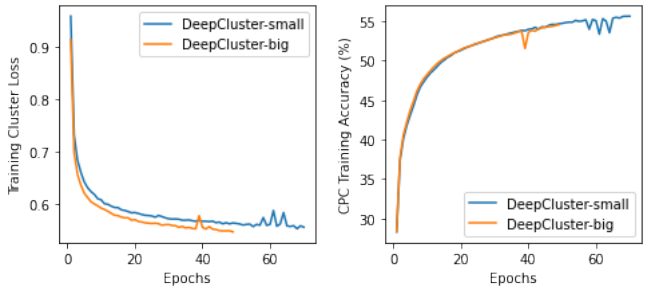

Fig. 2. The plot of the two loss functions used in the proposed approach. As seen here, both the cluster loss (left) and the CPC accuracy (right) are aligned well in the model training.

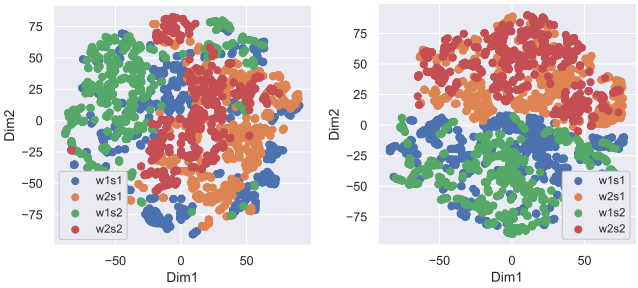

Fig. 3. t-SNE embeddings of representations $\boldsymbol{c}_t$ derived from two phoneme units w_1 and w_2 and from two speakers s_1 and s_2. The embeddings from the baseline CPC model (left) have more overlap of within speaker across phoneme representations, while the embeddings from the proposed approach (right) cluster the same phonemes across the two speakers while also being discriminative.

to train the model beneficial in terms of several evaluation metrics.

3.2. Architecture

The model architecture, shown in Figure 1, consists of initial processing steps similar to the CPC model described in Sec. 2.1. The first two processing steps consist of the non linear encoder $f : \mathcal{X} \mapsto \mathcal{Z}$, and the autoregressive model $\mathcal{G}$. At each time step t, the autoregressive model $\mathcal{G}$ takes as input the available embeddings $\boldsymbol{z}_1, \boldsymbol{z}_2 .... \boldsymbol{z}_t$ and produces a context representation $\boldsymbol{c}_t = \boldsymbol{\mathcal{G}}(\boldsymbol{z}_1, ... \boldsymbol{z}_t)$.

An initial clustering of the context vectors using k-means algorithm generates the pseudo-labels for training the deep clustering model. A feed-forward layer with a softmax linearity acts as the classification network. The encoder $f : \mathcal{X} \mapsto \mathcal{Z}$ is parameterized by a 5-layer 1-D convolutional network with kernel sizes of 10,8,4,4 and 4 and with stride lengths of 5,4,2,2 and 2 respectively. This results in the down-sampling factor of 160 and the $\boldsymbol{z}_t$ representations are of dimension 256. Thus, the input audio signal sampled at 16000 Hz will have embeddings $\boldsymbol{z}_t$ sampled at 100Hz.

The autoregressive model is a 2 layer LSTM model with 256 hidden units. The number of clusters is kept as 50 in this work. For the contrastive prediction task, we use simple linear classifier W_k that takes in $\boldsymbol{c}_t$ as the input and tries to predict the future K feature representations, $\{\boldsymbol{z}_{t+k}\}_{1 \leq k \leq K}$. We use the value of $K = 12$.

3.3. Loss function

We use a joint loss which combines the contrastive loss along with cluster loss. The contrastive loss is defined in Equation 1. The cluster loss is introduced to increase the discriminability of phoneme-like sub-word units. This is the cross-entropy loss, denoted as $\mathcal{L}_{clus}$.

The total loss function is, $\mathcal{L}_{total} = \mathcal{L}_{cpc} + \alpha \times \mathcal{L}_{clus}$. Our experiments showed that a high value of ($\alpha = 12$) gives the best performance. This indicates that the clustering loss is more important in the model learning compared to the contrastive loss[1].

4. EXPERIMENTAL SET UP

4.1. Data

The training data consists of audio from LibriSpeech [30] and the Libri-light dataset [31]. The CPC-small model is trained on the 100 hours of clean audio subset from the LibriSpeech data. The CPC-big model, also having the same architecture of the CPC-small model, is trained on a 6k-hour subset of the Libri-light data. The deep clustering model training is performed on the 100 hours of the clean subset. The final representations from the model, $\boldsymbol{c}_t$, are re-clustered using k-means and are used to generate the pseudo-text sequences on LibriSpeech 960 hours data for training the language model. Each of the four metrics is evaluated on the development set designed for specific tasks of the ZeroSpeech 2021 challenge [9]. We use the Track-1, speech based language modeling task, for the experiments reported in this work. The language models are developed to generate likelihood scores to novel utterances, indicating the probability of token sequence. More details about the dataset are available in Dunbar et al. [9].

4.2. Model training

The pre-trained CPC-small/big model is used as the extractor for the initial embeddings for the k-means clustering. The k-means clustering is run for 100 epochs with 50 centroids. These pseudo labels extracted for 100 hour clean subset of Librispeech dataset along with the raw audio are used to train our models. Both the CPC-small and the CPC-big models are re-trained for 200 epochs with early stopping criteria using the joint loss. A patience factor of 5 is used in the training.

For the language model based evaluation tasks, we cluster the embeddings ($\boldsymbol{c}_t$) into discrete tokens and train the LSTM language model with architecture similar to the one described in [9]. This language model is trained using 960 hours of LibriSpeech data. The model is trained using fairseq tools[2].

The loss plots are shown in Figure 2. It is interesting to note that, while the objectives of predicting future representations (CPC loss) and that of being categorical (clustering loss), are fundamentally different, they align well on the training data. The plots also indicate that, for both the CPC-small and big models, the loss function behavior is similar.

The t-distributed stochastic neighborhood (tSNE) [32] based visualization of the representations from the baseline CPC model [9] and the proposed approach is shown in Figure 3. Here, we plot the scatter of the two dimensions from two different phonemes (using the ground truth information) w_1, w_2 spoken by two different speakers s_1 and s_2. As seen in the plot, the baseline CPC model is unable to cluster the different phonemes from the same speaker.

[1]Our implementation of experimental setups can be found at `https://github.com/iiscleap/CPC_DeepCluster`

[2]`https://github.com/pytorch/fairseq`

Authorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on June 13,2022 at 04:49:36 UTC from IEEE Xplore. Restrictions apply.

System	Pre-Training Data	Cluster Loss Training Data	ABX			
			Clean Within	Clean Across	Other Within	Other Across
CPC-small : Baseline [9]	LS-100h	-	10.26	14.17	14.24	21.26
CPC-big : Baseline [9]	LL-6kh	-	6.38	8.26	10.22	14.86
Wav2Vec [24]	LS-960h	-	9.47	11.69	12.35	17.61
Vq-Wav2Vec k-means [23]	LS-960h	-	12.68	14.83	15.16	20.11
Vq-Wav2Vec Gumbel [24]	LS-960h	-	10.66	12.02	13.17	17.55
DeepCluster-small : Proposed	LS-100h	LS-100h	6.57	9.51	8.69	14.96
DeepCluster-big : Proposed	LL-6kh	LS-100h	**5.83**	**8.21**	**7.71**	**13.60**

Table 1. The performance of various systems in terms of ABX error (%) metric on the ZeroSpeech 2021 challenge data.

System	Pre-Training Data	LM Training Data	GPU Budget	sWUGGY	sBLIMP	sSIMI	
						synth.	libri.
CPC-big + BERT-small : Baseline	LL-6kh	LS-960h	60h	65.81	52.91	3.88	5.56
CPC-big + LSTM-small : Baseline	LL-6kh	LS-960h	60h	66.13	53.02	4.42	7.56
DeepCluster-big + LSTM-small : Proposed	LL-6kh	LS-960h	60h	61.20	**59.42**	3.96	**10.25**
CPC-small + BERT-big : Baseline	LS-100h	LS-960h	1536h	70.69	54.26	2.99	6.68
CPC-big + BERT-big : Baseline	LL-6kh	LS-960h	1536h	**75.56**	56.14	**6.25**	8.72

Table 2. The performance in terms of various spoken language modeling metrics, sWUGGY, sBLIMP and sSIMI.

The embeddings from the proposed approach show more speaker in-variance for the same phoneme representations while being discriminative with the other phoneme representations.

4.3. Performance metrics

Phonetic - Given three-phoneme words, a, x and b, where a and b differ in one of the three phoneme-like units, while x and a are same, the ABX metric computes the fraction of cases when a and x are more distant than a and b. An ABX error in the range of 5-10% corresponds to good separation; 20%-30% indicates some signal, but not very good separation.
Lexical - The sWUGGY "spot-the-word" [33] is used to differentiate between a legitimate word from a non-word, which is similar in the lexical sense. The metric measures the fraction of events where the pseudo probability of the real word is higher than that of the non-word.
Syntactical - The sBLIMP metric is used as the syntactical metric. This measure, derived from [34], differentiates a grammatical sentence from an incorrect sentence. The metric measures the fraction of events where the pseudo probability of a grammatically correct sentence is greater than the incorrect one.
Semantic - The sSIMI similarity measures the similarity between the representations of pairs of words and compares the results with 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.

5. RESULTS

The phonetic metric based results (ABX error rate) for the proposed approach is shown in Table 1. In these experiments, we compare the performance of various approaches, like CPC small/big [9], wav2vec [22], vq-wav2vec [23], wav2vec 2.0 [24]. As seen in this table, among the various wav2vec models, the basic wav2vec [24] gives the lowest ABX similarity error. The CPC model with large training (CPC-big) gives the best ABX error among all the baselines compared in this work. The proposed approach of deep clustering applied on CPC-small embeddings improves significantly over the CPC-small model itself. Further, the application of the deep clustering on the pre-trained CPC-big model gives the best ABX error among all the systems compared. Note that, the deep clustering model is trained with only 100 hours of Librispeech data in this work.

In terms of the language model based metrics reported in Table 2, the top panel consists of methods that are low budget training using small LSTM/BERT based models for language modeling. The bottom panel reports methods using larger language models with higher GPU budget. In terms of the various metrics compared here, the proposed deep clustering approach (DeepCluster-big) approach, using a small budget LSTM based LM, yields the best performance in sBLIMP (syntactical) and sSIMI (semantic) metric on the librispeech dataset. The performance on other metrics like sWUGGY (lexical) is worse than the baseline CPC system. As reported in the Zerospeech challenge [9], the amount of data used in training does not necessarily improve all metrics considered here. From Table 1 and 2, we see that, even with a reduced GPU budget and reduced data for the deep cluster model training, the proposed approach yields representations that improve on multiple metrics over big models trained with larger GPU budgets.

6. SUMMARY

This paper presents our work on proposing a deep clustering based representation learning framework for automatic unit discovery. We develop a self-supervised learning approach that relies on clustering of representations to create pseudo phoneme-like units from raw speech. The model is trained using a combination of contrastive and cluster based loss functions. The experiments are performed on the ZeroSpeech 2021 challenge dataset using various metrics measuring phonetic, syntactic, semantic and lexical information captured by the acoustic units derived from the deep clustering framework. In these experiments, we find the proposed model, with reduced GPU budget and data for re-training, improves over larger models in terms of phonetic, syntactic and semantic metrics. Further, the visualization of the embeddings shows that the model is able to generate speaker invariant phoneme-like units.

Authorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on June 13,2022 at 04:49:36 UTC from IEEE Xplore. Restrictions apply.

7. REFERENCES

[1] Kushal Lakhotia et al., "Generative spoken language modeling from raw audio," *arXiv preprint arXiv:2102.01192*, 2021.

[2] "unsupervised pattern discovery in speech," .

[3] "an investigation on the use of acoustic sub-word units for automatic speech recognition," .

[4] Chin-Hui Lee, Frank K Soong, and Bing-Hwang Juang, "A segment model based approach to speech recognition," in *ICASSP*, 1988, pp. 501–502.

[5] Balakrishnan Varadarajan, Sanjeev Khudanpur, and Emmanuel Dupoux, "Unsupervised learning of acoustic sub-word units," in *Proceedings of ACL-08: HLT, Short Papers*, 2008, pp. 165–168.

[6] Maarten Versteegh, Roland Thiolliere, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The zero resource speech challenge 2015," in *Interspeech*, 2015.

[7] Ewan Dunbar, Xuan Nga Cao, et al., "The zero resource speech challenge 2017," in *ASRU*. IEEE, 2017, pp. 323–330.

[8] Ewan Dunbar, Julien Karadayi, et al., "The zero resource speech challenge 2020: Discovering discrete subword and word units," *arXiv preprint arXiv:2010.05967*, 2020.

[9] Ewan Dunbar, Mathieu Bernard, et al., "The zero resource speech challenge 2021: Spoken language modelling," 2021.

[10] Michael Heck, Sakriani Sakti, and Satoshi Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to zerospeech 2017," in *ASRU*. IEEE, 2017, pp. 740–746.

[11] T Ansari, Rajath Kumar, Sonali Singh, Sriram Ganapathy, and Susheela Devi, "Unsupervised HMM posteriograms for language independent acoustic modeling in zero resource conditions," in *ASRU*. IEEE, 2017, pp. 762–768.

[12] Benjamin van Niekerk, Leanne Nortje, and Herman Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," *arXiv preprint arXiv:2005.09409*, 2020.

[13] T Ansari, Rajath Kumar, Sonali Singh, and Sriram Ganapathy, "Deep learning methods for unsupervised acoustic modeling—leap submission to zerospeech challenge 2017," in *ASRU*. IEEE, 2017, pp. 754–761.

[14] "learning problem-agnostic speech representations from multiple self-supervised tasks," .

[15] 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.

[16] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio, "Multi-task self-supervised learning for robust speech recognition," in *ICASSP*. IEEE, 2020, pp. 6989–6993.

[17] Andy T Liu, Shang-Wen Li, and Hung-yi 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.

[18] Takashi Maekaku, Xuankai Chang, Yuya Fujita, Li-Wei Chen, Shinji Watanabe, and Alexander Rudnicky, "Speech representation learning combining conformer cpc with deep cluster for the zerospeech challenge 2021," *arXiv preprint arXiv:2107.05899*, 2021.

[19] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[20] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, "Transformer vq-vae for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge," *arXiv preprint arXiv:2005.11676*, 2020.

[21] 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.

[22] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," 2019.

[23] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-Supervised Learning of Discrete Speech Representations," 2020.

[24] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," 2020.

[25] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation Learning with Contrastive Predictive Coding," 2019.

[26] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural Discrete Representation Learning," 2018.

[27] 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.

[28] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, "Deep Clustering for Unsupervised Learning of Visual Features," 2019.

[29] Herve Jégou, Matthijs Douze, and Cordelia Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[30] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," *ICASSP*, pp. 5206–5210, 2015.

[31] Jacob Kahn, Morgane Rivière, et al., "Libri-light: A benchmark for ASR with limited or no supervision," *CoRR*, vol. abs/1912.07875, 2019.

[32] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[33] Gaël Le Godais, Tal Linzen, and Emmanuel Dupoux, "Comparing character-level neural language models using a lexical decision task," in *European Chapter of the ACL*, 2017, pp. 125–130.

[34] Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman, "Blimp: A benchmark of linguistic minimal pairs for english," *CoRR*, vol. abs/1912.00582, 2019.

Authorized licensed use limited to: J.R.D. Tata Memorial Library Indian Institute of Science Bengaluru. Downloaded on June 13,2022 at 04:49:36 UTC from IEEE Xplore. Restrictions apply.