LABERT: A Combination of Local Aggregation and Self-Supervised Speech Representation Learning for Detecting Informative Hidden Units in Low-Resource ASR Systems

Kavan Fatehi[1], *Ayse Kucukyilmaz*[1]

[1]School of Computer Science, University of Nottingham

kavan.fatehi@nottingham.ac.uk, ayse.kucukyilmaz@nottingham.ac.uk

Abstract

With advances in deep learning methodologies, Automatic Speech Recognition (ASR) systems have seen impressive results. However, ASR in Low-Resource Environments (LREs) are challenged by a lack of training data for the specific target domain. We propose that data sampling criteria for choosing more informative speech samples can be critical to addressing the problem of training data bottleneck. Our proposed Local Aggregation BERT (LABERT) method for self-supervised speech representation learning fuses an active learning model with an adapted local aggregation metric. Active learning is used to pick informative speech units, whereas the aggregation metric forces the model to move similar data together in the latent space while separating dissimilar instances to detect hidden units in LRE tasks. We evaluate LABERT with two LRE datasets: I-CUBE and UASpeech to explore the performance of our model in the LRE ASR problems.

Index Terms: Self-Supervised Learning, BERT, Local Aggregation Function, Low-Resource Environment ASR

1. Introduction

Recently, there have been remarkable improvements in end-to-end (E2E) automatic speech recognition (ASR) systems, which is paralleled by the availability of a large amount of labeled speech data. However, applications in low-resource environments (LREs) are challenged in terms of the quality and diversity of data resources, where representative training data and labels are insufficient and difficult to collect [1]. Examples of LREs include understudied languages, such as Kyrgyz [2], or speakers with different accents [1]. As data selection criteria is a major problem in LREs, an active learning approach can decrease the training data requirements for a low-resource ASR task by selecting more informative speech samples during the training process. In LABERT , we propose a self-supervised learning (SSL) approach by combining an active learning model as our data sampling criteria with the Local Aggregation (LA) function as our metric to identify similar speech units in the latent space to obtain better speech representation in LREs.

SSL models learn general data representations from unlabeled examples, which are then fine-tuned on labeled data [3]. wav2vec is an SSL model [4], which uses the Contrastive Predictive Coding (CPC) loss function for pre-training speech representations by predicting the near future frames in the acoustic sequence. The vq-wav2vec [5] model integrates wav2vec and BERT model [6] to obtain BERT-like speech representations through a two-stage training. DiscreteBERT [7] improves and extends the vq-wav2vec model by using the BERT pre-trained model and fine-tuning it on the downstream ASR task. wav-BERT [8] is an SSL framework that trains the model to discretize speech data and learn contextualized speech representations by solving the masked prediction task. BEST-RQ [9] masks the speech input and feeds the masks into an encoder to learn masked parts of speech based on the unmasked part through random-projection quantizers. DeLoRes [10] learns general purpose audio representations through an invariance and redundancy reduction based objective function. wav2vec 2.0 [3] enhances vq-wav2vec through a single-stage training by masking the input speech data into the latent space and then solves a contrastive task defined over a quantization of the latent representations by computing the similarity between the predicted masked vectors and original vectors. TERA [11] is a self-supervised pre-training method that utilizes alteration along time, frequency, and magnitude to pre-train Transformer Encoders on a large amount of unlabeled speech.

In the ASR literature, clustering approaches are also employed as a method to obtain pseudo-labels for SSL. Deep Cluster [12] uses k-means algorithm to group similar instances and optimizing an encoder network through a classification loss. Hidden unit BERT (HuBERT) [13] uses an offline clustering step to provide noisy labels for a BERT-like prediction loss. SwAV [14] uses an online clustering assignment step that produces the pseudo-labels in a mini-batch structure. However, many clustering algorithms suffer from the seed selection problem, resulting with noisy clustering results, which would negatively affect the learning process in the LRE ASR task. Our approach, Local Aggregation BERT (LABERT), draws inspiration from Local Aggregation (LA) [15], and applies a local non-parametric aggregation in a latent feature space instead of within the global clustering algorithm. LABERT selects more informative speech units and feeds them into the LA function, which enables it to address the noisy and arbitrary clustering process and to model the interrelation similarity more accurately in the latent spaces for the LRE ASR system.

LABERT is a novel self-supervised representation learning model for learning speech representations for the low-resource ASR task. Inspired by HuBERT, our model consists of an offline hidden unit detection module to provide the noisy labels for a BERT-like pre-training model. LABERT adapts, for the first time, non-parametric aggregation in a latent feature space for visual embedding [15] instead of using a global clustering algorithm to detect hidden units to learn speech representations. To address the training data bottleneck in LREs, we integrate a committee-based active learning model with an LA function to select more informative speech units. This is done by enabling the LA to obtain high performance in identifying close neighbours around the speech samples in the latent space. Our procedure, of improving and fine-tuning the committee-based active learning model during training, provides more informative speech units in the latent space that enables LA to classify

the speech units with similar statistical structures into the same clusters. This procedure allows LABERT to select an informative and diverse subset of the data to train a model, and obtain more accurate speech units to achieve performance comparable to the full dataset to address the data bottleneck and model a well-suited representation for downstream LRE ASR task.

2. Proposed Approach

LABERT is an end-to-end (E2E) ASR model which explores how to effectively use speech-only data to improve the performance of the speech recognition system in a low-resource environment. The components of LABERT are shown in Figure 1, which consists of: a) hidden unit discovery through Local Aggregation (LA) function and b) masked target unit prediction. The first module is to extract the hidden units from the raw audio speech. To accomplish this, the local aggregation (LA) function moves similar audio units together in the embedding space, while enabling dissimilar units to separate from each other. A committee-based active learning approach is used to select the more informative initial unit seeds, to address the noisy clustering problem in the local aggregation function [15]. The most critical aspect of clustering is determining the features into which the waveform should be transformed for clustering. Mel-Frequency Cepstral Coefficients (MFCCs) are used for the first clustering iteration, and for subsequent clustering, selected representations from the Canonical Correlation Analysis (CCA) [16] module are used. In the second module, a masked language model objective, similar to what is done in BERT [6], is used for masked prediction of hidden units. To achieve this, we calculate the cosine similarity between the context vectors and every hidden unit embedding from all available hidden units. Cross-entropy loss is then used for prediction.

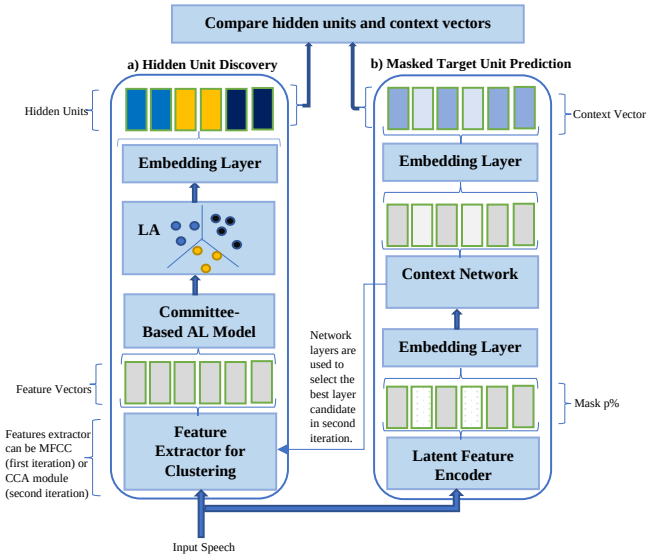

Figure 1: *The structure of LABERT model.*

2.1. Hidden Unit Discovery through LA function

In LABERT, LA function is adapted to extract hidden speech units from raw audio data. Our main objective is to train an embedding function $Y = f(X)$, which can effectively map the input speech $X = [X_1, ..., X_T]$ to the corresponding features, $Y = [Y_1, ..., Y_T]$, where similar units are grouped together and dissimilar ones are separated. To do this, we identify two sets of neighbours, close neighbours (C_i) and background neighbours (B_i), dynamically during the training of the embedding function for X_i and its embedding Y_i [15]. Close neighbours are embeddings that are similar to Y_i, while background neighbours establish the distance scale to determine the closeness to the embedding. Within the context of LREs, background neighbours enable the LABERT model to scale the measurement for the target downstream task to obtain a better performance in such environments. After detecting the close and background neighbours, LABERT forces the current embedding toward close neighbours but far from background ones.

In each step of optimization, for a given embedding space Y_i, the background neighbours are determined as the k closest embedding spaces $g_k(Y_i)$ within Y, where the cosine distance is used as the measure of similarity. To find the close neighbours, k-means clustering algorithm is applied to cluster all embedding spaces in Y to cluster the representations into P groups $\{1, ..., P\}$. At this step of LABERT, to address the noisy clustering results, we employ a committee-based active learning approach to obtain more informative embedding as the initial clustering seeds for the k-means algorithm. This helps LABERT to achieve more accurate classification of hidden units. The number k of background neighbours and number P of clusters are hyperparameters of the model.

Taking into account the definition of close and background neighbours, a LA level is defined as $L(C_i, B_i|\Theta, X_i)$ for each speech unit X_i. Θ parameters are tuned during the training to maximize the level of local aggregation. In [15], the probability that a feature Y to be considered as the i-th unit is defined as:

$$P(i|Y) = \frac{exp({Y_i}^T Y/\tau)}{\sum_{j=1}^{T} exp({Y_j}^T Y/\tau)} \tag{1}$$

where $\tau \in [0, 1]$ is a fixed hyperparameter.

The probability that a feature Y is classified as a unit in a speech frame T is computed as:

$$P(T|Y) = \sum_{i \in T} P(i|Y) \tag{2}$$

The level of local aggregation is defined as the negative log-likelihood of Y_i being a close neighbour (is in C_i), given that Y_i is recognized as a background neighbour (is in B_i):

$$L(C_i, B_i|\Theta, X_i) = -log \frac{P(C_i \bigcap B_i|Y_i)}{P(B_i|Y_i)} \tag{3}$$

Finally, the loss to be minimized is:

$$Loss = L(C_i, B_i|\Theta, X_i) + \lambda \|\Theta\|_2^2 \tag{4}$$

where λ is a regularization hyperparameter.

As explained earlier, LABERT employs a committee-based active learning (AL) approach to obtain more informative parts of speech data to consider them as seeds for the local aggregation function. Most of the classical committee-based AL approaches consist of two or more different structure models to present the difference among the models [17]. In [17], a single committee-based AL model, SMCA, is proposed, where the committee model and its variants are constructed by applying the dropout technique. The main drawback of the SMCA is the inconsistency between training and inference, which leads to reduced performance of the model in both high- and low-resource ASR scenarios. To address this issue, in LABERT, we use the R-Drop model [18], in which each speech input X_i is fed into the model twice at each step of training, producing two samples of the model, $\rho_1^{\omega}(l_i|x_i)$ and $\rho_2^{\omega}(l_i|x_i)$, where l_i is the

transcribed text of x_i. R-Drop regularizes the minimization of bidirectional Kullback-Leibler (KL) divergence between these distributions:

$$L_{KL}^{i} = \frac{1}{2}\Big(D_{KL}\Big(\rho_1^{\omega}(l_i|x_i)||\rho_2^{\omega}(l_i|x_i)) + D_{KL}(\rho_2^{\omega}(l_i|x_i)||\rho_1^{\omega}(l_i|x_i)\Big)\Big) \tag{5}$$

By applying the R-Drop method during the training step, the dropout hypotheses of the seed model could be different from the normal hypotheses in the model. The difference between these hypotheses at frame-level is considered as a data selection metric to obtain more informative speech units from utterance to start the clustering algorithm in local aggregation function. LABERT incorporates both informativeness and diversity to select a more informative subset of speech data to address the noisy clustering problem. LABERT employs the B_i set to dynamically compute the diversity during the pre-training process. Therefore, AL approach prevents LA function to select most or all of the data to be similar to each other, which enables LABERT to produce more accurate clusters of speech units.

2.2. Masked Target Unit Prediction

In this section, we summarize how to use the BERT model [6] and also how to select a good representation from BERT to use in the second iteration of the LA function. BERT is a language model which uses the masked prediction model over a large amount of text data. Therefore, a pre-trained BERT model can satisfy the lack of text data for an ASR system in the low-resource target task. LABERT adopts the same strategy as HuBERT [13] and wav2vec 2.0 [3] for mask generation, but only $p\%$ of the selected timesteps are masked to allow the model to receive real input to address the train-test inconsistency. In LABERT, we develop on our previous work [19] and use a layer analysis module to select the layer best-suited for the LRE target ASR task for our second iteration in the LA function. We use CCA [16] as a measure to detect the layer of the model, which is well-suited for the target LRE ASR task. CCA is a statistical approach to represent the maximum correlations between linear combinations of two continuous value vectors. Therefore, CCA can be used to calculate the similarity between the representations of the layers and the acoustic feature vector to evaluate how different layers of the model are adapted to the downstream target task. Through this strategy, we force the model to learn representations from the downstream ASR task to improve the performance of the model in the target domain.

3. Experiments

3.1. Datasets

For unsupervised pre-training, we use the full 960 hours of LibriSpeech (Libri) [20], full 81 hours of Wall Street Journal (WSJ) [21], 1k hours of Common Voice (CV) [22] and 450 hours of TED-LIUM 3 (TED3) as our high-resource environment (HRE) datasets. The performance of LABERT is evaluated in low-resource environments, using the following LRE datasets: 1) The UASpeech dataset [23], which is the largest corpus of dysarthric speech in American English. It is a collection of 541 speech recordings from 19 individuals with cerebral palsy. 2) The Industrial Co-bots Understanding Behavior (I-CUBE) dataset is a Human-Robot Collaboration dataset, generated through a user study in our lab, where participants were asked to interact with an actor posing as a robot (following the Wizard of Oz protocol) using natural language [24]. The dataset involves speakers who instruct and ultimately teach a robot how to sort different garments into four baskets as if they were sorting their own. During the experiments, the robot would also respond to the actions of the participant with its own actions or speech. Video recordings of each session were collected, resulting in a total of 42 videos, which represent 300 minutes of transcribed audio.

3.2. Experiment Setup and Metrics

For pre-training, we train LABERT with the selected HRE datasets in two settings: Base setting with 12 layers and Large setting with 24 layers of the encoder. For the model configuration, the mask span is set to $l = 10$ and $p = 8\%$ of the waveform encoder output frames are randomly selected as the initial mask. We set $k = 4096$ to compute B_i using the nearest neighbours procedure. In computing C_i, we run the k-means clustering algorithm with 50 and 100 clusters on 39-dimensional MFCC features, to obtain labels for LABERT pre-training over the HRE data sets. We considered 50 and 100 clusters on 100h, 300h and 500h of speech samples from LibriSpeech and fine-tuned the model with I-CUBE for cluster quality analysis. The input acoustic features are 80-dimensional filterbanks, extracted with a hop size of 10 ms and a window size of 25 ms, which are normalized with the mean and variance. For the WSJ setup, the number of output classes is 52, including the 26 letters of the alphabet, space, noise, symbols such as period and an unknown marker. To predict the probability distribution of all characters in the alphabet, we use the CTC loss function and use AdamW optimizer [25] to update the model with an initial learning rate of 0.001. The text is tokenized using SentencePiece [26] and we set the vocabulary size to 500.

Benchmarking results are presented for the pre-trained and fine-tuned wav2vec 2.0 and HuBERT models in Base and Large settings, as well as for QuartzNet and DiscreteBert. The primary evaluation metric we used is the word error rate (WER). We also compute the Phone Purity and Phone-Normalized Mutual Information (PNMI) to evaluate the quality of the obtained cluster assignments from LA function in different layers:

We obtain phonetic transcripts that are aligned at the frame level to quantify the correlation between the LA assignments and the underlying phonetic units. Let $[y_1, ..., y_t]$ and $[f_1, ..., f_t]$ be frame-level and LA function labels, respectively. The joint distribution of y and f is the normalized number of occurrences of the labels:

$$p_{y,f}(i,j) = \frac{\sum_{t=1}^{T}[y_t = i \wedge f_t = j]}{T} \tag{6}$$

where i and j demonstrate the i^{th} phoneme class and j^{th} LA function class label [13]. Phone Purity measures the frame-level phone accuracy if we transcribe each LA function class with the most likely phone label. It is defined as $\mathbb{E}_{p_f(j)}\Big[p_{y|f}(y^*(j)|j)\Big]$, where $p_{y|f}(y^*(j)|j)$ is the conditional probability of phone given a class label j and $y^*(j)$ is the most likely phone label for the j-th class. Higher purity indicates greater quality.

PNMI is an information-theoretic metric used to measure the similarity between two clusterings of data. It measures the percentage of uncertainty about the phone label y that is reduced after observing the class label f and is defined as follows, where H(.) is the entropy:

$$\frac{I(y,f)}{H(y)} = 1 - \frac{H(y|f)}{H(y)} \tag{7}$$

A higher value of PNMI in our analysis indicates that the quality of LA clustering is better.

3.3. Results

Table 1 presents the ASR performance of LABERT in terms of the word error rate (WER) when tested on I-CUBE and UASpeech LRE datasets, after being pre-trained and fine-tuned on I-CUBE and UASpeech, respectively. Comparisons are reported for wav2vec 2.0 and HuBERT in Base and Large settings, as well as DiscreteBERT [7] and QuartzNet[27]. We show that the performance of LABERT is improved by increasing the amount of unlabeled data during pre-training (see Section 3.1) which indicates the scalability of the proposed model. In the Base setup, after fine-tuning on I-CUBE, LABERT achieved WERs of 13.39%, 14.78%, 14.93% and 17.35% when pre-trained on LibriSpeech, TED, WSJ and CV corpora, respectively, which outperformed the other algorithms in the Base setting. LABERT achieved even better results in the Large setting when tested on I-CUBE dataset, with WER of 9.53%, 10.24%, 12.21% and 16.63% after pre-training over LibriSpeech, TED, WSJ and CV, respectively. LABERT significantly outperformed QuartzNet and DiscreteBERT as well. Similarly, after fine-tuning over UASpeech on Base and Large settings, LABERT achieved best results across all benchmark algorithms.

Table 1: *Word error rate (WER) results obtained with different methods pretrained in HRE datasets (Libri, TED, WSJ and CV) and fine-tuned in two LREs (I-CUBE and UASpeech). The best performing models in corresponding settings are highlighted.*

LRE	Method	Libri	TED	WSJ	CV
I-CUBE	**LABERT – Base**	**13.39**	**14.78**	**14.93**	**17.35**
	LABERT – Large	**9.53**	**10.24**	**12.21**	**16.63**
	wav2vec 2.0 – Base	17.38	15.45	16.61	18.42
	wav2vec 2.0 – Large	11.61	13.64	14.73	17.22
	HuBERT – Base	16.81	16.43	15.98	18.13
	HuBERT – Large	11.28	12.71	14.39	16.81
	QuartzNet	26.51	29.75	28.39	31.53
	DiscreteBERT	27.93	31.35	29.48	33.38
UASpecch	**LABERT – Base**	**17.28**	**18.65**	**21.13**	**23.91**
	LABERT – Large	**11.27**	**12.28**	**15.11**	**17.93**
	wav2vec 2.0 – Base	19.07	21.31	23.94	25.18
	wav2vec 2.0 – Large	14.28	15.91	16.23	18.87
	HuBERT – Base	19.31	21.18	24.49	25.93
	HuBERT – Large	14.93	15.58	16.39	18.98
	QuartzNet	29.15	34.93	31.79	36.75
	DiscreteBERT	31.48	36.75	32.19	37.21

The PNMI results are shown in Table 2. These results demonstrate that the PNMI increases with the amount of pre-training speech data, which enhance the quality of the cluster results. A possible explanation for this might be that by increasing the pre-training data, the committee-based active learning approach can select more informative speech units for seed initialization of the LA function, therefore LABERT can improve the quality of the clusters in LRE tasks.

Finally, we evaluate the quality of the LA function for detecting hidden units in each layer of LABERT. In this analysis, we considered the first two iterations of the LABERT after pre-training the model on LibriSpeech dataset and fine-tuning it with I-CUBE and UASpeech. The results are compared with HuBERT since it achieved the next best WER results in the earlier analysis. Phone Purity and PNMI are shown in Figures 2 and 3, for each layer of the model. We observe that the phone purity gradually increased in the first layers after pre-training and fine-tuning with both I-CUBE and UASpeech. The interesting finding is that in the last layers of both models, phone purity decreased. The same trend is observed in the PNMI after fine-tuning the models with both I-CUBE and UASpeech. The middle layers (7-9) of the LABERT, which were selected by the CCA module to feed into the AL model for selecting initial seeds for hidden units detection process, exhibited the highest PNMI. In both LRE settings, fine-tuning on I-CUBE and UASpeech, significant phone purity and PNMI results were observed at these middle layers, suggesting that they are well-suited for the downstream LRE ASR task. Notably, the phone purity and PNMI analysis revealed that the LABERT model demonstrates more stable clusters, indicating that it performs reasonably well for low-resource ASR tasks.

Table 2: *PNMI values for different cluster numbers and pre-training data size. Fine-tuning is done using I-CUBE.*

Feature	Number of Clusters	PNMI		
		100h	**300h**	**500h**
MFCC	50	0.384	0.387	0.388
	100	0.432	0.435	0.435
Selected Layer From CCA	50	0.631	0.633	0.633
	100	0.785	0.787	0.787

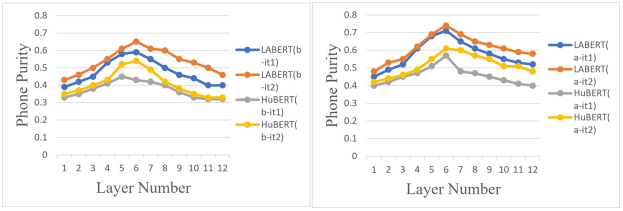

Figure 2: *Phone purity of LABERT and HuBERT in Base configuration after pre-training on LibriSpeech and fine-tuning over I-CUBE and UASpeech for the first and second iteration. a and b stand for I-CUNE and UASpeech, respectively.*

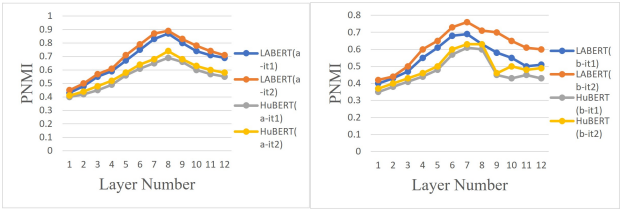

Figure 3: *PNMI of LABERT and HuBERT in Base configuration after pre-training on LibriSpeech and fine-tuning over I-CUBE and UASpeech for the first and second iteration. a and b stand for I-CUNE and UASpeech, respectively.*

4. Conclusions

We proposed LABERT, a self-supervised speech representation learning model for ASR in LREs. LABERT integrates a committee-based active learning model with a local aggregation (LA) function as a hidden unit detection module. Active learning is used to select informative speech units, whereas the LA function learns feature embeddings, which classify similar speech units together and move apart dissimilar ones. We pre-train LABERT on four well-known high-resource datasets and fine-tune it with two LRE datasets. Our model achieves up to 16.63% WER reduction on the LR data, which outperforms state-of-the-art ASR models. In conclusion, LABERT creates representations, which are useful to a variety of speech recognition tasks in low-resource settings.

5. References

[1] J. Meyer, "Multi-task and transfer learning in low-resource speech recognition," Ph.D. dissertation, The University of Arizona, 2019.

[2] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 826–27 839, 2021.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

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

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

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: https://doi.org/10.18653/v1/n19-1423

[7] A. Baevski, M. Auli, and A. Mohamed, "Effectiveness of self-supervised pre-training for speech recognition," *arXiv preprint arXiv:1911.03912*, 2019.

[8] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.

[9] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.

[10] S. Ghosh, A. Seth, and S. Umesh, "Delores: Decorrelating latent spaces for low-resource audio representation learning," *arXiv preprint arXiv:2203.13628*, 2022.

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

[12] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 132–149.

[13] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[14] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9912–9924, 2020.

[15] C. Zhuang, A. L. Zhai, and D. Yamins, "Local aggregation for unsupervised learning of visual embeddings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6002–6012.

[16] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in statistics*. Springer, 1992, pp. 162–190.

[17] X. Sun, B. Wang, S. Liu, T. Lu, X. Shan, and Q. Yang, "Lmc-smca: A new active learning method in asr," *IEEE Access*, vol. 9, pp. 37 011–37 021, 2021.

[18] L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liu *et al.*, "R-drop: Regularized dropout for neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 890–10 905, 2021.

[19] K. Fatehi, M. Torres Torres, and A. Kucukyilmaz, "ScoutWav: Two-Step Fine-Tuning on Self-Supervised Automatic Speech Recognition for Low-Resource Environments," in *Proc. Interspeech 2022*, 2022, pp. 3523–3527.

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

[21] D. B. Paul and J. Baker, "The design for the wall street journal-based csr corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*, 1992.

[22] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 4218–4222. [Online]. Available: https://aclanthology.org/2020.lrec-1.520

[23] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. S. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[24] N. Dahlbäck, A. Jönsson, and L. Ahrenberg, "Wizard of oz studies—why and how," *Knowledge-based systems*, vol. 6, no. 4, pp. 258–266, 1993.

[25] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[26] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 66–75. [Online]. Available: https://aclanthology.org/P18-1007

[27] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6124–6128.