# SUPERB: Speech Understanding and PERformance Benchmark

*Shu-wen Yang*[1] *Po-Han Chi*∗[1] *Yung-Sung Chuang*∗[1] *Cheng-I Lai*∗[2] *Kushal Lakhotia*∗[3] *Yist Y. Lin*∗[1] *Andy T. Liu*∗[1] *Jiatong Shi*∗[4] *Xuankai Chang* *Daniel Lin*[1] *Tzu-Hsien Huang*[1] *Wei-Cheng Tseng*[1] *Godic Lee*[1] *Darong Liu*[1] *Zili Huang*[4] *Annie Dong*†[5] *Shang-Wen Li*†[5] *Shinji Watanabe*[6] *Abdelrahman Mohamed*[3] *Hung-yi Lee*[1]

[1]National Taiwan University [2]Massachusetts Institute of Technology [3]Facebook AI Research
[4]Johns Hopkins University [5]Amazon AI [6]Carnegie Mellon University

leo19941227@gmail.com, kushall@fb.com, jefflai108@gmail.com, jshi34@jhu.edu, shangwel@amazon.com, shinjiw@ieee.org, hungyilee@ntu.edu.tw

## Abstract

Using self-supervised learning methods to pre-train a network on large volumes of unlabeled data followed by fine-tuning for multiple downstream tasks has proven vital for advancing research in natural language representation learning. However, the speech processing community lacks a similar setup that systematically measures the quality of learned representations across a wide range of downstream speech applications. To bridge this gap, we introduce the Speech Understanding and Performance Benchmark (SUPERB). SUPERB is a leaderboard to benchmark the performance of learned speech representations on ten speech processing tasks. We present a complete framework for learning and evaluating specialized prediction heads for each task given the pre-trained speech representations. Our results on many publicly-available self-supervised models demonstrate their generalization abilities to multiple speech tasks with limited supervised and minimal architecture changes. All the materials are open-sourced and reproducible in the s3prl toolkit to facilitate future research in speech representation learning.

**Index Terms**: Self-Supervised Learning, Representation Learning, Model Generalization, Benchmark, Leaderboard, Evaluation

## 1. Introduction

Starting from ELMo [1] and BERT [2] in natural language processing (NLP), the effectiveness of Self-supervised learning (SSL) is evident in various domains [3, 4]. It is becoming a new paradigm to solve problems by pretraining a shared network with self-supervision using a large amount of unlabeled data to encode general-purpose knowledge, and adapting the network to various downstream tasks with limited labeled examples. This paradigm achieves state-of-the-art (SOTA) performance in many applications.

In addition to the performance, the paradigm is desirable for its re-usability across tasks to democratize deep learning to more application scenarios. Developing deep neural networks is expensive nowadays in terms of data collection, modeling, computing power, and training time, and repeating the same process for each specific use case is prohibitively costly for both academic and industrial researchers. SSL can significantly speed up and lower the entry barrier for model development in Speech, as using a shared pretrained model and limited finetuning on downstream tasks has been shown to consistently outperform purely supervised methods in NLP [2] and CV [5]. A well-established benchmark is essential to evaluate the re-usability and generalizability of pretrained models across a wide range of downstream tasks. Thus, GLUE [6] was proposed in NLP and VISSL [7] is leveraged in CV. These benchmarks contain a group of tasks to generally evaluate the model's capability on text or image processing and fuel the latest SSL research progress.

SSL has been explored in speech, including pretraining with generative loss [8, 9, 10] or discriminative loss [11, 12], and pretraining for short segments [8] or the entire utterances [9, 10, 11, 12, 13]. The ways researchers pretrain and use the learned models are diverse. [8] extracted fixed-dimension segmental representation for query-by-example; [9, 10, 13] demonstrated a sequence of extracted representations for entire utterances can capture phonetic, speaker, or emotion characteristics; [12, 14, 15, 16] showed that SSL can establish new principles to solve ASR problem. Researchers have investigated these pretrained models' capabilities on tasks including speaker verification [17], language identification [17], emotion recognition [13], speech translation [9] and spoken language understanding [18]. While these works showed promising results on various speech processing tasks, unlike CV or NLP areas, the works were investigated with different datasets and experimental setups. Absence of a shared benchmark makes it hard to compare, and to draw insights across the techniques. Furthermore, tasks evaluated in these works mostly require complex and specialized downstream settings with cumbersome training of the entire networks [19, 12, 15]. Both factors limit the impact of SSL on speech processing in research and industry.

We introduce Speech Understanding and Performance Benchmark (SUPERB) to address the problem. SUPERB collects ten benchmark tasks in speech processing that have common real-world applications. We further propose a simple framework to solve the ten tasks with a general-purpose pretrained network and lightweight prediction heads specialized for each task. The pretrained network has parameters shared across tasks and is used as a representation extractor; each head is finetuned with limited labeled data for the corresponding task. With SUPERB, one can readily compare the effectiveness of different model structures, and pretraining corpora across tasks. There are existing benchmark corpora [20, 21] proposed to evaluate representation learned with SSL. As compared to the existing efforts that focus on analyzing fine-grained characteristics of the learned representation, SUPERB evaluates in a task-oriented fashion and examines the generalizability of networks across tasks. Our experiment results show that the

*Equal contribution; sorted alphabetically

†Work done independently outside Amazon employment

proposed framework yields competitive performance in most tasks, and the SSL learned representations outperforms classic features used in speech domains by a large margin. Despite the promising results, our framework does not achieve SOTA in all of the SUPERB tasks, suggesting the space for future research. Thus, we present SUPERB as a challenge with a leaderboard[1]. We welcome researchers to participate in the challenge via our open-sourced evaluation toolkit[2], which supports benchmarking most of the existing SSL methods and any customized model, to drive the SSL research frontier.

## 2. Self-Supervised Pretrained Models

In this section, we describe the SSL pretrained methods investigated in this paper. We summarize the methods in Table 1, and categorize them into two classic modeling approaches: generative modeling and discriminative modeling.

### 2.1. Generative Modeling

Generative modeling has long been a prevailing approach for learning speech representation and has been shown effective in many domains [8, 9, 10]. Instances of generative modeling investigated here include APC [9], VQ-APC [22], and TERA [19]. APC adopts the language model-like training scheme on a sequence of acoustic features with unidirectional RNN. APC optimizes the model to generate future frames conditioning on past frames. VQ-APC further applies vector-quantization (VQ) layers to the representation of APC. This method imposes a bottleneck that forces the model to learn compact and low bit-rate representations. TERA adopts the BERT-like [2] masked pretraining on Transformer encoders by masking timestamps and the frequency bins of input acoustic features and generating the masked parts.

### 2.2. Discriminative Modeling

Contrastive learning as a branch of discriminative modeling for SSL receives great attention recently, and the methods studied here include CPC [11], wav2vec [12], vq-wav2vec [14], and wav2vec 2.0 [15]. CPC learns to discriminate the correlated positive samples from negative samples with InfoNCE loss. The loss is inspired by classic NCE loss and designed to maximize the mutual information between the raw data and the representation. wav2vec follows the same loss while using deeper networks for both the feature encoder and the context network in CPC. vq-wav2vec learns BERT-like speech representations through a two-stage training pipeline, where a VQ module is inserted between the feature encoder and the context network in wav2vec. In the first stage, the same InfoNCE loss is employed, and the speech signal is discretized to a sequence of tokens. Tokens then are used as pseudo-text to train the standard BERT model for contextualized representations in the second stage. wav2vec 2.0 improves upon vq-wav2vec, where the two-stage training is merged into one end-to-end training scheme, by applying time masking in the latent space and replacing the token classification softmax with InfoNCE discrimination. HuBERT [23] is trained with a masked prediction task similar to BERT [2] but with masked continuous audio signals as inputs. The targets are obtained through unsupervised clustering of raw speech features or learned features from earlier iterations, motivated by DeepCluster.

[1] webpage

[2] `https://github.com/s3prl/s3prl`

## 3. Speech Understanding and Performance Benchmark

We propose a set of ***10*** downstream tasks to jointly benchmark the capability of SSL approaches and learned representations. The tasks are collected by speech communities for various aspects to comprehensively examine the approaches. We also build a framework to leverage SSL approaches and solve the tasks. As a heavyweight downstream adaptation is cumbersome in model development, we freeze the SSL pretrained model to extract fixed representation for the downstream usage, and limit the parameters and network architectures utilized for task fine-tuning. Having the downstream adaptation lightweight benefits the efficiency in data usage, computation, and development. We categorize the ten tasks into two tracks, linear separability and advanced applications, and discuss the details of tasks and the application of proposed frameworks to each problem.

### 3.1. Linear Separability

Following the conventional evaluation protocol [11, 9, 19], we probe representations with linear models in ***5*** tasks, Phoneme Recognition, Keyword Spotting, Intent Classification, Speaker Identification, and Emotion Recognition. These tasks serve as a direct indication of representations' capability without any dependency on specific downstream models. The default setting for these tasks is a mean-pooling of representations from pretrained networks followed by a linear layer as the downstream model. The model is optimized by Adam with batch size 32 and cross-entropy loss. We use the standard training/validation/testing splits in each dataset and accuracy (ACC) as the evaluation metric.

**Phoneme Recognition, PR** classifies each frame on the smallest content units. In [9, 10, 19], phoneme classification is conducted with force-aligned frame-wise phoneme labels. We avoid the potential inaccurate alignment by offloading the alignment to the downstream model and CTC loss. The downstream model is the same frame-wise single linear layer as in [9, 10, 19]. We use LibriSpeech [24] train-clean-100, dev-clean, and test-clean subsets for training/validation/testing. Phoneme transcriptions are obtained from the LibriSpeech official grapheme-to-phoneme model *g2p-model-5* and the conversion script from Kaldi *librispeech s5* recipe. The reported metric is phone error rate (PER).

**Keyword Spotting, KS** is to detect specific keywords from utterances with minimal effort without transcribing the utterances. KS formulates the problem by classifying utterances into a predefined set of words. The task is usually performed on-device for the fast response time. Thus, accuracy, model size, and inference time are crucial. We utilize the widely used Speech Commands dataset v1.0 [25] for evaluation. The dataset consists of ten classes of keywords, a class for silence, and an *unknown* class to include the false positive.

**Intent Classification, IC** aims to infer high-level semantics from speech and is a crucial component in common Spoken Language Understanding (SLU) systems We design our IC task on top of previous end-to-end literatures [26], where intent labels are predicted directly from speech. We use the Fluent Speech Commands [26] dataset for our experiments, where each utterance is tagged with three labels: action, object, and location. In our downstream model, three separate linear layers projecting a single mean pooled latent vector are used for the respective three labels.

**Speaker Identification, SID** classifies each utterance for

its speaker identity. SID is a conventional evaluation task for SSL representation [11, 9, 10, 19, 13]. Instead of using LibriSpeech or Wall Street Journal, we opt for a more challenging dataset from the community, VoxCeleb1 [27]. The dataset comprises utterances from 1251 speakers collected in the wild.

**Emotion Recognition, ER** predicts emotion in each utterance. We adopt ER as one of the evaluation tasks because it is interesting to study the paralinguistics learned by SSL beyond the content and speaker properties. We utilize the most widely used ER dataset IEMOCAP [28], and follow the conventional evaluation protocol: we drop the unbalance emotion classes to leave the final four classes with a similar amount of data points and cross-validates on five folds of the standard splits.

### 3.2. Advanced Applications

We further examine the capability of SSL approaches and learned representations in ***5*** real-world problems, Automatic Speech Recognition, Query by Example Spoken Term Detection, Slot Filling, Automatic Speaker Verification, and Speaker Diarization. Although complex and task-specific model architectures are common to push the SOTA performance in these problems, when designing our models, we choose to follow the principle to keep downstream models as lightweight and general as possible while achieving competitive performance. The principle is essential to focus our comparison on the representations learned by different SSL approaches, and to examine the generalizability, accessibility, and re-usability of each approach and the system built upon it. We describe the five problems and our models in detail below. We also summarize our experiment datasets. The standard training/validation/testing splits in each dataset are used if not explicitly mentioned.

**Automatic Speech Recognition, ASR** is the most studied application in SSL, where [12, 15, 16, 19] all show promising results of utilizing pretrained models in different ways. Usually, complicated designs are required to achieve SOTA performance for a classic ASR system. However, we aim at exploring how easy ASR can be when leveraging powerful speech representations. We adopt a simple 2-layer 1024-unit bidirectional LSTM for the downstream model, which is also used in [16]. We train the downstream model by CTC loss on letters and decode with the official LibriSpeech 4-gram language model powered by KenLM and flashlight toolkit for faster inference. The LibriSpeech train-clean-100/dev-clean/test-clean subsets are used for training/validation/testing. Preliminary results show that the LSTM model is prone to overfit; hence we also apply SpecAugment. The reported metric is word error rate (WER).

**Query by Example Spoken Term Detection, QbE** is to search for a spoken term (query) in an audio database (documents), without speech-to-text conversion. QbE has no dependency on ASR systems and relies more on the feature extraction and the detection algorithm. We mostly follow the system proposed by GTTS-EHU for QUESST at MediaEval 2014 [29], but replace the conventional supervised phoneme posteriorgram with SSL representations. Representations are extracted for every utterance and normalized along each feature dimension. We apply Dynamic Time Warping (DTW) to the representations with the package: *dtw-python* and obtain a score for each query-document pair. The scores belonging to each query are normalized separately. Experiments are conducted on the non-native English subset of QUESST 2014 [30] because all SSL representations were pretrained on English corpora. Combinations of distance functions and step functions are treated as hyperparameters and tuned with validation set. The evaluation metrics is maximum term weighted value (MTWV).

**End-to-End Slot Filling, SF** is another essential task in SLU, where a sequence of semantic slot-types are predicted from raw audio with or without an intermediate natural language understanding (NLU) module [18]. Note that in contrast to IC, a SF model should predict a slot-type sequence basing on the *predicted text* sequence [18]. To gauge the effectiveness of SSL in inferring semantics directly from raw audio, we adopt one of the baseline models in [18], where slot-type labels are represented as special tokens in transcriptions to re-formulate SF as an ASR problem. The training scheme is the same as in our ASR task, except for the pre-processing and post-processing of the transcriptions to include the slot-type labels. The metrics include slot-type F1 score and slot-value CER [31]. The slot-type F1 score is computed to evaluate the predicted slot-types' correctness without considering the slot-values. For each ground-truth slot, a predicted slot with the same slot-type is chosen, and CER between their slot-values is further computed to ensure whether the model is predicting the correct slot-type grounded on the correct content.

**Automatic Speaker Verification, ASV** verifies whether the speakers of a pair of enrollment and testing utterances match. The speakers in the testing set may not appear in the training. As compared to SID, where the speakers of training and testing sets are identical, ASV is an open-set and more challenging problem. We adopt the well-known x-vector [32] as the downstream model by only replacing the statistical pooling with attentive pooling, and we train on VoxCeleb1 without noise augmentation to stick to our lightweight-downstream principle. The metric is equal error rate (EER) with cosine backend.

**Speaker Diarization, SD** targets to address the *who spoken when* problem. Different from SID and ASV, SD is conducted under frame level. Representations have to be rich in speaker characteristics for each frame and compatible with mixtures of signals, which is not presented as pretraining data for existing SSL approaches. For our lightweight-downstream principle, we employ the end-to-end speaker diarization with permutation-invariant training (PIT) loss [33] instead of the clustering-based methods, and use only a single-layer 512-unit LSTM for the downstream model. We adopt LibriMix [34] for diarization. The time-coded speaker labels were generated using alignments from Kaldi Librispeech ASR model. Diarization error rate (DER) is used as the evaluation metric.

## 4. Policy and Experiment

To keep a fair and easy evaluation policy for all the SSL representations, we limit the space for downstream hyper-parameter tuning. In this paper, we search the best learning across 1.0E-1 to 1.0E-7 in log-scale for each representation/downstream pair. More downstream hyper-parameters will be available to search in the final release of the challenge, but they can not be many in principle.

The results are presented in table 2. First, for the linear separability tasks in the left half part of the table, it is almost impossible for FBANK to work on any task and SSL representations all perform well to some degree. Their capability differs a lot though. wav2vec 2.0 is the best in phonetics, while HuBERT performs best across KS, IC, and ER. On the other hand, TERA ranks the first place on SID.

As for advanced applications, HuBERT outperforms all the existing representations on ASR greatly, which makes training an ASR system much easier than before. FBANK becomes a competitive feature that its WER is not much higher than many

| Method | Network | #Params | Stride | Input Feature | Pre-train | Learning Style |
|---|---|---|---|---|---|---|
| FBANK | - | 0 | 10ms | 80-dim + delta 2 | - | - |
| APC [9] | 3-GRU | 4.06M | 10ms | 80-dim log Mel | LS 360 hr | autoregressive generation |
| vq-APC [22] | 3-GRU | 4.06M | 10ms | 80-dim log Mel | LS 360 hr | autoregressive generation + VQ |
| TERA [19] | 3-Trans | 21.33M | 10ms | 80-dim log Mel | LS 960 hr | masked reconstruction |
| CPC [11, 35] | 5-Conv 1-Trans | 1.84M | 10ms | waveform | LL 60k hr | contrastive |
| wav2vec [12] | 19-Conv | 29.39M | 10ms | waveform | LS 960 hr | contrastive |
| vq-wav2vec [14] | 20-Conv | 6.04M | 10ms | waveform | LS 960 hr | contrastive + VQ |
| wav2vec 2.0 Base [15] | 7-Conv 12-Trans | 95.04M | 20ms | waveform | LS 960 hr | contrastive + latent masking |
| wav2vec 2.0 Large [15] | 7-Conv 24-Trans | 317.38M | 20ms | waveform | LS 960 hr | contrastive + latent masking |
| HuBERT Base [23] | 7-Conv 12-Trans | 94.68M | 20ms | waveform | LS 960 hr | masked pseudo-label prediction |
| HuBERT Large [23] | 7-Conv 24-Trans | 316.61M | 20ms | waveform | LL 60k hr | masked pseudo-label prediction |

Table 1: ***Details of baseline features and recent SSL methods.*** *LibriSpeech and LibriLight are denoted as* ***LS*** *and* ***LL****, respectively.*

| | PR | KS | IC | SID | ER | ASR | | QbE | SF | | SV | SD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PER ↓ | Acc ↑ | Acc ↑ | Acc ↑ | Acc ↑ | w/o ↓ | w/ LM ↓ | MTWV ↑ | F1 ↑ | CER ↓ | EER ↓ | DER ↓ |
| FBANK | 82.01 | 8.63 | 9.10 | 6.0E-4 | 35.39 | 23.18 | 15.21 | 0.0043 | 69.64 | 52.94 | 11.171 | 10.05 |
| APC [9] | 42.22 | 91.24 | 75.06 | 22.42 | 59.81 | 21.61 | 15.09 | 0.0267 | 71.26 | 50.76 | **10.625** | 11.29 |
| VQ-APC [22] | 42.87 | 90.68 | 71.92 | 16.54 | 59.26 | 21.72 | 15.37 | 0.0224 | 65.72 | 58.60 | 10.699 | 10.49 |
| TERA [19] | 47.53 | 88.64 | 50.94 | **55.19** | 58.16 | 18.45 | 12.44 | 1.29E-4 | 63.28 | 57.91 | 18.791 | 10.71 |
| CPC [11, 35] | 41.66 | 92.02 | 65.01 | 31.38 | 55.57 | 20.02 | 13.57 | 0.0056 | 74.18 | 46.66 | 11.770 | 11.00 |
| wav2vec [12] | 34.45 | 94.32 | 80.04 | 29.43 | 61.10 | 16.40 | 11.30 | 0.0307 | 77.52 | 41.75 | 11.574 | 10.79 |
| vq-wav2vec [14] | 55.12 | 92.79 | 62.04 | 21.42 | 57.88 | 18.70 | 12.69 | 0.0302 | 70.57 | 50.16 | 11.845 | 10.70 |
| wav2vec 2.0 Base [15] | **29.84** | 92.24 | 56.79 | 40.52 | 56.68 | 9.57 | 6.30 | 8.77E-4 | 79.94 | 37.81 | 11.622 | 7.58 |
| wav2vec 2.0 Large [15] | - | - | - | - | - | 16.54 | 10.41 | 3.58E-4 | - | - | - | - |
| HuBERT Base [23] | - | **95.98** | 95.94 | - | 63.96 | 6.74 | 4.93 | **0.0759** | 86.24 | 28.52 | - | 6.76 |
| HuBERT Large [23] | - | 93.15 | **98.37** | - | **66.98** | **3.67** | **2.91** | 0.0360 | **88.68** | **23.05** | - | **6.23** |

Table 2: ***Evaluating various SSL representations on various downstream tasks.*** *Phoneme recognition is denoted as* ***PR****, keyword spotting as* ***KS****, intent classification as* ***IC****, speaker identification as* ***SID****, emotion recognition as* ***ER****, automatic speech recognition as* ***ASR****, query-by-example as* ***QbE****, slot filling as* ***SF****, speaker verification as* ***ASV****, and speaker diarization as* ***SD****.*

SSL representations despite it totally fails in PR. The ranking on PR aligns with ASR to some degree but not completely. VQ-APC surpasses TERA in PR while this is not reflected on ASR. However, we can observe that the significant improvement on phonetics still transfers to ASR, like wav2vec and wav2vec 2.0.

While wav2vec 2.0 performs competitively in ASR, it is at the last second place and even worse than FBANK in QbE, which requires representation to be rich in content by simple distance metrics between queries and documents. HuBERT ranks the top one. The most widely used feature in this task is supervised-pretrained phoneme posteriorgram (PPG). Since our focus is on English here, we implement it with TIMIT which contains ground-truth phoneme boundaries. The result of TIMIT PPG is 0.052 in MTWV, which suggests that HuBERT turns out to be a competitive representation in this task and also ranks high in others concurrently.

As for SF, HuBERT takes the first place, and the improvement is significant for both slot-type F1 and slot-value CER. While TERA achieves the third place for ASR on LibriSpeech, it fails to generalize to SNIPS dataset.

The results of SV does not align with the linear evaluation on SID. TERA performs the best in SID but it is not strong enough in a classic x-vector system. FBANK fails under the standard linear evaluation while is competitive after some non-linear transforms.

As the final speaker task, SD suggests the similar situation, that FBANK surpasses APC, VQ-APC, TERA, and wav2vec in the real-world application. Furthermore, as SD is a complicated task where mixtures of signals are introduced, the results show that most of the SSL representations are not robust enough to deal with signal mixtures. HuBERT dominates this task and the improvement is significant. The decomposition of the DER shows that it has much lower false alarm rate than other upstream methods.

Despite there is a specific SSL pretrained model outperforming all the others in each task, they fail to generalize well to all the tasks with the exception of HuBERT which is best in ASR, SF, KS, IC, ER, QbE, and SD. On the other hand, only APC and VQ-APC can surpass FBANK in SV, demonstrating the research possibilities for developing more powerful and more generalizable SSL pretrained models.

## 5. Conclusion

We present Speech Understanding and PERformance Benchmark (SUPERB), a ten-task challenge to generally benchmark the capability of SSL pretrained models on speech processing. We also propose a simple framework to solve the tasks jointly. The framework utilizes a shared SSL network to extract representations and infers with prediction heads finetuned for each task. We impose constraints to the heads on the architectures, finetuning labels, and computation. The constraints guarantee the simplicity of model development and the generalizability and accessibility of our framework for various tasks. Our results suggest that the framework yields competitive performance in most tasks, and the research direction is promising and of great opportunities. We open-sourced the evaluation toolkit and datasets as a challenge and will release the detailed challenge policy on the leaderboard website. We welcome the community to participate and drive the research frontier.

# 6. References


[1] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding."

[3] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8342–8360.

[4] A. Newell and J. Deng, "How useful is self-supervised pretraining for visual tasks?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[6] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355.

[7] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6391–6400.

[8] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," *Interspeech 2016*, pp. 765–769, 2016.

[9] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An Unsupervised Autoregressive Model for Speech Representation Learning," in *Proc. Interspeech 2019*, 2019, pp. 146–150.

[10] 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," *ICASSP 2020*, May 2020.

[11] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018.

[12] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Interspeech*, 2019.

[13] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *Proc. Interspeech 2019*, pp. 161–165, 2019.

[14] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *International Conference on Learning Representations*, 2020.

[15] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[16] S. Ling and Y. Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[17] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

[18] C.-I. Lai, Y.-S. Chuang, H.-Y. Lee, S.-W. Li, and J. Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," *ICASSP*, 2021.

[19] A. T. Liu, S.-W. Li, and H. yi Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," 2020.

[20] "The Zero Resource Speech Benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," in *Self-Supervised Learning for Speech and Audio Processing Workshop @ NeurIPS*, 2020.

[21] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. de Chaumont Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, "Towards learning a universal non-semantic representation of speech," *Proc. Interspeech 2020*, pp. 140–144, 2020.

[22] Y.-A. Chung, H. Tang, and J. Glass, "Vector-quantized autoregressive predictive coding," *Interspeech 2020*, pp. 3760–3764, 2020.

[23] 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 *Neural Information Processing Systems Workshop on Self-Supervised Learning for Speech and Audio Processing Workshop*, 2020.

[24] 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)*, 2015, pp. 5206–5210.

[25] P. Warden, "Speech commands: A public dataset for single-word speech recognition." *Dataset available online*, 2017.

[26] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *Proc. Interspeech 2019*, pp. 814–818, 2019.

[27] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[28] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[29] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "Gtts-ehu systems for quesst at mediaeval 2014," vol. 1263. Barcelona, Spain: Martha A. Larson et al. (Eds.) CEUR Workshop Proceedings (CEUR-WS.org), October 16-17 2014.

[30] X. Anguera, L. Rodriguez-Fuentes, A. Buzo, F. Metze, I. Szöke, and M. Penagarikano, "Quesst2014: Evaluating query-by-example speech search in a zero-resource setting with real-life queries," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5833–5837.

[31] N. Tomashenko, A. Caubrière, Y. Estève, A. Laurent, and E. Morin, "Recent advances in end-to-end spoken language understanding," in *International Conference on Statistical Language and Speech Processing*, 2019, pp. 44–55.

[32] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[33] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," *Proc. Interspeech 2019*, pp. 4300–4304, 2019.

[34] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Librimix: An open-source dataset for generalizable speech separation," *arXiv preprint arXiv:2005.11262*, 2020.

[35] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.