SUPERB: Speech processing Universal PERformance Benchmark

Shu-wen Yang$^1$ Po-Han Chi*1 Yung-Sung Chuang$^1$ Cheng-I Jeff Lai$^2$ Kushal Lakhotia*3
Yist Y. Lin$^1$ Andy T. Liu$^1$ Jiatong Shih$^4$ Xuankai Chang$^6$ Guan-Ting Lin$^1$
Tzu-Hsien Huang$^1$ Wei-Cheng Tseng$^1$ Ko-tik Lee$^5$ Da-Rong Liu$^1$ Zili Huang$^4$ Shuyan Dong$^{15}$
Shang-Wen Li$^{15}$ Shinji Watanabe$^6$ Abdelrahman Mohamed$^3$ Hung-ye Lee$^1$

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

leo19941227@gmail.com, kushall@fb.com, clai24@mit.edu, jshl34@jhu.edu,
shangwel@amazon.com, shinjiv@ieee.org, hungyilee@ntu.edu.tw

Abstract

Self-supervised learning (SSL) has proven vital for advancing research in natural language processing (NLP) and computer vision (CV). The paradigm pretrains a shared model on large volumes of unlabeled data and achieves state-of-the-art (SOTA) for various tasks with minimal adaptation. However, the speech processing community lacks a similar setup to systematically explore the paradigm. To bridge this gap, we introduce Speech processing Universal PERformance Benchmark (SUPERB). SUPERB is a leaderboard to benchmark the performance of a shared model across a wide range of speech processing tasks with minimal architecture changes and labeled data. Among multiple usages of the shared model, we especially focus on extracting the representation learned from SSL due to its preferable re-usability. We present a simple framework to solve SUPERB tasks by learning task-specialized lightweight prediction heads on top of the frozen shared model. Our results demonstrate that the framework is promising as SSL representations show competitive generalizability and accessibility across SUPERB tasks. We release SUPERB as a challenge with a leaderboard and a benchmark toolkit to fuel the research in representation learning and general speech processing.

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

1. Introduction

Starting from ELMo [1] and BERT [2] in NLP, the effectiveness of SSL is evident in various domains [3-4]. It is becoming a new principle to solve problems by pretraining a shared model with self-supervision tasks on a large amount of unlabeled data to encode general-purpose knowledge. The model can then be specialized in various downstream tasks through concatenating prediction layers and simple finetuning. This approach achieves SOTA performance in many applications.

SSL is desirable for its outstanding performance as well as generalizability and re-usability across tasks to democratize deep learning to various application scenarios. Developing deep neural networks is expensive nowadays in terms of data collection, modeling, computing power, and training time. Repeating the same process for each specific use case is time- and cost-prohibitive for both academic and industrial researchers. SSL can significantly speed up and lower the entry barrier for model development, as the pretrained model is powerful to encode generally applicable knowledge, and only requires low resource to extract task-specific knowledge for different use cases. Well-established benchmark, such as GLUE [5] in NLP and VISSL [6] in CV, is essential to evaluate the generalizability and re-usability of pretrained models across a wide range of tasks.

SSL has been explored in speech, including pretraining with generative loss [7], discriminative loss [11, 12], or multi-task [15, 16]. Researchers have investigated these SSL models’ capabilities on tasks including phoneme classification [11], speaker identification [7, 8], speaker verification [7, 17], emotion recognition [15], ASR [9, 12, 10, 16], speech translation [7], spoken language understanding [18], voice conversion [19] and TTS [20]. While these works showed promising results of SSL on various speech processing tasks, unlike CV or NLP areas, they were investigated with different datasets and experimental setups. Absence of a shared benchmark makes it hard to compare and draw insights across the techniques. Furthermore, existing works explored a limited number of tasks or require heavyweight downstream training [9, 12, 14], blurring the generalizability and re-usability of SSL models across tasks. Both factors limit the impact of SSL on speech processing in research and industry.

We introduce Speech processing Universal PERformance Benchmark (SUPERB) to address the problem. SUPERB aims to 360-degree examine models’ capability and collects various tasks with limited labeled data from speech communities to align with common research interests. There are existing benchmarks proposed to evaluate representations extracted from SSL pretrained models [21, 22]. SUPERB focuses on representations’ quality without any downstream training, and [22] excludes the context recognition tasks like ASR. Compared to existing efforts, SUPERB targets at the direct usability of pretrained models on various popular tasks through any usage. As finetuning pretrained models typically requires huge resources and hinders the re-usability, in this paper, we focus on investigating a simple framework solving all SUPERB tasks with a frozen, shared pre-
trained model, and lightweight prediction heads finetuned for each task. Our results show that the framework yields competitive performance compared to traditional supervised pipelines by leveraging powerful SSL representations, and they outperform log mel filterbank (FBANK), a widely used feature in all speech domains, by a large margin. Both results demonstrate the possibility of developing powerful, generalizable, and reusable pretrained models to democratize the advance in speech processing. We invite researchers to participate and submit new results to drive the research frontier together.

2. Speech processing Universal PERformance Benchmark

We establish and release Speech processing Universal PERformance Benchmark (SUPERB), aiming to offer the community a standard and comprehensive testbed for evaluating the generalizability of pretrained models on various tasks covering all aspects of speech. General speech processing can be categorized into discriminative and generative tasks. The former discriminates from continuous speech into discrete decisions like a match in query-by-example, words in ASR, and classes in speaker identification; the latter generates continuous speech from any input like TTS, voice conversion, and source separation. We focus on the former for the initial release of SUPERB. Tasks are designed with the following principles: (1) standard evaluation protocols from speech communities, (2) publicly available datasets for everyone to participate, (3) limited labeled data to effectively benchmark the generalizability of models. Ten tasks are presented here to investigate four aspects of speech: content, speaker, semantics, and paralinguistics.

2.1. Content

Four tasks are collected from ASR and Spoken Term Detection communities. The former aims to transcribe speech into text content; the latter is to detect the spoken content with minimal effort even without transcribing.

Phoneme Recognition, PR transcribes an utterance into the smallest content units. We include alignment modeling in the PR task to avoid the potential inaccurate forced alignment. LibriSpeech [24] train-clean-100/dev-clean/test-clean subsets are adopted in SUPERB for training/validation/testing. Phoneme transcriptions are obtained from the LibriSpeech official g2p-model-5 and the conversion script in Kaldi librispeech s5 recipe. The evaluation metric is phone error rate (PER).

Automatic Speech Recognition, ASR transcribes utterances into words. While PR analyzes the improvement in modeling phonetics, ASR reflects the significance of the improvement in a real-world scenario. LibriSpeech train-clean-100/dev-clean/test-clean subsets are used for training/validation/testing. The evaluation metric is word error rate (WER).

Keyword Spotting, KS detects preregistered keywords 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 all crucial. We choose the widely used Speech Commands dataset v1.0 [24] for the task. The dataset consists of ten classes of keywords, a class for silence, and an unknown class to include the false positive. The evaluation metric is accuracy (ACC).

Query by Example Spoken Term Detection, QbE detects a spoken term (query) in an audio database (documents) by binary discriminating a given pair of query and document into a match or not. The English subset in QUESST 2014 [25] challenge is adopted since we focus on investigating English as the first step. The evaluation metric is maximum term weighted value (MTWV) which balances misses and false alarms.

2.2. Speaker

Three tasks are collected to analyze speaker modeling.

Speaker Identification, SID classifies each utterance for its speaker identity as a multi-class classification, where speakers are in the same predefined set for both training and testing. The widely used VoxCeleb1 [26] is adopted, and the evaluation metric is accuracy (ACC).

Automatic Speaker Verification, ASV verifies whether the speakers of a pair of utterances match as a binary classification, and speakers in the testing set may not appear in the training set. Thus, ASV is more challenging than SID. VoxCeleb1 [26] is used without VoxCeleb2 training data and noise augmentation. The evaluation metric is equal error rate (EER).

Speaker Diarization, SD predicts who is speaking when for each timestamp, and multiple speakers can speak simultaneously. The model has to encode rich speaker characteristics for each frame and should be able to represent mixtures of signals. LibriMix [27] is adopted where LibriSpeech train-clean-100/dev-clean/test-clean are used to generate mixtures for training/validation/testing. We focus on the two-speaker scenario as the first step. The time-coded speaker labels were generated using alignments from Kaldi LibriSpeech ASR model. The evaluation metric is diarization error rate (DER).

2.3. Semantics

Two tasks are collected from Spoken Language Understanding (SLU) community. While most works for these tasks are done in two stages: transcribing speech into text and predicting semantics on transcribed text, we focus on inferring high-level semantics directly from raw audio in an end-to-end fashion.

Intent Classification, IC classifies utterances into predefined classes to determine the intent of speakers. We use the Fluent Speech Commands [28] dataset, where each utterance is tagged with three intent labels: action, object, and location. The evaluation metric is accuracy (ACC).

Slot Filling, SF predicts a sequence of semantic slot-types from an utterance, like a slot-type FromLocation for a spoken word Taipei, which is known as a slot-value. Both slot-types and slot-values are essential for an SLU system to function [13]. The evaluation metrics thus include slot-type F1 score and slot-value CER [29]. Audio SNIPS [13] is adopted, which synthesized multi-speaker utterances for SNIPS [30]. Following the standard split in SNIPS, US-accent speakers are further selected for training, and others are for validation/testing.

2.4. Paralinguistics

Emotion Recognition, ER predicts an emotion class for each utterance. The most widely used ER dataset IEMOCAP [31] is adopted, and we 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. The evaluation metric is accuracy (ACC).
3. Framework: Universal Representation

There are multiple ways to utilize pretrained models for solving SUPERB tasks. Our framework aims to explore how simple and general the solution can be. Thus, we freeze the parameters of pretrained models across tasks and extract fixed representations to be fed into each task-specialized prediction head (small downstream model). Compared to previous setups in speech representation learning, the framework puts an explicit constraint on downstream models to be as lightweight as possible for all tasks, as their parameter size and required training resources are also crucial for the framework to be simple and re-usable in various use cases. With the above principles, the pretrained model solving all SUPERB tasks in this framework would be a universal representation extractor. In the following, we first describe the SSL pretrained models leveraged and then introduce the downstream models and training policies.

3.1. Self-supervised pretrained models

SSL models explored in this paper are summarized in Table 1 and categorized into three learning approaches: generative modeling, discriminative modeling, and multi-task learning.

**Generative modeling** has long been a prevailing approach to learn speech representation. Instances of generative modeling investigated here include APC [7], VQ-APC [32], Mockingjay [8], TERA [9], and NPC [33]. APC adopts the language model-like pretraining scheme on a sequence of acoustic features (FBANK) with unidirectional RNN and generates future frames conditioning on past frames. VQ-APC further applies vector-quantization (VQ) layers onto APC’s representation to make it compact and low bit-rate. Mockingjay adopts the BERT-like pretraining on Transformer encoders by masking the input acoustic features in time axis and re-generating the BERT -like pretraining on Transformer encoders by masking.

**Discriminative modeling** for SSL studied here include CPC [11, 12], wav2vec2.0 [13], and HuBERT [35]. CPC discriminates the correlated positive samples from negative samples with contrastive InfoNCE loss, which maximizes the mutual information between raw data and representations. Modified CPC [34] and wav2vec [12] proposed several architecture changes to improve CPC. wav2vec2.0 introduces a VQ module to wav2vec2.0. The module discretizes speech into a sequence of tokens after InfoNCE pretraining. Tokens are used as pseudo-text to train a BERT as did in NLP for contextualized representations. wav2vec2.0 merges the pipeline of vq-wav2vec into one end-to-end training scheme by applying time masking in the latent space and replacing BERT’s token classification with InfoNCE’s negative sampling to handle the intractable normalization on continuous speech. Motivated by DeepCluster [56], HuBERT [35] enables BERT’s token classification via off-line unsupervised clustering on acoustic features or learned features from earlier iterations. The continuous audio signals are masked and the corresponding clustered labels are predicted.

**Multi-task learning** is applied in PASE+ [16], where lots of pretraining objectives are adopted: waveform generation, prosody features regression, contrastive InfoMax objectives, and more. Multiple contaminations are also applied to input speech like reverberation and additive noise.

3.2. Downstream models and policies

The framework keeps the downstream models and their fine-tuning as simple as possible while ensuring the performance across pretrained models are comparable and the best model in each task is competitive. For a fair evaluation policy, it limits the space for downstream hyper-parameter tuning. In this paper, we search the best learning rate across 1.0E-1 to 1.0E-7 in log-scale for each SSL representation and downstream. Downstream models or algorithms are summarized in the following and will be released in detail as a part of the challenge policy.

PR, KS, SID, IC, ER are simple tasks that are solvable with linear downstream models. Hence, we use a frame-wise linear transformation for PR with CTC loss to learn the recognition; mean-pooling followed by a linear transformation with cross-entropy loss is utilized for utterance-level tasks (KS, SID, IC, and ER). These 5 tasks also serve as the direct indication of representations’ quality following the conventional representation evaluation protocol.

For ASR, a vanilla 2-layer 1024-unit BLSTM is adopted and optimized by CTC loss on characters. The trained model is decoded with LibriSpeech official 4-gram LM powered by KenLM [57] and flashlight [58] toolkit. SpecAugment [59] is also applied to the representations to avoid overfitting.

We mostly follow the system proposed by GTTS-EHU for QUEST at MediaEval 2014 [60] for QbE but replace the conventional supervised phoneme posteriorgram (PPG) with SSL representations. Representations are extracted for every utterance and normalized along each feature dimension to make the numerical values at the same scale. We run Dynamic Time Warping on the representations with standard distance functions and obtain a score for each query-document pair. The scores belonging to each query are normalized separately.

Regarding SF, slot-type labels are represented as special tokens to wrap the slot-values in transcriptions. SF is then re-formulated as an ASR problem. The finetuning scheme is the same as in our ASR task, except for the pre-processing to encode slot-types into transcriptions and post-processing to decode slot-types and slot-values from hypothesis. As for ASV, we adopt the well-known x-vector [41] as the downstream model, and simply change Softmax loss to AMSSoftmax loss with the same hyper-parameters as [26]. The simple cosine-similarity backend is used to produce pairwise matching scores. We employ the end-to-end training scheme with permutation-invariant training (PIT) loss [62] to SD, instead of using clustering-based methods. We leverage a single-layer 512-unit LSTM for the downstream model.

4. Experiment

To extract representations from pretrained models, in principle we follow the official release as summarized in Table 1 for model definitions, pretrained weights, if released, and representation extraction pipelines, if documented, or in downstream code examples. Some noteworthy details are: (1) NPC’s representations are extracted from the last unmasked CNN as it is shown to be better by the official. (2) NPC repository is used to pretrain APC and VQ-APC as it is more flexible. (3) wav2vec 2

More hyper-parameters will be available to search in the challenge, but there will not be many in principle.

https://dynamictimewarping.github.io/python/

It is preferable for its on-the-fly FBANK extraction to enable testing representations on more corpora. Its APC implementation is mostly the same as the official but with CMVN on FBANK. Its VQ-APC is an
2.0 did not officially release the fixed representation usage. We extract the last-layer representation for the Base model as was done in decor2 0 [10], which showed promising ASR results. (4) For vq-wav2vec, we take the continuous features before the VQ module instead of the representations after BERT. Since its BERT implementation limits the utterance length, which is not long enough for some SUPERB tasks. (5) We take the context vector from wav2vec, as it has shown superior to the latent vector in [18] and our preliminary results. (6) The results for Large models of wav2vec 2.0 and HuBERT will be reported in the following paper update.

The results are presented in Table 2. For the tasks using linear models, it is impossible for FBANK to work on any task and SSL representations all perform well to some degree with different specializations. TERA is good at SID while the case in ER; VQ-APC and NPC behave conversely. It is a surprise that HuBERT conquers PR and IC with just a linear transform instead of linear separability on SID, while HuBERT rules them both again. Under our simple framework, we find that it is non-trivial for SSL representations to generalize to all SUPERB tasks, while HuBERT generalizes well and consistently perform the best. Even more, with only lightweight prediction heads trainable, the framework still achieves highly competitive performance compared to traditional supervised pipeline on many tasks, demonstrating that it is doable and promising to develop more powerful, generalizable and re-usable pretrained models.

5. Conclusion

We present Speech processing Universal PERformance Benchmark (SUPERC), a challenge to generally benchmark the capability of SSL pretrained models on speech processing. We demonstrate a simple framework to solve all SUPERB tasks which leverages a frozen, shared pretrained model and achieves competitive performance with minimal architecture specializations and downstream finetuning. We have open-sourced the evaluation tool[18] and will release the detailed challenge policy on the leaderboard website[19]. We welcome the community to participate and drive the research frontier.

Table 1: Details of investigated SSL representations. LibriSpeech and LibriLight are denoted as LS and LL, respectively. For the pretraining methods, we abbreviate "vector quantization" as VQ, "future" as F, "masked" as M, "generation" as G, "contrastive discrimination" as C, and "token prediction/classification" as P. Parameters for both pretraining and inference are counted.

| Method   | Network                  | #Params    | Stride | Input  | Corpus | Pretraining | Official Github |
|----------|--------------------------|------------|--------|--------|---------|-------------|-----------------|
| FBANK    | -                        | 7.83M      | 10ms   | waveform | -       | -           | -               |
| PASE+ [16] | SimCNet, 1-Conv, 1-GRU   | 4.11M      | 10ms   | waveform | LS 360 hr | F-G         | tiansheng / PASE |
| APC [1]  | 3-GRU                    | 4.63M      | 10ms   | F-BANK | LS 360 hr | F-G + VQ    | liamyunangchung / APC |
| VQ-APC [12] | 3-GRU               | 4.63M      | 10ms   | F-BANK | LS 360 hr | F-G + VQ    | liamyunangchung / VQ-APC |
| NPC [33] | 4-Conv, 4-Masked Conv    | 31.33M     | 10ms   | F-BANK | 800 hr    | M-G         | Alexander-H-Liu / NPC |
| Mockingjay [8] | 12-Trans              | 85.12M     | 10ms   | F-BANK | 360 hr    | time M-G    | -               |
| TERA [9] | 3-Trans                  | 21.33M     | 10ms   | F-BANK | 960 hr    | time M-G    | -               |
| modified CPC [34] | 5-Conv, LSTM | 1.84M      | 10ms   | waveform | LL 60 hr | F-C        | facebookresearch / CPC |
| wav2vec [12] | 19-Conv            | 32.54M     | 10ms   | waveform | LS 960 hr | F-C        | pytorch / fairseq |
| vq-wav2vec [13] | 20-Conv         | 34.15M     | 10ms   | waveform | LS 960 hr | F-C + VQ   | pytorch / fairseq |
| wav2vec 2.0 Base [14] | 7-Conv, 12-Trans | 95.04M     | 20ms   | waveform | LS 960 hr | M-C + VQ   | -               |
| HuBERT Base [15] | 7-Conv, 12-Trans | 94.68M     | 20ms   | waveform | LS 960 hr | M-P + VQ   | under-open-source-process |

Table 2: Evaluating SSL representations on various downstream tasks. The numbers are collected with public-available checkpoints or codes, and we welcome researchers to re-submit the results to our online leaderboard.

| Method   | PR PER | KS Acc | IC w/ LM | ER w/o LM | ASR (WER) | QbE w/ w/o LM | SF MTWV w/ w/ LM | SV CER w/ w/o LM | SD DER w/ w/o LM |
|----------|--------|--------|----------|-----------|----------|---------------|------------------|------------------|------------------|
| FBANK    | 82.01  | 8.36  | 9.10     | 8.5E-4    | 35.39    | 23.18         | 15.21           | 0.0584           | 69.64            | 52.94            | 9.36             | 10.52            |
| PASE+ [16] | 58.88  | 82.37 | 30.29    | 35.84     | 57.64    | 24.92         | 16.61           | 0.0746           | 60.41            | 62.77            | 10.91            | 8.32             |
| APC [1]  | 41.85  | 91.04 | 74.64    | 59.79     | 58.84    | 21.60         | 15.09           | 0.0268           | 71.26            | 50.76            | 8.81             | 10.32            |
| VQ-APC [12] | 42.86  | 90.52 | 70.52    | 49.57     | 58.31    | 21.72         | 15.37           | 0.0205           | 69.62            | 52.21            | 9.29             | 10.49            |
| NPC [33] | 52.67  | 68.44 | 64.04    | 50.77     | 59.55    | 20.94         | 14.69           | 0.0220           | 67.43            | 54.63            | 10.28            | 9.59             |
| Mockingjay [8] | 80.01  | 82.67 | 34.50    | 45.72     | 23.72    | 15.94         | 3.1E-10         | 60.83            | 61.15            | 23.22            | 11.24            |
| TERA [9] | 47.53  | 88.09 | 48.8     | 58.67     | 54.76    | 18.45         | 12.44           | 8.7E-5           | 63.28            | 57.91            | 16.49            | 9.54             |
| modified CPC [34] | 41.66  | 92.02 | 63.01    | 42.29     | 39.28    | 20.02         | 13.57           | 0.0081           | 74.18            | 46.66            | 9.67             | 11.00            |
| wav2vec [12] | 32.39  | 94.09 | 78.91    | 19.38     | 58.17    | 16.40         | 11.30           | 0.0307           | 77.52            | 41.75            | 9.83             | 10.79            |
| vq-wav2vec [13] | 53.49  | 92.28 | 59.4     | 39.04     | 55.89    | 18.70         | 12.69           | 0.0302           | 70.57            | 50.16            | 9.50             | 9.93             |
| wav2vec 2.0 Base [14] | 28.37  | 92.31 | 58.34    | 45.62     | 56.93    | 9.57          | 6.32            | 8.8E-4           | 79.94            | 37.81            | 9.69             | 7.48             |
| HuBERT Base [15] | 6.85   | 95.98 | 95.94    | 64.84     | 62.94    | 6.74          | 4.93            | 0.0759           | 86.24            | 28.52            | 7.22             | 6.76             |
6. References

[1] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in NAACL, 2018, pp. 2227–2237.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in NAACL, 2019, pp. 4171–4186.

[3] S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, “Don’t stop pretraining: Adapt language models to domains and tasks,” in ACL, 2020, pp. 8342–8360.

[4] A. Newell and J. Deng, “How useful is self-supervised pretraining for visual tasks?” in CVPR, 2020.

[5] 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 EMNLP, 2018, pp. 353–355.

[6] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, “Scaling and benchmarking self-supervised visual representation learning,” in ICCV, 2019, pp. 6391–6400.

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

[8] A. T. Liu, S.-W. Yang, P.-H. Chi, P.-C. Hsu, and H.-Y. Lee, “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in ICASSP, 2020.

[9] A. T. Liu, S.-W. Li, and H.-Y. Lee, “Tera: Self-supervised learning of transformer encoder representation for speech,” arXiv preprint arXiv:2007.06028, 2020.

[10] S. Ling and Y. Liu, “DeCoAR 2.0: Deep contextualized acoustic representations with vector quantization,” arXiv preprint arXiv:2012.06559, 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,” in Interspeech, 2019.

[13] A. Baevski, S. Schneider, and M. Auli, “vq-wav2vec: Self-supervised learning of discrete speech representations,” in ICLR, 2020.

[14] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in NeurIPS, 2020.

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

[16] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, “Multi-task self-supervised learning for robust speech recognition,” in ICASSP, 2020, pp. 6989–6993.

[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,” in ICASSP, 2021.

[19] Y. Y. Lin, C.-M. Chien, J.-H. Lin, H.-Y. Lee, and L.-s. Lee, “Fragmentv: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention,” arXiv preprint arXiv:2010.14150, 2020.

[20] D. Álvarez et al., “Problem-agnostic speech embeddings for multi-speaker text-to-speech with samplernn,” in Proc. 10th ISCA Speech Synthesis Workshop, pp. 35–39.

[21] T. A. Nguyen et al., “The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling,” in NeurIPS Workshop on Self-Supervised Learning for Speech and Audio Processing, 2020.

[22] J. Shor, A. Jansen, R. Maor, O. Lang, O. Tuval, F. de Chaumont Quiry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Hayiv, “Towards learning a universal non-semantic representation of speech,” in Interspeech, 2020, pp. 140–144.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in ICASSP, 2015, pp. 5206–5210.

[24] P. Warden, “Speech commands: A public dataset for single-word speech recognition,” Dataset available online, 2017.

[25] X. Anguera, L. Rodriguez-Fuentes, A. Buzo, F. Metze, I. Szőke, and M. Penagarikano, “Quess2014: Evaluating query-by-example speech search in a zero-resource setting with real-life queries,” in ICASSP, 2015, pp. 5833–5837.

[26] A. Nagarani, J. S. Chung, W. Xie, and A. Zisserman, “Voiceceleb: Large-scale speaker verification in the wild,” Computer Speech & Language, vol. 60, p. 101027, 2020.

[27] J. Cosentino, M. Pariente, S. Cornell, A. Delerfoge, and E. Vincent, “Librinix: An open-source dataset for generalizable speech separation,” arXiv preprint arXiv:2005.11262, 2020.

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

[29] N. Tomashenko et al., “Recent advances in end-to-end spoken language understanding,” in International Conference on Statistical Language and Speech Processing, 2019, pp. 44–55.

[30] A. Coucke et al., “Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces,” arXiv preprint arXiv:1805.10190, 2018.

[31] C. Busso et al., “Jemocap: Interactive emotional dyadic motion capture database,” Language resources and evaluation, vol. 42, no. 4, pp. 335–359, 2008.

[32] Y.-A. Chung, H. Tang, and J. Glass, “Vector-quantized autoregressive predictive coding,” in Interspeech, 2020, pp. 3760–3764.

[33] A. H. Liu, Y.-A. Chung, and J. Glass, “Non-autoregressive predictive coding for learning speech representations from local dependencies,” arXiv preprint arXiv:2011.00406, 2020.

[34] M. Rivièrè, P.-E. Mazaré, and E. Dupoux, “Unsupervised pretraining transfers well across languages,” in ICASSP, 2020, pp. 7414–7418.

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

[36] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in ECCV, pp. 132–149.

[37] K. Heafield, “Kenlm: Faster and smaller language model queries,” in Proceedings of the sixth workshop on statistical machine translation, 2011, pp. 187–197.

[38] V. Pratap et al., “Wav2letter++: A fast open-source speech recognition system,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6460–6464.

[39] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” in Interspeech, 2019, pp. 2613–2617.

[40] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, “Gtts-ehu systems for quesst at mediaeval 2014,” in Working Notes Proceedings of the MediaEval 2012 Workshop 2012, 2014.

[41] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in ICASSP, 2018, pp. 5329–5333.

[42] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, “End-to-end neural speaker diarization with permutation-free objectives,” in Interspeech, 2019, pp. 4300–4304.