INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

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 Shi*[4*], *Xuankai Chang*[6], *Guan-Ting Lin*[1], *Tzu-Hsien Huang*[1], *Wei-Cheng Tseng*[1], *Ko-tik Lee*[1], *Da-Rong Liu*[1], *Zili Huang*[4], *Shuyan Dong*[5†], *Shang-Wen Li*[5†], *Shinji Watanabe*[6], *Abdelrahman Mohamed*[3], *Hung-yi Lee*[1]

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

leo19941227@gmail.com, kushall@fb.com, clai24@mit.edu, jshi34@jhu.edu, shangwel@amazon.com, shinjiw@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 for 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[1] and a benchmark toolkit[2] 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 resources 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 pretrained models' generalizability and re-usability across a wide range of tasks.

SSL has been explored in speech, including pretraining with generative loss [7, 8, 9, 10], discriminative loss [11, 12, 13, 14], or multi-task [15, 16]. Researchers have investigated these SSL models' capabilities on tasks including phoneme classification [11, 7], speaker identification [7, 8], speaker verification [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]. [21] focuses on representations'

[*]Equal contribution; sorted alphabetically
[†]Work done independently outside Amazon employment
[1]https://superbbenchmark.org: SUPERB welcomes pretrained model submissions. The framework described in this paper will be used in the constrained track, in which the pretrained models are frozen, and the prediction heads for downstream tasks are the same for all pretrained models. We will open an unconstrained track for submissions with any approach, including finetuning pretrained models and other non-SSL approaches in the future.
[2]https://github.com/s3prl/s3prl: All the materials are open-sourced and reproducible in s3prl toolkit which supports to benchmark most existing and customized pretrained models.

http://dx.doi.org/10.21437/Interspeech.2021-1775

quality without any downstream training, and [22] excludes the content recognition tasks like ASR. Compared to existing efforts, SUPERB targets at the direct usability of pretrained models on various popular tasks through any usage[3]. 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* pretrained 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[1].

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[4]. Tasks are designed with the following principles: (1) conventional 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 [23] 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 [18]. The evaluation metrics thus include slot-type F1 score and slot-value CER [29]. Audio SNIPS [18] 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.

[3]Finetuning pretrained models or using them as representation extractors are two common usages.

[4]SUPERB is a long-term maintained and continuously developing project. More pretrained models will be included in the leaderboard, and we plan to release generative tasks as the second challenge, like voice conversion and source separation.

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 (neutral, happy, sad, angry) 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

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 [9, 12, 13], 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 encoder. 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 [7, 8, 10]. 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 masked parts. TERA extends Mockingjay to further mask the frequency bins. NPC improves the inference speed upon APC by replacing RNN with CNN and changing the future generation to masked reconstruction as Mockingjay.

Discriminative modeling for SSL studied here include CPC [11, 34], wav2vec [12], vq-wav2vec [13], wav2vec 2.0 [14] 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. vq-wav2vec introduces a VQ module to wav2vec. 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. wav2vec 2.0 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 prediction with InfoNCE's negative sampling to handle the intractable normalization on continuous speech. Motivated by DeepCluster [36], HuBERT [35] enables BERT's token prediction via off-line clustering on representations. The clustered labels at the masked locations are then 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

We design our framework to keep the downstream models and their finetuning simple, while ensuring the performance across pretrained models is comparable and the best model in each task is competitive. Since the last-layer representation is not always the best, the framework collects multiple hidden states from the pretrained model and weighted-sum them as the final representation. For a fair comparison, we also limit the space for downstream hyper-parameters tuning[5]. Downstream models and 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; mean-pooling followed by a linear transformation with cross-entropy loss for utterance-level tasks (KS, SID, IC, and ER). These five tasks also serve as the direct indication of representations' quality following the conventional linear 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 [37] and flashlight [38] toolkit. We mostly follow the system proposed by GTTS-EHU for QUESST at MediaEval 2014 [39] for **QbE** but replace the conventional supervised phoneme posteriorgram (PPG) with SSL representations. We run Dynamic Time Warping[40] on all hidden states separately with standard distance functions and obtain a score for each query-document pair. The best distance function / hidden state pair is reported. 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 hypotheses. As for **ASV**, we adopt the well-known x-vector [41] as the downstream model and change Softmax loss to AMSoftmax 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 [42] 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, we follow the official release as summarized in Table 1 for model definitions, pretrained weights, and extraction pipelines if not mentioning specifically. Some noteworthy details are: (1) NPC repository is used to pretrain APC and VQ-APC as it

[5]We search for the best learning rate across 1e-1 to 1e-7 in log-scale for each combination of SSL representation and the downstream tasks. More details about the allowed hyper-parameters tuning will be available as we announce the challenge, but there will not be many hyper-parameters to keep tuning simple.

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	-	0	10ms	waveform	-	-	-
PASE+ [16]	SincNet, 7-Conv, 1-QRNN	7.83M	10ms	waveform	LS 50 hr	multi-task	santi-pdp / pase
APC [7]	3-GRU	4.11M	10ms	FBANK	LS 360 hr	F-G	iamyuanchung / APC
VQ-APC [32]	3-GRU	4.63M	10ms	FBANK	LS 360 hr	F-G + VQ	iamyuanchung / VQ-APC
NPC [33]	4-Conv, 4-Masked Conv	19.38M	10ms	FBANK	LS 360 hr	M-G + VQ	Alexander-H-Liu / NPC
Mockingjay [8]	12-Trans	85.12M	10ms	FBANK	LS 360 hr	time M-G	s3prl / s3prl
TERA [9]	3-Trans	21.33M	10ms	FBANK	LS 960 hr	time/freq M-G	s3prl / s3prl
modified CPC [34]	5-Conv, 1-LSTM	1.84M	10ms	waveform	LL 60k hr	F-C	facebookresearch / CPC_audio
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	pytorch / fairseq
wav2vec 2.0 Large [14]	7-Conv 24-Trans	317.38M	20ms	waveform	LL 60k hr	M-C + VQ	pytorch / fairseq
HuBERT Base [35]	7-Conv 12-Trans	94.68M	20ms	waveform	LS 960 hr	M-P + VQ	pytorch / fairseq
HuBERT Large [35]	7-Conv 24-Trans	316.61M	20ms	waveform	LL 60k hr	M-P + VQ	pytorch / fairseq

Table 2: *Evaluating various 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.*

	PR	KS	IC	SID	ER	ASR (WER)		QbE	SF		ASV	SD
	PER ↓	Acc ↑	Acc ↑	Acc ↑	Acc ↑	w/o ↓	w/ LM ↓	MTWV ↑	F1 ↑	CER ↓	EER ↓	DER ↓
FBANK	82.01	8.63	9.10	8.5E-4	35.39	23.18	15.21	0.0058	69.64	52.94	9.56	10.05
PASE+ [16]	58.87	82.54	29.82	37.99	57.86	25.11	16.62	0.0072	62.14	60.17	11.61	8.68
APC [7]	41.98	91.01	74.69	60.42	59.33	21.28	14.74	0.0310	70.46	50.89	8.56	10.53
VQ-APC [32]	41.08	91.11	74.48	60.15	59.66	21.20	15.21	0.0251	68.53	52.91	8.72	10.45
NPC [33]	43.81	88.96	69.44	55.92	59.08	20.20	13.91	0.0246	72.79	48.44	9.4	9.34
Mockingjay [8]	70.19	83.67	34.33	32.29	50.28	22.82	15.48	6.6E-04	61.59	58.89	11.66	10.54
TERA [9]	49.17	89.48	58.42	57.57	56.27	18.17	12.16	0.0013	67.50	54.17	15.89	9.96
modified CPC [34]	42.54	91.88	64.09	39.63	60.96	20.18	13.53	0.0326	71.19	49.91	12.86	10.38
wav2vec [12]	31.58	95.59	84.92	56.56	59.79	15.86	11.00	0.0485	76.37	43.71	7.99	9.9
vq-wav2vec [13]	33.48	93.38	85.68	38.80	58.24	17.71	12.80	0.0410	77.68	41.54	10.38	9.93
wav2vec 2.0 Base [14]	5.74	96.23	92.35	75.18	63.43	6.43	4.79	0.0233	88.30	24.77	6.02	6.08
wav2vec 2.0 Large [14]	4.75	**96.66**	95.28	86.14	65.64	3.75	3.10	0.0489	87.11	27.31	5.65	**5.62**
HuBERT Base [35]	5.41	96.30	98.34	81.42	64.92	6.42	4.79	**0.0736**	88.53	25.20	**5.11**	5.88
HuBERT Large [35]	**3.53**	95.29	**98.76**	**90.33**	**67.62**	**3.62**	**2.94**	0.0353	**89.81**	**21.76**	5.98	5.75

is more flexible[6]. (2) For vq-wav2vec, we do not propagate through BERT since its BERT implementation limits the utterance length which is not long enough for some tasks.

We present the results in Table 2. For the tasks using linear models, FBANK cannot work on any task, while SSL representations all perform well to some degree with different specializations. It is a surprise that wav2vec 2.0 and HuBERT conquers PR and IC with just linear models and outperforms others by a large margin. Their results on SID and ER are also highly competitive. FBANK achieves competitive performance when allowing non-linear downstream models in ASR, SF, ASV, and SD, and yields better performance than some SSL representations. We also observe that the ranking on PR aligns with ASR weakly, while a significant improvement on phonetics still transfers to ASR, like wav2vec, wav2vec 2.0, and HuBERT. Furthermore, wav2vec 2.0 and HuBERT demonstrate that it becomes much easier than before to train an ASR system by leveraging powerful SSL representations. HuBERT ranks the top one in QbE with MTWV 0.074. The prevailing feature for QbE is PPG which we implemented with TIMIT due to the current focus on English, and the result of TIMIT PPG is 0.052 in MTWV, suggesting that HuBERT turns out to be a very competitive representation for QbE. As for SF, we can also observe a significant improvement from wav2vec 2.0 and HuBERT over all other representations. The CER in SF is generally high compared to ASR as many slot-values are named entities. The results in ASV and SD show that many SSL representations are worse than FBANK when it comes to real-world speaker problems beyond SID, while HuBERT improves upon popular FBANK from 9.56 to 5.10 without additional VoxCeleb2 or augmentation. Although we find it non-trivial for SSL representations to generalize to all SUPERB tasks, wav2vec 2.0 and HuBERT achieve highly competitive performance with only lightweight prediction heads trainable, compared to traditional supervised techniques. The experiment results exhibit the efficacy of developing a more generalizable and re-usable pretrained model.

[6]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 improved version as stated in the official repository.

5. Conclusion

We present SUPERB, 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 changes and downstream finetuning. We have open-sourced the evaluation toolkit[2] and will release the detailed challenge policy on the leaderboard website[1]. We welcome the community to participate and drive the research frontier.

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. 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 *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," *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:2012.06659*, 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, "Fragmentvc: 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 Quitry, M. Tagliasacchi, I. Shavitt, D. Emanuel, and Y. Haviv, "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. Szöke, and M. Penagarikano, "Quesst2014: Evaluating query-by-example speech search in a zero-resource setting with real-life queries," in *ICASSP*, 2015, pp. 5833–5837.

[26] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language*, vol. 60, p. 101027, 2020.

[27] 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.

[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.*, "Iemocap: 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. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP*, 2020, pp. 7414–7418.

[35] 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," *arXiv preprint arXiv:2106.07447*, 2021.

[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] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "Gtts-ehu systems for quesst at mediaeval 2014," in *MediaEval*, 2014.

[40] T. Giorgino, "Computing and visualizing dynamic time warping alignments in R: The dtw package," *Journal of Statistical Software*, vol. 31, no. 7, pp. 1–24, 2009.

[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.