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

Using Large Self-Supervised Models for Low-Resource Speech Recognition

Krishna D N, Pinyi Wang, Bruno Bozza

Freshworks Inc, India
{krishna.nanjappa, pinyi.wang, bruno.bozza}@freshworks.com

Abstract

Recently, self-supervised pre-training has shown significant improvements in many areas of machine learning, including speech and NLP. The self-supervised models are trained on a large amount of unlabelled data to learn higher-level representations for downstream tasks. In this work, we investigate the effectiveness of many self-supervised pre-trained models for the low-resource speech recognition task. We adopt pre-trained wav2vec2.0 [1] models for the speech recognition task for three Indian languages Telugu, Tamil, and Gujarati. We examine both English and multilingual pre-trained models. Our experiments show that fine-tuning the multilingual pre-trained model obtains an average relative reduction in WER of 2.88% compared to the previous state-of-the-art supervised method. We carefully analyze the generalization capability of multilingual pre-trained models for both seen and unseen languages. We also show that fine-tuning with only 25% of the training data gives competitive WER to the previous best methods.

Index Terms: self-supervision, low-resource speech recognition, pre-training, wav2vec2.0

1. Introduction

Studies have shown that more than 22 main spoken languages in India, and apart from Hindi and Indian English, many other languages are considered low resources due to the scarcity of the data and the speaker population. Many Indian languages have acoustic similarity, and it makes sense to use a multilingual speech recognition approach. Conventional or Hybrid speech recognition models use a combination of Gaussian Mixture Models(GMMs) with Hidden Markov Models (HMMs) or Deep neural networks (DNNs) with HMMs. Hybrid approaches have been proposed in the past [3,4,5] for multilingual speech recognition. These approaches use initial layers of the acoustic models as feature extractors for language-dependent layers during adaptation.

Conventional speech recognition frameworks such as hybrid models consist of an acoustic model, pronunciation model, and language model. In the case of E2E speech recognition, acoustic models, pronunciation dictionaries, and language models are integrated into a single framework. Sequence-to-sequence [6] models are a particular class of E2E models, which use encoder and decoder framework to learn a mapping from acoustic data to text. [7] proposed an attention-based E2E model called LAS using LSTM based encoder and decoder. Multilingual end-to-end models have been explored for multilingual speech recognition tasks [8,9]. Recently, Transformers [10] have shown state-of-the-art performance for many Natural language processing problems [19]. These transformer models are also being used in many areas of speech, including speech recognition [1,2], emotion recognition [12], spoken language translation [13], etc. Even though the E2E framework simplifies speech recognition using a single framework, these models require a large amount of labeled data to outperform hybrid models. Especially to learn speaker and domain invariant representations, the training data should contain a variety of speech data from multiple speakers, different domains, and multiple accents [14,15].

Collecting a large amount of labeled speech data is very expensive and time-consuming. Recent advancements in self-supervised learning help us leverage unlabelled data to pre-train high capacity models for learning higher-level speech representations, which can then be used for the downstream tasks [16,17,18]. For speech, many researchers have explored different self-supervised learning objectives such as Contrastive Predictive Coding (CPC) [19], Autoregressive Predictive Coding (APC) [20], Masked Predictive Coding(MPC) [21], and Problem agnostic speech encoder(PASE) [22] for learning better speech representations from unlabeled data. The CPC-based approach is being used in many speech models these days. It helps the model learn high-level representation by predicting future frames (acoustic feature or latent representations) conditioned on the current frame. Recently, many neural network architectures have been proposed for self-supervised pre-training using CPC. wav2vec [23] is one such architecture where it learns latent features from raw audio waveform using initial Convolution layers followed by autoregressive layers (LSTM or Transformer) to capture contextual representation. [24] proposed to use quantization layers for wav2vec to learn discrete latent representations from raw audio. The recently proposed wav2vec2.0 [1] model shows that by combining BERT [17] style sequence modeling and discrete CPC training with a large amount of unlabelled English audio data, we can obtain very powerful self-supervised models for ultra low resource speech recognition. A similar architecture is used to pre-train a multilingual model [25] to see the model's capability on cross-lingual feature transfer.

This paper investigates the effectiveness of large self-supervised pre-trained models for low-resource and ultra-low resource speech recognition tasks for three Indian languages Telugu, Tamil, and Gujarati. We use the pre-trained models as base feature extractors and fine-tune them with a single projection layer followed by a CTC. We observe that multilingual fine-tuning consistently obtains better performance than the monolingual English pre-trained model trained on 50K Hrs of audio data. We also observe that the monolingual English model trained on 1000Hrs of Librispeech gives a competitive performance with a multilingual model trained on 50K Hrs of audio data. We also investigate the generalization capability of the multilingual model for unseen languages during the pre-training stage. We compare our approach to the state-of-the-art supervised methods [2]. Our experiments show that with just 25% of the training data, we can obtain WER close state of the art for both seen and unseen languages.

The organization of the paper is as follows. Section 2 explains our methodology in detail. In section 3, we give details of

http://dx.doi.org/10.21437/Interspeech.2021-631

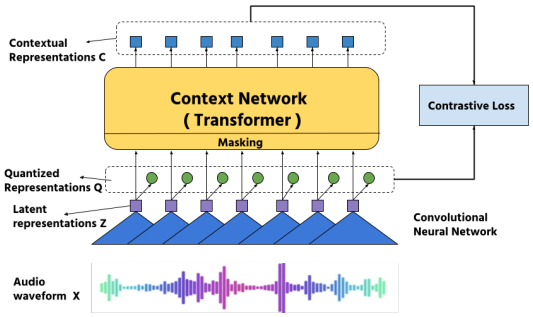

Figure 1: *Wav2vec2.0 Architecture.*

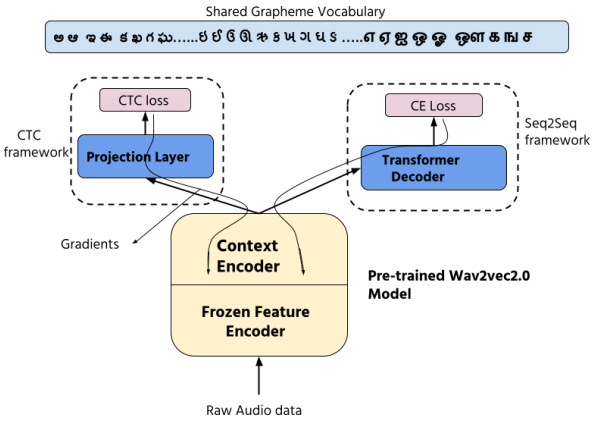

Figure 2: *Adaptation of wav2vec2.0 for ASR task*

the dataset. Finally, section 4 describes our experimental setup in detail.

2. Methodology

First, we explain the architecture of the wav2vec2.0 [1] model, and then we describe the self-supervised pre-training objective proposed in [1]. Finally, we describe the adaptation of pre-trained models to downstream speech recognition tasks.

2.1. Wav2Vec2.0 Architecture

Recently, self-supervised models such as BERT, GPT-2 have shown promising results for many natural language processing tasks. These models learn robust high-level representations from large unlabelled corpora and could be helpful for downstream applications. In the speech domain, wav2vec2.0 is one such architecture where we can leverage this model for the ultra low-resource speech recognition task. The wav2vec2.0 consists of three main components a Feature encoder, a context encoder, and a quantization module, as shown in Figure 1. The feature encoder module consists of temporal convolution blocks followed by layer normalization and GELU activations. It takes raw waveform $\mathbf{X}$ as input and outputs latent speech representation $\mathbf{Z}$. The feature encoder helps the model learn latent representation directly from raw audio every 20ms (assuming 16Khz sampling rate). The context encoder block helps capture contextual information and high-level feature representations from the latent feature sequence $\mathbf{Z}$. It consists of multilayer self-attention blocks similar to transformer [10], but instead of fixed positional encoding, the model uses relative positional encoding. The context encoder block takes masked latent vectors $\mathbf{Z}$ as input and builds contextualized representations $\mathbf{C}$ as outputs. The quantization block quantizes the continuous latent representations $\mathbf{Z}$ into quantized representation $\mathbf{Q}$. [24] shows that learning discrete latent representation is more helpful compared to continuous representation. Incorporating quantization into neural networks could be tricky because of the non-differentiable nature of quantization. Gumble softmax [26] trick is used to sample the codebook vectors, and this helps in making the model fully differentiable. In theory, these discrete units $\mathbf{Q}$ should resemble the acoustics units like phonemes or syllables of a language.

2.2. Self-Supervised Learning Objective

Pre-training neural networks with unlabelled data have been one the main challenge in machine learning. Reconstruction loss and similarity loss have been explored in the past to learn from unlabelled data. Recently, CPC has shown great promise for self-supervised pre-training tasks. Masked predictive coding (MPC) [21], on the other hand, adopts BERT style pre-training where the model predicts the missing parts of the input. In wav2vec2.0, Contrastive loss or CPC is the primary training objective during the pre-training stage. The contrastive loss is also augmented with diversity loss to use the codebook entries efficiently. The masking technique randomly masks out parts of latent representation $\mathbf{Z}$ before passing it to the context encoder. During pre-training, the contrastive learning objective forces the model to distinguish the quantized representation at masked time steps from a set of distractors from other time steps. This training technique seems to be quite powerful for learning better speech representation.

2.3. Adaptation to ASR tasks

Learning representations from unlabeled data using self-supervision techniques yield good feature extractors for downstream tasks. The fine-tuning process is described in Figure 2. These pre-trained models act as an encoder during the adaptation stage. There are mainly two approaches for fine-tuning the pre-trained models. 1) CTC framework: A randomly initialized projection layer is added on top of the context network of the pre-trained model to predict output units for every frame. 2)Seq2Seq framework: Considering the pre-trained model as an encoder, we can connect it to a sequential decoder network to predict the sequence of output units conditioned on the contextualized representations $\mathbf{C}$. In our case, we experiment with both approaches. In the CTC framework, during fine-tuning, only the last projection layer is trained for the first 10K iterations, and the remaining network is frozen (no back-propagation). After the initial 10K iterations, the context network (transformer) gets updated along with the projection layer. Similarly, in the Seq2Seq framework, only the decoder is trained for the first 10K iteration by keeping the rest of the network frozen. After 10K iterations, both decoder and context networks get updated jointly.The output units can be characters (grapheme units of the language), phonemes, or subwords of a

particular language. In this work, we explore only the character units due to computation constraints keeping the other methods for future work. In the case of multilingual fine-tuning, the output vocabulary is shared across all the languages, and in the case of monolingual fine-tuning, we use a language-specific vocabulary. We explore both monolingual and multilingual fine-tuning scenarios for three Indian languages.

3. Dataset

We conduct our experiments on the data set released by Microsoft and SpeechOcean.com as part of a special session on "Low resource speech recognition challenge on Indian languages in INTERSPEECH 2018". The dataset includes data for three Indian languages - Telugu (TE), Tamil(TA), and Gujarati(GU). All the audio files are sampled at 16KHz, and the transcriptions contain UTF-8 text for every language. The dataset contains train, development, and evaluation sets for each language. Unfortunately, the transcripts for the evaluation sets are not made public. Due to this reason, we report all our results on the development set. Each language contains 40hrs of training data and 5hrs development data. We use 3hrs of randomly selected audio files from the train set from each language for the validation set. We use characters as the output units for each language. Tamil (TA) language has 48 character units, Telugu (TA) has 58 unique characters, and Gujarati has 63 character units. We report all our results on the development set and compare our model performance with previous research published on the same set.

4. Experimental Setup

We use publicly available wav2vec2.0 models[1] for our experimentation. We use *wav2vec2.0-base*, *wav2vec2.0-large* and *XLSR* models for the fine-tuning experiments. *wav2vec2.0-base* model is pre-trained on ∼1000 hrs of Librispeech dataset [28]. *wav2vec2.0-base* model contains seven temporal convolution layers followed by layer normalization and GELU activation. It uses 12 self-attention layers with eight attention heads. On the other hand, *wav2vec-2.0-large* model is trained on ∼50K hrs of LibriVox[2] data, and the model has seven layers of temporal convolution and 24 self-attention layers with 16 attention heads each. Finally, the *XLSR* model is a multilingual wav2vec2.0 model trained with more than ∼53K hrs of multilingual data. The model architecture of *XLSR* is same as *wav2vec2.0-large* but the training data comes from various sources such as MLS [27], CommonVoice[3] and BABEL[4]. We conduct fine-tuning experiments with these three self-supervised models to understand their adaptation capability for different languages in low-resource settings. We use two different fine-tuning methods 1) Using the CTC framework and 2) using the seq2seq framework. In the CTC framework, we use a projection layer with an output size that is the same as the length of the vocabulary. In the case of the seq2seq framework, we use a two-layer transformer decoder. We fine-tune our models on 1 NVIDIA RTX 3090 GPU for less than 10 hours using Adam [32] optimizer. We warm up the learning rate for 8000 steps to a peak of 2x10-5, hold it for 32000 steps. We train all the models up to 100000 iterations with an exponential decay learning rate. Most of the configurations are inherited from the fairseq library [5].

[1]https://dl.fbaipublicfiles.com/fairseq/wav2vec/

[2]https://librivox.org/

[3]https://commonvoice.mozilla.org/

[4]http://www.reading.ac.uk/AcaDepts/ll/speechlab/babel/

4.1. Comparing Pre-trained Models

In order to understand the effectiveness of different pre-trained models, we fine-tune three different pre-trained models for three Indian languages Telugu (TE), Tamil (TA), and Gujarati(GU). We add a projection layer on top of every pre-trained model and fine-tune the ASR model with CTC loss. The results of our experiments are shown in Table 1. It can be seen that the multilingual pre-trained model outperforms all the other pre-trained models across all languages. This is due to the fact that the *XLSR* model has rich cross-lingual representations. On the other hand, the *wa2vec2.0-base*+CTC shows comparable performance and has a good transferability to all three Indian languages. We observe some intriguing results for *wa2vec2.0-base*+CTC model, as can be seen from Table 1. Even though both *wa2vec2.0-base*+CTC and *wa2vec2.0-large*+CTC models are trained on English, fine-tuning small pre-trained model seems to outperform the large *wa2vec2.0-large*+CTC. These results seem interesting and tell us that even the small pre-trained models can provide good cross-lingual transfer for low-resource languages. For the experiment *wa2vec2.0-large*+CTC+LM, we use a 3-gram language model during CTC decoding, and it can be seen that the language model integration consistently gives a significant improvement for all the languages.

Table 1: *WER comparasion with various pre-trained models. GU (Gujarati), TE (Telugu) and TA (Tamil). Bold indicates the best WER (lower the better)*

Models	GU	TA	TE
wa2vec2.0-base+CTC	31.2	34.41	36.9
wa2vec2.0-large+CTC	37.0	39.05	43.71
XLSR+CTC	29.71	33.20	36.27
XLSR+CTC+LM	**22.92**	**31.92**	**22.90**

4.2. Supervised vs Self-Supervised

In this section, we compare our self-supervised approach with the latest supervised approaches [2] proposed on the same dataset. We use state-of-the-art end-to-end models for supervised training for each language separately. We use BILSTM, Transformers [2], and Conformer [29] models for the supervised experiments. The results of the supervised and self-supervised models are shown in Table 2. For *Transformer-Multitask*) (row 2 in table 2) and *Conformer-Multitask* (row 3 in Table 3). We use 12 layer encoder and one layer decoder with eight attention heads for each layer for *Transformer-Multitask*. We use 8 layer encoder and one layer decoder with eight attention heads for each layer for *Conformer-Multitask*. The *BILSTM-Multitask* has four Bi-LSTM layers in the encoder network and a single Bi-LSTM layer in the decoder. All the supervised models are trained with a joint CTC-Attention multitask learning framework [30]. We can see that the multilingual pre-trained model *XLSR*+CTC outperforms state-of-the-art supervised system *Transformer-Multitask* for Gujarati and Tamil. It can also be seen that *Conformer-Multitask* obtains the best WER for Telugu. We observe that the *wav2vec2.0-base*+CTC outperforms the previous best approach for Gujarati and gets degradation for Telugu and Tamil. Note that language models are not used during decoding for any supervised systems.

[5]https://github.com/pytorch/fairseq

Table 2: *WER comparison between supervised and self-supervised approaches*

Models	GU	TA	TE
Supervised			
BILSTM-Multitask[3]	45.3	39.6	43.9
Transformer-Multitask[3]	31.9	34.1	36.0
Conformer-Multitask	30.23	33.71	**34.80**
Self-Supervised			
wa2vec2.0-base+CTC	31.2	34.41	36.9
XLSR+CTC	**29.71**	**33.20**	36.27

4.3. Effect of training data size

In this section, we describe the effect of the amount of training data on the word error rate for Gujarati, Tamil, and Telugu. We split the training data into four parts 10hrs, 20hrs, 30hrs, and 40hrs for each language. We fine-tune *XLSR*+CTC for each part, and we test it on the development set for the corresponding language. The results are shown in Figure 3. Tamil is considered to be the seen language because the XLSR model has seen Tamil as part of BABEL during its pre-training stage. In contrast, Telugu and Gujarati are considered unseen languages. It can be seen that, due to the presence of the Tamil language during pre-training, there are no significant changes between 20hrs,30hrs, and 40hrs models for Tamil. Since the model has learned good speech representations for the Tamil language, we can reach the state-of-the-art WER with only 50% of the training data (20hr model). In case unseen languages such as Gujarati and Telugu, the model gains up to 2-3% WER (absolute) for every additional 10hrs of data.

4.4. Multilingual Adaptation vs Monolingual Adaptation

In this section, we compare multilingual fine-tuning with monolingual fine-tuning. In monolingual fine-tuning, the pre-trained models are adapted for each language separately with their respective vocabulary (set of grapheme units of the language). In comparison, multilingual fine-tuning uses a shared vocabulary and trains a single model for all the languages. The shared vocabulary contains grapheme units of all three languages combined, and the length of the vocabulary is 169 character units. We do not provide any language label information to the model during training. The WER values for the monolingual model vs. multilingual models are shown in Table 3. It can be seen that multilingual fine-tuning doesn't perform as well as monolingual fine-tuning. We use both greedy CTC decoding and language model decoding for the multilingual model. As per Table 3, The *XLSR-Multi+CTC*+greedy is a model trained with multilingual CTC (shared vocab) with greedy decoding, and *XLSR-Multi+CTC*+LM is a model trained with multilingual CTC but with 3-gram Language model-based decoding. We train a 3-gram multilingual language model for this experiment, and we use KenLM[6] toolkit to train the language models.

4.5. CTC framework vs Seq2Seq framework

In this section, we compare the CTC fine-tune framework with seq2seq fine-tune framework. For the CTC framework, we add a single projection layer on top of the last transformer layer of the context network. The output size of the projection layer is the same as the length of the vocabulary. In the case of the seq2seq framework, we attach a two-layer decoder to the pre-trained models. In this framework, the pre-trained model acts as the encoder. Table 4 shows the results of this experiment. We find that seq2seq models cannot achieve better results compared to CTC models. This may be due to the fact that the amount of training data for each language is very low, and the seq2seq model couldn't generalize well.

[6]https://github.com/kpu/kenlm

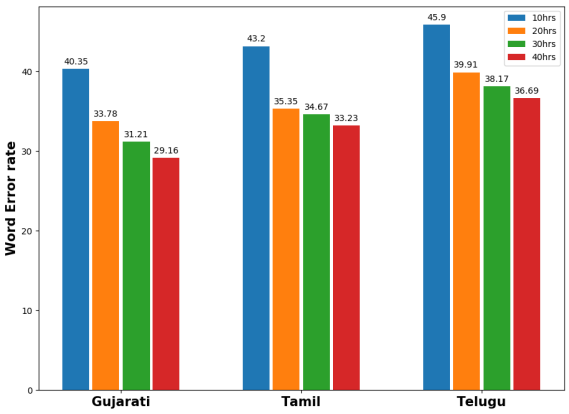

Figure 3: *Bar plot of WER w.r.t training data size*

Table 3: *Monolingual vs multilingual fine-tuning*

Models	GU	TA	TE
XLSR-Mono-CTC+greedy	29.71	33.20	36.27
XLSR-Mono-CTC+LM	**22.92**	**31.92**	**22.90**
XLSR-Multi-CTC+greedy	35.82	39.05	43.71
XLSR-Multi-CTC+LM	25.27	34.82	34.50

Table 4: *CTC framework vs Seq2Seq framework.*

Models	GU	TA	TE
wav2vec2.0-base+CTC	31.2	34.41	36.9
wav2vec2.0-base+seq2seq	36.12	42.89	39.20
XLSR+CTC	**29.71**	**33.20**	**36.27**
XLSR+seq2seq	35.32	41.01	38.40

5. Conclusions

Building speech recognition systems for low-resource languages from scratch is a challenging problem in speech processing. In this work, we examined multiple self-supervised models for ASR adaptation for Indian languages to understand their cross-lingual adaptation capability. We adapted both English and multilingual pre-trained models and observed that multilingual pre-trained models obtain the best performance due to their rich cross-lingual feature representation. We explored multiple adaptation strategies and showed that fine-tuning with a simple projection layer obtains a very good WER. We compared the state-of-the-art transformer-based supervised speech recognition systems with our adaptation methods. We also showed that, for seen languages Tamil, we could obtain WER close to state of the art with only 50% of the training data. In future work, we would like to study the role of each context network layer and its effect on system performance.

6. References

[1] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[2] V. M. Shetty, M. Sagaya Mary N J, and S. Umesh, "Improving the performance of transformer based low resource speech recognition for Indian languages," in *IEEE ICASSP*, 2020, pp. 8279–8283.

[3] Karel Vesely, Martin Karafiat, Frantisek Grezl, MilosJanda, and Ekaterina Egorova, "The language independent bottleneck features," in 2012 *IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 336–341.

[4] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in 2013 *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7319–7323.

[5] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in 2013 *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7304–7308.

[6] Ilya Sutskever, Oriol Vinyals, and Quoc Le, "Sequence to Sequence Learning with Neural Networks", In *Neural Information Processing Systems*, 2014.

[7] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in 2016 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[8] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, "Multilingual speech recognition with a single end-to-end model," in 2018 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4904–4908.

[9] Shinji Watanabe, Takaaki Hori, and John R Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in 2017 *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 265–271.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[11] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rmi Louf, Morgan Funtowicz, and Jamie Brew. 2019. "HuggingFace's Transformers: State-of-the-art natural language processing." *arXiv:1910.03771*.

[12] N., K.D., Patil, A. (2020) "Multimodal Emotion Recognition Using Cross-Modal Attention and 1D Convolutional Neural Networks." *Proc. Interspeech 2020*, 4243-4247.

[13] Mattia Di Gangi, Matteo Negri, and Marco Turchi, "Adapting Transformer to end-to-end spoken language translation," in *Interspeech*, 2019.

[14] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. "Deep speech 2: End-to-end speech recognition in english and mandarin", In *International Conference on Machine Learning*, pp. 173–182, 2016.

[15] T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve, "Rethinking evaluation in asr: Are our models robust enough?" *arXiv preprint* arXiv:2010.11745, 2020.

[16] Jaiswal, Ashish, A. R. Babu, Mohammad Zaki Zadeh, D. Banerjee and F. Makedon. " A Survey on Contrastive Self-supervised Learning." *ArXiv* abs/2011.00362 (2020):

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACLHLT*, 2019.

[18] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019) "Language models are unsupervised multitask learners". *OpenAI Blog*, 1 (8), 9.

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

[20] C. Yu-An, H. Wei-Ning, T. Hao, and G. James, "An unsupervised autoregressive model for speech representation learning," *Interspeech* 2019, Sep 2019.

[21] Ruixiong Zhang, Haiwei Wu, Wubo Li, Dongwei Jiang, Wei Zou, and Xiangang Li, "Transformer based unsupervised pre-training for acoustic representation learning," *CoRR*, vol. abs/2007.14602, 2020.

[22] Santiago Pascual, Mirco Ravanelli, Joan Serra, Antonio Bonafonte, and Yoshua Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Interspeech*, 2019.

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

[24] A. Baevski, S. Schneider, and M. Auli. "vq-wav2vec: Self-supervised learning of discrete speech representations". In *Proc. of ICLR*, 2020.

[25] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv:2006.13979,* 2020.

[26] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax." *arXiv, abs/1611.01144*, 2016.

[27] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. "MLS: A Large-Scale Multilingual Dataset for Speech Research". In *Proc. Interspeech 2020*, pages 2757–2761.

[28] 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.

[29] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020

[30] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-toend speech recognition using multi-task learning," in Proc. *ICASSP*, 2017, pp. 4835–4839.

[31] Daniel S Park, Yu Zhang, Chung-Cheng Chiu, Youzheng Chen, Bo Li, William Chan, Quoc V Le, and Yonghui Wu, "Specaugment on large scale datasets ", In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6879–6883. IEEE, 2020.

[32] Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization", In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014

[33] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, "Automatic differentiation in PyTorch", in *NIPS*, 2017