Interspeech 2022
18-22 September 2022, Incheon, Korea

Investigating Self-supervised Pretraining Frameworks for Pathological Speech Recognition

Lester Phillip Violeta, Wen-Chin Huang, Tomoki Toda

Nagoya University, Japan

violeta.lesterphillip@g.sp.m.is.nagoya-u.ac.jp

Abstract

We investigate the performance of self-supervised pretraining frameworks on pathological speech datasets used for automatic speech recognition (ASR). Modern end-to-end models require thousands of hours of data to train well, but only a small number of pathological speech datasets are publicly available. A proven solution to this problem is by first pretraining the model on a huge number of healthy speech datasets and then fine-tuning it on the pathological speech datasets. One new pretraining framework called self-supervised learning (SSL) trains a network using only speech data, providing more flexibility in training data requirements and allowing more speech data to be used in pretraining. We investigate SSL frameworks such as the wav2vec 2.0 and WavLM models using different setups and compare their performance with different supervised pretraining setups, using two types of pathological speech, namely, Japanese electrolaryngeal and English dysarthric. Our results show that although SSL has shown success with minimally resourced healthy speech, we do not find this to be the case with pathological speech. The best supervised setup outperforms the best SSL setup by 13.9% character error rate in electrolaryngeal speech and 16.8% word error rate in dysarthric speech.

Index Terms: speech recognition, self-supervised learning, pathological speech

1. Introduction

Automatic speech recognition (ASR) is the task of converting speech audio into its text transcripts. Early methodologies trained linguistic, acoustic, and lexicon modules separately [1, 2]; however, up until a decade ago, neural networks showed an improvement of performance while also being significantly easier to train, with all of the modules trained altogether in an end-to-end fashion [3, 4]. With the recent successful use of end-to-end models in ASR, many of these have become commercially available as products, bringing in many benefits and conveniences to humans by enabling the control of several devices through speech. With such conveniences, ASR has the potential to benefit many people; however, disabled patients, who would benefit the most from these devices, are hindered from using these owing to their loss of control of their speech organs [5]. One common speech pathology is dysarthria, which makes them speak at a slower rate and with a slurred characteristic. Another speech pathology could be due to laryngectomy, which is the replacement of the larynx with an electro-larynx to recreate its function, but results in robotic-like speech production. In both of these speech pathologies, communication becomes difficult, which causes the produced speech to be misunderstood by most commercial ASR devices [6]. Knowing the benefits of ASR, much more research to make ASR accessible to speakers with speech pathologies should be carried out. Unfortunately, only having a small number of publicly available datasets, coupled with the huge data requirements to train networks, have made this a difficult task.

One mainstream method to resolve minimally resourced ASR is by pretraining, which is training the network on a large dataset of healthy speech before fine-tuning on limited pathological speech datasets. Investigations such as those in [7] show drastic improvements in performance through a pretraining framework using a Recurrent Neural Network Transducer (RNN-T) pretrained on 1,000 hours of healthy speech data, even with only less than an hour of pathological speech in the fine-tuning stage. Building on this study, increasing pretraining data to roughly 162,000 hours in [8, 9] makes personalized RNN-T models become capable of outperforming human listeners on speech with low intelligibility rates and allow 90% of speakers to reach 15% word error rate even with only using less than 250 utterances in fine-tuning. In the aforementioned investigations, using a good pretraining strategy and more data proves that the network can adapt to the pathological speech features easily even when only fine-tuned on a small number of data. However, one limitation in the previously mentioned works is that the model is pretrained in a supervised fashion, which heavily relies on having a dataset that provides both the speech audio and its corresponding text labels to succeed. Although many speech data has been made publicly available, large-scale data labeling is still a very tedious task, making it not a sustainable pretraining method for use in the future.

In this paper, we investigate a novel pretraining method called self-supervised learning, which provides flexibility in the training requirements by removing the need for labeled data to train the network. The general idea of this method is to make the model learn a universal representation of the data using joint embedding objective functions [10, 11] to compare embedding representations of the data. For speech in particular, several frameworks have been proposed [12] such as generative modeling [13], discriminative/contrastive modeling [14, 15, 16], and multitask learning [17]. Research in [18] has shown contrastive-learning-based SSL pretraining capable of outperforming supervised pretraining on dysarthric speech; however, the research could still be further extended by using more SSL frameworks, while also using other supervised pretraining setups as baseline.

We contribute the following points in this investigation:

- We investigate two self-supervised learning pretraining frameworks (contrastive and masked region prediction) for pathological speech recognition and compare them with supervised pretraining.
- We analyze two types of pathological speech, namely, electrolaryngeal and dysarthric.
- We show that although SSL pretraining has been successful on minimally resourced healthy speech, extensive experiments show that this is not exactly the case for pathological speech.

Copyright © 2022 ISCA

10.21437/Interspeech.2022-10043

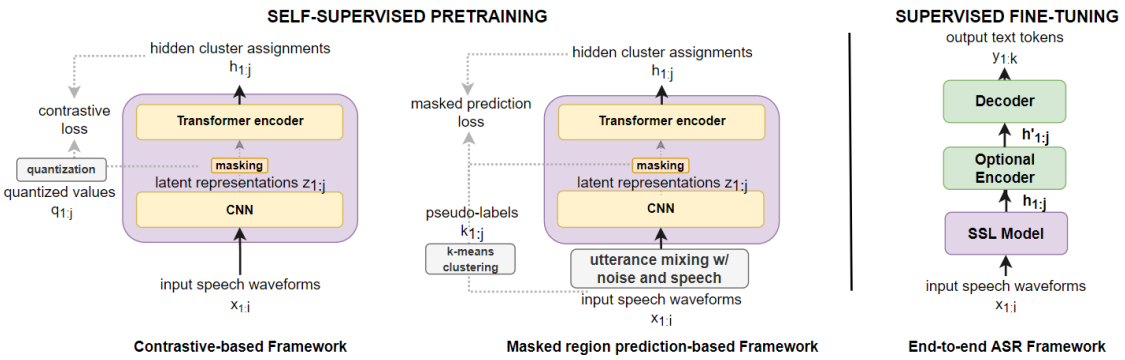

Figure 1: *Overview of SSL training. The SSL models wav2vec 2.0 [14] (left) and WavLM [16] (middle) are first pretrained by a contrastive or masked region prediction framework. The SSL model is then fine-tuned by supervised training, treated similarly to an end-to-end model (right).*

2. Using self-supervised learning pretraining frameworks for ASR

With the successful use of previous pretraining methods, we explore the effectiveness of a new pretraining framework called self-supervised learning (SSL). In this section, we present how SSL models learn speech representations and how they can be used for ASR tasks. An overview of using SSL models, which is divided into pretraining and fine-tuning stages, is shown in Figure 1.

2.1. Pretraining SSL models to learn speech representations

We investigate two self-supervised pretraining frameworks that are similar to the one used in [18]: contrastive loss [14] and masked region prediction [16]. The first framework is wav2vec 2.0 [14], one of the most widely used SSL frameworks in speech data based on contrastive loss and proven to be successfully used in different minimally resourced speech recognition tasks [19, 20]. This framework uses a multilayer convolutional neural network (CNN) to encode speech into a latent space $z_{1:j}$ (Eq. 1). The latent space representations are then quantized as $q_{1:j}$ (Eq. 2) and randomly masked before being passed onto the multihead self-attention layers of the Transformer encoder backbone (Eq. 3). The model is optimized using a contrastive loss function by identifying a q from $q_{1:j}$ among a set of distractors q' given the masked context representation vector $h_{1:j}$ from the Transformer encoder.

$$z_{1:j} = \text{CNN}(x_{1:i}) \tag{1}$$

$$q_{1:j} = \text{quantization}(z_{1:j}) \tag{2}$$

$$h_{1:j} = \text{Transformer}(\text{masking}(z_{1:j})) \tag{3}$$

Next, we investigate a new SSL framework called WavLM [16] that instead uses masked region prediction to learn speech representations. WavLM does this through a denoising and prediction framework, mixing utterances by adding interfering speech and noise to the input speech, and making the model predict the original clean speech. Its architecture is similar to that of wav2vec 2.0, where the noisy input speech is passed onto a set of CNN layers to make $z_{1:j}$ and randomly masked before passing it onto the Transformer network, which then predicts the pseudo-labels of the masked regions $h_{1:j}$. The target pseudo-labels $k_{1:j}$ are generated by performing two iterations of k-means clustering: one on the MFCC of the original clean speech $x_{1:i}$ (Eq. 4) and another on the latent representations $z_{1:j}$ (Eq. 5). Moreover, a gated relative position bias is added to the Transformer backbone to better capture the sequence ordering of the noisy input speech. Training the model through this denoising framework proves to make the model robust to acoustic variations in the speech.

$$k_{1:j} = \text{k-means}(\text{MFCC}(x_{1:i})) \tag{4}$$

$$k_{1:j} = \text{k-means}(z_{1:j}) \tag{5}$$

2.2. Fine-tuning SSL models for ASR tasks

After learning the speech representations, the SSL model is then fine-tuned to predict the text tokens by attaching a decoder, making it similar to a regular end-to-end framework, as seen in the rightmost portion of Figure 1. Given an input speech sequence of $x_{1:i}$, the SSL model maps these into a hidden latent space as $h_{1:j}$ and is decoded into the text $y_{1:k}$. Another encoder can also be optionally added after the SSL model to further preprocess $h_{1:j}$ into $h'_{1:j}$. Frameworks such as the Connectionist Temporal Classification (CTC) [21] decode $h_{1:j}$ by predicting a text token for each frame. Other popular frameworks predict the text using attention (ATTN) with decoders such as the Recurrent Neural Network [22] or the Transformer [23]. Combining CTC and attention (CTC-A) [24, 25] through a weighted sum has also proven to be effective to decode the text.

3. Pretraining implementation of pathological speech recognition

In the following subsections, we describe how we implement the pretraining frameworks for this investigation. In general, we first pretrain a model on a large dataset to make it learn a general idea of speech and subsequently fine-tune it on the pathological dataset for adaptation.

3.1. Self-supervised pretraining setup

After pretraining the SSL model, we fine-tune it by attaching a decoder to its last layer. Two decoder setups for fine-tuning are compared. The first is using the original ASR setup in wav2vec 2.0 and WavLM, where a linear projection layer is attached and trained by CTC. Next, following the convention in [18], we train by CTC-A by using a vanilla 2-layer 1024-unit LSTM

decoder, as recommended in the ASR setup in [12]. As pathological speech is extremely acoustically different from healthy speech, we also investigate the effectiveness of adding an extra encoder to further pre-process the SSL features before passing it onto the decoder. Two additional encoders are compared with preprocessing the SSL representations, the Conformer [26] and Transformer [23], owing to the strong ability of self-attention to learn representations. We use two attention heads and two layer blocks for both encoders. We then refer to using the SSL representations directly as using an Identity encoder. We find better performance when only the CNN layers of the SSL models are frozen during fine-tuning. The rest of the SSL model components, the encoder, and the decoder are fine-tuned.

3.2. Supervised pretraining baseline setup

We compare SSL pretraining with the following supervised pretraining setups. We use the Transformer [23] and Conformer [26] as the encoders, both of which have produced the lowest word error rate when using the Librispeech [27] dataset in supervised training[1]. Both encoders have eight attention heads, although the Transformer has 18 layer blocks, while the Conformer only has 12 layer blocks. The decoder for both encoders is a Transformer composed of eight attention heads and eight layer blocks, and trained by CTC-A. The pretraining stage outputs and predicts byte-pair encoding (bpe) tokens [28]; however, in the fine-tuning stage, we initialize a new output layer to decode English and Japanese character tokens instead. For UASpeech, we also compare our results with those obtained using end-to-end models found in [29] that predict the character tokens by using a CONV+BLSTM encoder trained using CTC and ATTN with mel-scale filter banks and pitch as inputs.

4. Experimental setup

4.1. Datasets

To pretrain the supervised models, we use the 960 hours of Librispeech [27]. On the other hand, SSL pretraining varies per model, with wav2vec 2.0 pretrained on 60k hours of speech and WavLM pretrained on 94k hours of speech. In the fine-tuning stage, we use two pathological speech datasets, as summarized in Table 1. We first investigate an in-house recorded dataset of Japanese electrolaryngeal speech, which we refer to as ELSpeech in this paper. The dataset is recorded by a laryngeal speaker by speaking 1000 different sentence utterances both with an electrolarynx and with their normal voice. We use 85% of the data for the training set, and the rest for the test set. Both the healthy and electrolaryngeal speeches are used in the train and dev sets but only the electrolarygneal speech used in the test set. As the dataset is in Japanese, we use the character error rate (CER) as the evaluation metric.

The next dataset we use is UASpeech [30], a dataset containing parallel English word utterances of 15 dysarthric speakers and 13 healthy speakers. We follow the recommendations in [29] and use both the dysarthric and healthy speakers to train the ASR system but only use the dysarthric speakers for the test set, and remove excessive silences at the start and end of each utterance using a GMM-HMM system[2]. The train and dev sets are from the B1 and B3 blocks, whereas the test set is from the B2 block. We follow the convention of the said paper and use the word error rate (WER) as our evaluation metric.

[1] https://github.com/syhw/wer_are_we

[2] https://github.com/ffxiong/uaspeech

Table 1: *Duration (in number of hours) of each split used in the experiment. UASpeech results are obtained after silence removal.*

Dataset	Healthy		Pathological		Test
	Train	Dev	Train	Dev	
UASpeech [30]	2.38	0.45	3.60	0.68	2.09
ELSpeech	1.21	0.37	1.40	0.46	0.42

Table 2: *CER comparison between using different SSL features and raw features in ELSpeech. ** indicates end-to-end models pretrained with 960h Librispeech.*

Sys.	Features	Encoder	Decoder	Loss	CER%
1	Raw	**Conf.****	**Transf.****	**CTC-A**	**27.9**
2	waveform	Transf.**	Transf.**	CTC-A	61.5
3	wav2vec 2.0	Identity	Linear	CTC	93.3
4		Identity	LSTM	CTC-A	89.2
5		Conf.	LSTM	CTC-A	54.1
6		Transf.	LSTM	CTC-A	50.2
7	wav2vec 2.0 (XLSR)	Identity	Linear	CTC	99.6
8		Identity	LSTM	CTC-A	89.9
9		Conf.	LSTM	CTC-A	90.2
10		Transf.	LSTM	CTC-A	71.5
11	WavLM	Identity	Linear	CTC	54.9
12		**Identity**	**LSTM**	**CTC-A**	**41.8**
13		Conf.	LSTM	CTC-A	42.5
14		Transf.	LSTM	CTC-A	46.0

4.2. Implementation

We use ESPnet [31, 32] to implement the end-to-end ASR models and the s3prl frontend [12] to use the SSL models. Both toolkits are open-sourced. The SSL models we use are the wav2vec 2.0 Large and WavLM Large models, which are both available to the public. All networks in this investigation target predicting the character tokens. We fine-tune the networks' learning rate in steps of 10^{-n} where $n \in \{0, 1, 2, 3, 4, 5\}$ and use the best results from each run. The language models used are based on the Transformer [23] architecture and trained on each dataset's training set text. In Tables 2 and 3, we will refer to the Conformer architecture as "Conf." and the Transformer architecture as "Transf.".

5. Results

5.1. Comparison of SSL frameworks

First, we analyze the two SSL model frameworks. As seen in Sys. 12 in Table 2 and Sys. 10 in Table 3, using the WavLM model with an LSTM decoder outperforms all setups using wav2vec 2.0 in both datasets, as it produces the lowest CER (41.8%) and WER (51.8%), making it the best SSL setup. This performance can be attributed to the speech denoising framework used in masked region prediction, which makes the model more robust to acoustic variations in speech features found in pathological speech and the attention-based LSTM decoder, thereby improving the representation learning. A similar trend in both datasets is also seen when using an additional Conformer or Transformer encoder (Sys. 5, 6, 13, and 14 in Table 2 and Sys. 7, 8, 11, and 12 in Table 3) to preprocess the SSL features, where it degrades the performance with WavLM, but improves wav2vec 2.0. This shows that the wav2vec 2.0 features are not as strong in projecting the pathological speech into

Table 3: *WER comparison between using different SSL features, mel-scale filter bank, and raw waveforms in UASpeech. ** indicates models pretrained with 960h Librispeech.*

Sys.	Features	Encoder	Decoder	Loss	WER% of each intelligibility rating				
					Overall	Very Low	Low	Mid	High
1	Mel-scale filter bank	Conv+BLSTM [29]	Linear	CTC	58.0	87.1	62.4	55.3	37.8
2		**Conv+BLSTM**** [29]	**LSTM****	**ATTN**	**35.0**	**68.7**	**39.0**	**32.5**	**12.2**
3	Raw waveform	Conf.**	Transf.**	CTC-A	67.2	90.6	73.6	64.1	46.6
4		Transf.**	Transf.**	CTC-A	68.2	89.6	73.9	66.2	48.7
5	wav2vec 2.0	Identity	Linear	CTC	75.7	93.1	79.4	74.1	60.4
6		Identity	LSTM	CTC-A	96.4	97.9	97.5	95.8	95.0
7		Conf.	LSTM	CTC-A	71.7	92.3	74.6	68.7	55.3
8		Transf.	LSTM	CTC-A	71.8	93.8	83.7	69.7	48.4
9	WavLM	Identity	Linear	CTC	91.5	97.3	92.3	90.5	87.1
10		**Identity**	**LSTM**	**CTC-A**	**51.8**	**71.5**	**50.0**	**46.0**	**40.6**
11		Conf.	LSTM	CTC-A	70.1	94.1	81.3	67.1	48.5
12		Transf.	LSTM	CTC-A	77.2	94.5	83.9	75.3	60.3

a latent space and needs further processing for it to properly work. Moreover, as the SSL models are trained in English and the ELSpeech dataset is in Japanese, we also test the performance of the multilingually pretrained wav2vec XLSR model [15], which has been successfully used in decoding languages not included in the pretraining data. However, we do not find any success to this approach, as seen in Table 2, where we see a degradation in performance when using XLSR for all setups compared with its wav2vec 2.0 counterparts.

5.2. Comparison of SSL pretraining and supervised pretraining

We present two findings when comparing these two pretraining methods. First, we see a trend similar to that in [18], where the best SSL setups in Sys. 12 in Table 2 and Sys. 10 in Table 3 outperform the pretrained Transformer setups in Sys. 2 in Table 2 and Sys. 4 in Table 3, proving that SSL pretraining has the potential to outperform supervised pretraining.

However, our second finding is that other supervised setups not used in [18] may still outperform SSL pretraining. As seen Sys. 1 in Table 2, using a pretrained Conformer-Transformer model produces the lowest CER (27.9%) for ELSpeech, outperforming the best SSL setup by 13.9%. On the other hand, although the Conformer-Transformer setup (Sys. 3 in Table 3) did not perform very well in UASpeech, [29] showed that using a pretrained Conv+LSTM model with 40-dimensional mel-scale filter banks as inputs can still outperform our best SSL setup by 16.8%, as seen in Sys. 2 in Table 3.

5.3. Visualizing differences of dysarthric speech in the latent space

As seen in the results, SSL pretraining did not exactly outperform supervised pretraining in both pathological speech datasets. We attribute this to the stark acoustical differences in pathological speech, as similarly seen in previous research on automatically categorizing dysarthric speech intelligibility [33], where supervised models outperformed the pretrained SSL model. We then try to visualize the high-dimensional extractions using mel-log filterbank and pitch, raw waveforms, and SSL features with the UASpeech dataset and flatten these on a 2D space using t-SNE [34]. Each point represents one frame in the utterance and is colored according to the speakers' intelligibility rating. We use the first 20 utterances in the B1 block from three speakers in each intelligibility rating. We see in Figure 2 that raw audio waveforms are projected similarly to the SSL features, proving that there is space to further improve SSL models on pathological speech and improve them at the same level as supervised pretraining. Furthermore, more of the lower intelligibility utterances (green and red) are pushed to the outside, whereas more of the higher intelligibility utterances (purple and blue) are clustered at the center. However, we also see that the mel-scale filter bank extractions seem to blend together the pathological and healthy utterances. One hypothesis we draw from this is that the model may not be able to differentiate the pathological features from the healthy features when using this input, which may have improved the performance.

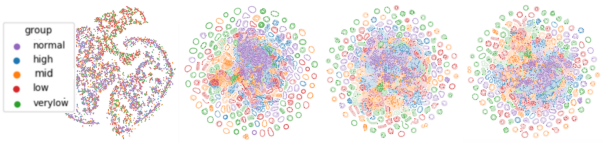

Figure 2: *Pathological speech frame extractions visualized in 2D via t-SNE [34]. Illustrated left to right are mel-scale filter banks with pitch, raw waveforms, WavLM, and wav2vec 2.0*

6. Conclusions

With ASR providing several conveniences to pathological speech patients, more work should be done to make commercial ASR devices useful for them. We investigated two types of SSL-based pretraining frameworks, contrastive and masked region prediction, on two different pathological speech datasets and compared them with a supervised pretraining framework. Although SSL pretraining frameworks have shown success with minimally resourced healthy speech, this does not seem to be the case with pathological speech. Further investigations in improving the training strategy can be conducted to improve performance in SSL pretraining. Training speaker-dependent models, which has been recommended in several studies [7, 8, 9], will be one of the primary targets in the future.

7. Acknowledgements

This work was partly supported by AMED under Grant Number JP21dk0310114, Japan, and a project, JPNP20006, commissioned by NEDO.

8. References

[1] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[2] H. A. Bourlard and N. Morgan, *Connectionist speech recognition*. Kluwer Academic Publishers, 1994.

[3] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[4] A. Graves and N. Jaitly, "Towards End-to-End Speech Recognition with Recurrent Neural Networks," in *Proc. ICML*, 2014, p. 1764–1772.

[5] R. D. Kent, "Research on Speech Motor Control and Its Disorders: A Review and Prospective," *Journal of Communication Disorders*, vol. 33, no. 5, pp. 391–428, 2000.

[6] M. Moore, H. Venkateswara, and S. Panchanathan, "Whistle-blowing ASRs: Evaluating the Need for More Inclusive Speech Recognition Systems," in *Proc. Interspeech*, 2018, pp. 466–470.

[7] J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, F. Vieira, M. McNally, T. Charbonneau, M. Nollstadt, A. Hassidim, and Y. Matias, "Personalizing ASR for Dysarthric and Accented Speech with Limited Data," in *Proc. Interspeech*, 2019, pp. 784–788.

[8] J. R. Green, R. L. MacDonald, P.-P. Jiang, J. Cattiau, R. Heywood, R. Cave, K. Seaver, M. A. Ladewig, J. Tobin, M. P. Brenner, P. C. Nelson, and K. Tomanek, "Automatic Speech Recognition of Disordered Speech: Personalized Models Outperforming Human Listeners on Short Phrases," in *Proc. Interspeech*, 2021, pp. 4778–4782.

[9] J. Tobin and K. Tomanek, "Personalized automatic speech recognition trained on small disordered speech datasets," *arXiv preprint arXiv:2110.04612*, 2021.

[10] S. Becker and G. E. Hinton, "Self-Organizing Neural Network that Discovers Surfaces in Random-Dot Stereograms," *Nature*, vol. 355, no. 6356, pp. 161–163, Jan 1992.

[11] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, and R. Shah, "Signature Verification Using a "siamese" Time Delay Neural Network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 07, no. 04, pp. 669–688, 1993.

[12] S.-W. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H.-Y. Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

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

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

[15] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2426–2430.

[16] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, and F. Wei, "WavLM: Large-Scale Self-Supervised Pre-training for Full Stack Speech Processing," *arXiv*, 2021.

[17] 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 *Proc. ICASSP*, 2020, pp. 6989–6993.

[18] L. Wu, D. Zong, S. Sun, and J. Zhao, "A Sequential Contrastive Learning Framework for Robust Dysarthric Speech Recognition," in *Proc. ICASSP*, 2021, pp. 7303–7307.

[19] K. D. N, P. Wang, and B. Bozza, "Using Large Self-Supervised Models for Low-Resource Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 2436–2440.

[20] R. Al-Ghezi, Y. Getman, A. Rouhe, R. Hildén, and M. Kurimo, "Self-Supervised End-to-End ASR for Low Resource L2 Swedish," in *Proc. Interspeech 2021*, 2021, pp. 1429–1433.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[22] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

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

[25] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM," in *Proc. Interspeech 2017*, 2017, pp. 949–953.

[26] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[28] P. Gage, "A new algorithm for data compression," *C Users J.*, vol. 12, no. 2, p. 23–38, Feb. 1994.

[29] S. Liu, M. Geng, S. Hu, X. Xie, M. Cui, J. Yu, X. Liu, and H. Meng, "Recent progress in the cuhk dysarthric speech recognition system," *IEEE/ACM TASLP*, vol. 29, pp. 2267–2281, 2021.

[30] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. S. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," in *Proc. Interspeech*, 2008, pp. 1741–1744.

[31] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[32] S. Watanabe, F. Boyer, X. Chang, P. Guo, T. Hayashi, Y. Higuchi, T. Hori, W.-C. Huang, H. Inaguma, N. Kamo, S. Karita, C. Li, J. Shi, A. S. Subramanian, and W. Zhang, "The 2020 ESPnet Update: New Features, Broadened Applications, Performance Improvements, and Future Plans," in *Proc. IEEE Data Science and Learning Workshop (DSLW)*, 2021, pp. 1–6.

[33] S. Venugopalan, J. Shor, M. Plakal, J. Tobin, K. Tomanek, J. R. Green, and M. P. Brenner, "Comparing Supervised Models and Learned Speech Representations for Classifying Intelligibility of Disordered Speech on Selected Phrases," in *Proc. Interspeech 2021*, 2021, pp. 4843–4847.

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, no. 86, pp. 2579–2605, 2008.