Enhancing Wav2Vec 2.0 Model Performance for Speech Representation

Progress Report

W.A.R.T. Fonseka - 210170G
Aug 23, 2025

Introduction

Neural networks are very beneficial for various tasks over traditional machine learning and statistical approaches and they dominate all major domains such as Natural Language Processing, Computer Vision, Speech and Audio Processing, and Time Series and Forecasting, but one of the major drawbacks of neural networks is that they are very data hungry, and in particular the unavailability of labeled data in domains like speech and audio processing became a serious problem in the last decades; however, after the rise of self-supervised learning researchers became eager to find models that produce context-aware representations of input data rather than building models to perform specific tasks, and with these pretrained models we can simply add a task-specific head on top so that the context-aware representations can be easily adapted to downstream tasks, bringing huge performance improvements across all domains especially in speech and audio processing, but compared to other domains, speech and audio processing still face difficulties such as variability in acoustic conditions, speaker differences, challenges in modeling temporal dependencies, limited labeled data, and the inherent ambiguity of sounds.

This research aims to improve Wav2vec 2.0, a state-of-the-art speech representation model, using different strategies. We specifically focus on lightweight enhancements that consume very low computational resources due to the limited resources available. In addition, the proposed methods are designed to improve robustness to speech variations, enhance the efficiency of learned representations, and reduce the reliance on heavy data augmentation, making the model more practical and adaptable for low-resource settings.

Literature Review

Wav2vec 2.0[1] is one of the most successful self-supervised, state-of-the-art models in the speech and audio processing domain, built based on the original Wav2Vec[2] and CPC[3] ideas. Wav2vec 2.0 has three main components: a feature encoder, a context network, and a quantization model. First, it converts raw audio data into vector representations using a 1D-CNN-based feature encoder, and then these encoded timesteps are fed into the context network, while the quantization model maps continuous latent representations to a discrete codebook, which is later used as targets for the context network. For the context network, a

Transformer model is used, and before feeding the feature encoder outputs into the Transformer, some timesteps are randomly masked to make the task more challenging. The context network output is then compared with the quantization model output in a contrastive manner. These design choices allow Wav2vec 2.0 to learn robust, context-aware representations of speech without labeled data, improve generalization across different speakers and acoustic conditions, and capture long-term dependencies effectively through the Transformer architecture while leveraging discrete codebooks to stabilize learning and provide meaningful contrastive targets.

With the great success of Wav2vec 2.0, researchers became eager to further improve and address some of its limitations by developing new models based on its architecture. One such model is HuBERT[4], which uses self-supervised learning to produce context-aware speech representations but differs from Wav2vec 2.0 in its approach to generating targets. HuBERT first generates pseudo-labels through clustering of acoustic features and then trains the model to predict these labels in an iterative manner. It uses the predicted labels from one stage are to refine the training targets in the next stage. This repeating process allows HuBERT to progressively capture both acoustic and linguistic structures more effectively.

Data2vec[5] is a multimodal architecture inspired by Wav2vec 2.0 that learns contextual representations not only from speech but also from text and images using a unified framework. WavLM[6] is an extended version of Wav2Vec 2.0 specialized for speech, which integrates masked speech prediction with additional training strategies to better capture speaker and acoustic information.

One of the main drawbacks of the above-mentioned models is that they apply CNN kernels for a fixed number of timesteps. This leads to the same speech being encoded differently by the feature encoder. The [7] paper explains the importance of speed perturbation empirically using data augmentation, suggesting that the original Wav2vec 2.0 model is not robust to speed variation of speech data but can benefit from data augmentation. Alternatively, we can use different CNN kernel sizes at the same stage, which also has some advantages over the augmentation method, such as larger kernels capturing long-range dependencies and smaller kernels capturing finer local details, leading to several benefits like invariance to small time-scale variations, reduced need for some augmentation, richer feature representations, and potentially better generalization. The [8] paper used similar methods but for sentence classification.

One of the other possible problems is that all codebooks may collapse into a single representation. Authors effectively handle the intra-codebook similarity problem using the diversity loss function, but not the inter-codebook similarity, which can lead to several inefficiencies. Ideally, these distinct codebooks are supposed to capture different aspects of the input, but due to the uncontrollable nature of codebooks, they may produce redundant quantization, meaning the same information is repeated and codebooks are wasted. This fails to improve the representation beyond what a single codebook could provide. To solve this issue, we can use several approaches such as modifying the loss function. The ERVQ[9] paper, introduced by Zheng et al., uses a similar approach to solve the inter-codebook similarity problem by modifying their loss function.

Methodology Outline

In this research, we initially plan to use two possible enhancements to improve the performance of the Wav2vec 2.0 model as mentioned below:

1. Use of different lengths of CNN kernels in parallel

Here we plan to use different lengths of CNN kernels in parallel, meaning at the same level, so that different speed speeches but with similar context are encoded more similarly in the feature encoder outputs. The primary objective of these additional CNN layers is to make a speed-perturbation-robust Wav2vec variant.

We plan to train the new model in two stages; first, we freeze all the original Wav2vec 2.0 parameters and train only the newly added CNN layers and the FFN, which maps CNN outputs to the original dimension size in the Wav2vec 2.0 context network. After that, we allow the model to update all parameters with a small learning rate and a few epochs to adapt to the new architecture scenario. We plan to train this way since we have very limited computational resources and should manage them effectively.

2. Update loss function to penalize inter-codebook similarity

Through this, we hope to address the inter-codebook similarity issue and mitigate possible inefficiencies in codebooks. By explicitly penalizing redundant quantization across codebooks, the model can encourage each codebook to capture distinct aspects of the input, leading to richer representations, better utilization of the available codebooks, and potentially improved generalization.

Datasets

Unlabled data set for pre training;

- LibriSpeech: A widely used English speech dataset derived from audiobooks.
- LibriVox (53.2k h):A large collection of audiobooks in English read by volunteers, offering around 53,000 hours of unlabeled speech for pretraining.

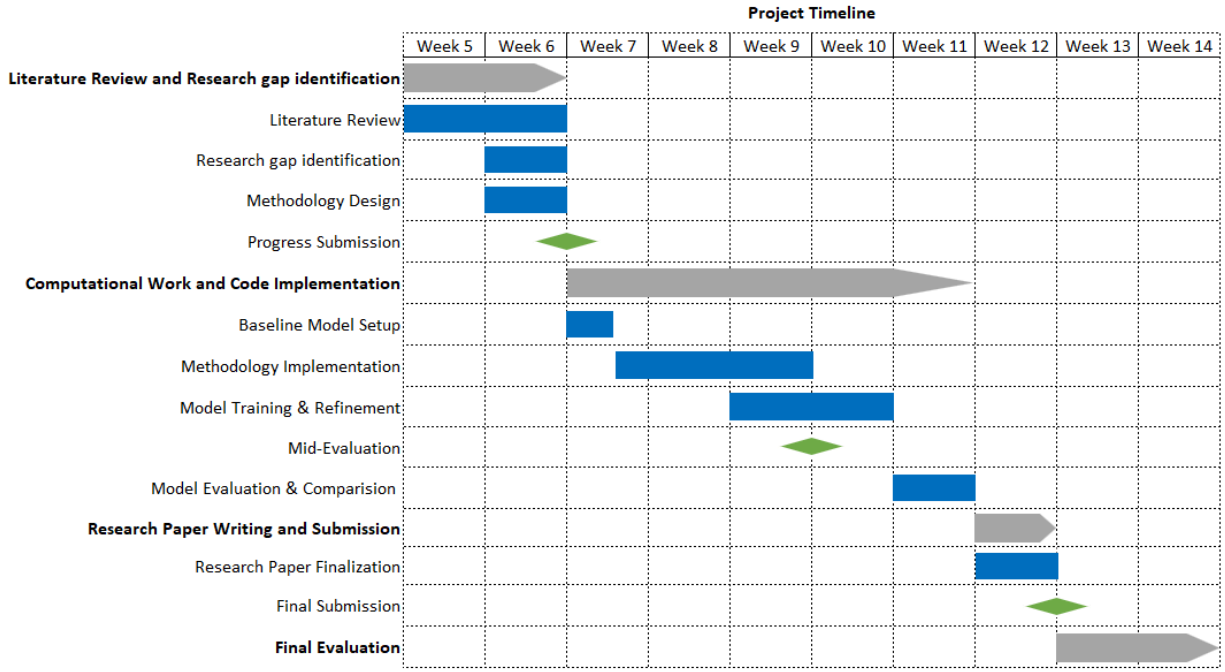
Labeled data set for fine tuning;

- LibriSpeech (1h, 10h, 100h, 960h): Labeled subsets of LibriSpeech with varying amounts of annotated data, commonly used for fine-tuning and benchmarking low-resource speech recognition.
- TIMIT: A phonetically rich dataset with time-aligned phonetic and word-level transcriptions, widely used for speech recognition and acoustic-phonetic studies.

Evaluation Metrics

- Word Error Rate (WER): The main evaluation metric for ASR.
- Phoneme Error Rate (PER): For phoneme recognition tasks on TIMIT and LibriSpeech phoneme subsets.

Project Timeline



References

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," Oct. 22, 2020, *arXiv*: arXiv:2006.11477. doi: 10.48550/arXiv.2006.11477.

[2] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," Sept. 11, 2019, *arXiv*: arXiv:1904.05862. doi: 10.48550/arXiv.1904.05862.

[3] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," Jan. 22, 2019, *arXiv*: arXiv:1807.03748. doi: 10.48550/arXiv.1807.03748.

[4] 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," June 14, 2021, *arXiv*: arXiv:2106.07447. doi: 10.48550/arXiv.2106.07447.

[5] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language," Oct. 25, 2022, *arXiv*: arXiv:2202.03555. doi: 10.48550/arXiv.2202.03555.

[6] S. Chen *et al.*, “WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing,” *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022, doi: 10.1109/JSTSP.2022.3188113.

[7] M. Huh, R. Ray, and C. Karnei, “A Comparison of Speech Data Augmentation Methods Using S3PRL Toolkit,” Mar. 29, 2024, *arXiv*: arXiv:2303.00510. doi: 10.48550/arXiv.2303.00510.

[8] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Sept. 03, 2014, *arXiv*: arXiv:1408.5882. doi: 10.48550/arXiv.1408.5882.

[9] R.-C. Zheng, H.-P. Du, X.-H. Jiang, Y. Ai, and Z.-H. Ling, “ERVQ: Enhanced Residual Vector Quantization with Intra-and-Inter-Codebook Optimization for Neural Audio Codecs,” June 11, 2025, *arXiv*: arXiv:2410.12359. doi: 10.48550/arXiv.2410.12359.