Task 3 Training Report

1. Introduction

This report compares the performance of the pretrained (*wav2vec2-large-960h*) and finetuned (*wav2vec2-large-960h-cv*) models on the *cv-valid-dev* dataset using Word Error Rate (WER) and Character Error Rate (CER). The goal is to analyse improvements from finetuning and propose further optimisations.

2. Model Comparison

The following metrics were computed for both model:

Model	WER	CER
Pretrained (*wav2vec2-large-960h*)	0.1176	0.0492
Finetuned (*wav2vec2-large-960h-cv*)	**0.0426**	**0.0182**

Observations:

- The finetuned model outperforms the pretrained model, reducing WER by ~64% and CER by 63%.
- This improvement confirms that finetuning on the Common Voice (CV) dataset enhances speech recognition accuracy for the target domain.

3. Proposed Improvements

While the finetuning is successful, further optimisations can be made in dataset preparation, model architecture, and hyperparameter tuning.

3.1 Dataset Improvements

(a) Full Dataset Utilisation

(i) Current Approach: The *cv-valid-train* dataset is further split into 70-30 training-validation sets.

(ii) Proposed Change: Use the entire training set for finetuning and evaluate on *cv-valid-dev* or *cv-valid-test*.

(iii) Benefits:

(1) More training data leads to better generalisation and robustness.

(b) Retain Long Audio Sequences

(i) Current Approach: Long audio sequences are removed to reduce memory consumption.

(ii) Proposed Change: Keep all sequences.

(iii) Benefits:

(1) Besides improving generalisation and robustness, more diverse training samples can lead to better handling of long-form speech.

(c) Silence Trimming

(i) Technique: Trim leading/trailing silence using dB thresholding (e.g., -30dB cutoff).

(1) Source: https://brentspell.com/blog/2022/gmm-trim/

(ii) Benefits:

(1) Improves alignment between audio and text tokens.

(2) Prevents the model from learning silence as a feature.

(d) Audio Augmentation

(i) Techniques:

(1) Noise/Reverb Injection (to simulate noisy and realistic environment)

a) Source: https://arxiv.org/html/2410.15609v1

(2) Speed Perturbation (increase or decrease audio speed to simulate fast and slow speaker)

a) Source: https://www.danielpovey.com/files/2015_interspeech_augmentation.pdf

(3) SpecAugment (performs frequency masking and time masking to the feature representations of the audio waveform)

a) Source: https://research.google/blog/specaugment-a-new-data-augmentation-method-for-automatic-speech-recognition/

(4) ROAR: Reinforcing Original to Augmented Data Ratio Dynamics for Wav2Vec2.0 Based ASR (dynamically balance clean vs augmented samples to prevent overfitting)

a) Source: https://arxiv.org/abs/2406.09999

(ii) Benefits:

(1) Improve robustness to real-world speech variations

3.2 Model Architecture Tweaks

(a) Unfreezing More Layers

(i) Current Approach: The feature extractor + first 6 encoder layers are frozen.

(ii) Proposed Change: Experiment with gradually unfreezing more layers (with lower learning rates).
    (1) Source: https://scispace.com/pdf/deep-transfer-learning-for-automatic-speech-recognition-202ykftf.pdf
(iii) Benefits:
    (1) Better adaptation to the target dataset.
    (2) Higher accuracy potential (but requires regularization to avoid overfitting).

(b) Adapter Layers / Low-Rank Adaptation (LoRA)

(i) Current Approach: Freeze layers to reduce computational cost.
(ii) Proposed Change: Add adapter layers or LoRA while freezing other layers.
    (1) Source: https://arxiv.org/pdf/2306.05617
(iii) Benefits:
    (1) Reduce computational cost, while keeping the finetuning efficient.

(c) LayerDrop Regularisation

(i) Current Approach: Does not have layer drop regularisation according to the pretrained model configuration
    (1) Source: https://huggingface.co/facebook/wav2vec2-large-960h/blob/6c9a7175e837a339d8c51851c3738d3b38640e4a/config.json
(ii) Proposed Change: Add layer drop to randomly skip a percentage of transformer layers during training.
(iii) Benefits:
    (1) Improves generalisation

3.3 Language Model (LM) Integration

(a) N-gram LM Fusion (KenLM)

(i) Technique: Wav2Vec2 predicts tokens independently, while an LM adds linguistic context.
    (1) Source: https://colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/Boosting_Wav2Vec2_with_n_grams_in_Transformers.ipynb#scrollTo=gUPAx3_MdyQv
(ii) Benefits:
    (1) Enables contextual error correction
        a) Eg. Wav2Vec2 might transcribe “I went too the store”, but KenLM corrects to “I went to the store”.

(2) Improves beam search by combining Wav2Vec2's acoustic confidence score and KenLM's language model scores.
   a) Eg.
      i) Without LM, the transcription may be "The quick brown fox jumps over the lazy dog." → "The kwik brown foks jumps over the lazy dog."
      ii) With LM, the misspellings ("kwik" and "foks") may be corrected based on N-gram probabilities.

3.4 Hyperparameter Tuning

(a) Differential Learning Rates

(i) Current Approach: Single learning rate for all layers.

(ii) Proposed Change: Lower learning rate for pretrained layers and higher learning rate (5e-4 to 1e-3) for the classification head.
   (1) Source: https://scispace.com/pdf/deep-transfer-learning-for-automatic-speech-recognition-202ykftf.pdf

(iii) Benefits:
   (1) Prevents catastrophic forgetting while allowing task adaptation.

(b) Dynamic Adjustment of Learning Rate

(i) Current Approach: Learning rate changes according to training step.

(ii) Proposed Change: Learning rate can be automatically reduced when validation loss plateaus
   (1) Source: https://medium.com/@zhonghong9998/adaptive-learning-rate-scheduling-optimizing-training-in-deep-networks-14d4f95a45d6

(iii) Benefits:
   (1) Allows better convergence and avoids suboptimal minima.