Task 4: Fine-tuning ASR Model Training Report

Lye En Lih

1 Introduction

This report compares the performance of my fine-tuned Wav2Vec2 model **(enlihhhhh/wav2vec2-large-960h-cv)** in *Task 3* against the baseline Wav2Vec2 model **(facebook/wav2vec2-base-960h)** in *Task 2a* The primary evaluation metric is **Word Error Rate (WER)**, which measures the percentage of words incorrectly predicted by the ASR model. By comparing the results, potential strategies will then be proposed to further improve the accuracy of the the fine-tuning results.

2 Word Error Rate (WER) Comparison

Model	Evaluation Dataset	Word Error Rate (WER)
`facebook/wav2vec2-base-960h`	cv-valid-dev	11.74%
`enlihhhhh/wav2vec2-large-960h-cv`	cv-valid-test	18.93%

Table 1: Comparison Results between fine-tuned and baseline Wav2Vec2 model

3 Proposed Strategy to improve fine-tuning results

3.1 Dataset Improvements

3.1.1 Increase the size of fine-tuning data

The current fine-tuned model only use **10% subset** of the whole Common Voice Train dataset after data filtering and pre-processing due to lack of computational resources. By providing more diverse training examples, the model will be able to learn better.

3.1.2 Train on Multiple Datasets

Common Voice by itself might be sufficient to allow the model to capture the diverse types of audio speeches, hence datasets like **TED-LIUM and TIMIT Acoustic-Phonetic Continuous Speech Corpus** can be used in conjunction with Common Voice to fine-tune the model for better results.

3.1.3 Audio Augmentation Techniques

As we are working with **Automatic Speech Recognition (ASR)**, we can augment the audio files by doing **noise injection, speed perturbation, pitch shifting, time stretching, etc.** These methods will help the ASR model to be more robust and perform better when the quality of audio files are poor.

3.2 Model and Training Optimisations

Even though I have done my own experiments of hyperparameter tuning, there are still ways to improve on it. **Optuna** can be employed to determine the best hyperparameters which results in minimising the CTC loss function. As it can run in parallel, it will work and the results from Optuna can directly be used with **HuggingFace's Trainer Function**. Furthermore, introducing a Language Model can improve transcription accuracy. **KenLM**, a statistical n-gram based language model can be used to refine ASR outputs by leveraging linguistic probabilities.

4 Conclusion

In conclusion, while my fine-tuning showed a degradation in the WER compared to the baseline model, being able to achieve a WER close to baseline model by fine-tuning on a smaller dataset shows that the fine-tuning methods adopted were effective in adapting the model to a constrained dataset while still maintaining a competitive level of performance.

This suggests that even with limited data, careful selection of fine-tuning techniques, hyperparameter tuning, and decoding strategies can significantly improve ASR model performance.