Introduction

This report evaluates the performance of the fine-tuned Wav2Vec2 model from Task 3, wav2vec2-large-960h-cv against the baseline results from Task 2a on the cv-valid-dev MP3 dataset. The objective is to analyze the model's performance, identify key issues, and propose improvements for better transcription accuracy.

Observations

After evaluating the fine-tuned model on the cv-valid-dev dataset, the following key observations were noted:

- All transcribed outputs were blank.
- The model did not produce meaningful text.
- No extensive testing was conducted by carrying out hyperparameter testing
- The performance of the baseline model was significantly better in comparison.

These observations indicate that the fine-tuning process negatively affected the model's ability to generate transcriptions, likely due to issues in training, data preprocessing, or model configuration.

Understanding of the problem

To understand the cause of the poor performance, several factors were considered:

1. **Data Preprocessing:** Potential issues in feature extraction for e.g normalization and removal of length of audio more than 5 seconds might have caused information loss.
2. **Tokenizer Alignment:** A mismatch between the tokenizer and model vocabulary could have led to invalid outputs.
3. **Training Instability:** If the learning rate was too high or the model overfitted, it may not have learned meaningful patterns.

Proposed Improvements

To improve the accuracy and reliability of the Wav2Vec2 model, the following steps are recommended:

Debug the Fine-Tuning Process

1. **Verify Data Preprocessing:** Ensure that audio files are correctly loaded, normalized, and transformed into suitable input features.
2. **Check Tokenizer Compatibility:** Confirm that the tokenizer used during inference matches the one used in training.

3. **Analyze Loss Curves:** Monitor training loss trends to detect overfitting or underfitting.
4. **Sanity Check Outputs:** Perform inference on a few audio samples manually to confirm whether the issue lies in data processing or model prediction.

Experiment with Training Strategies

1. **Reduce Learning Rate:** A smaller learning rate (e.g., 1e-5 or 1e-6) might improve model stability.
2. **Increase Training Epochs and Steps:** Extend training duration while monitoring validation loss.
3. **Test with different Optimizers:** Carry out training with different optimizers to evaluate which optimizer suits our use case best.

Improve Data Quality

1. **Augment Data:** Introduce techniques like time-stretching, pitch shifting, and background noise addition to enhance generalization.
2. **Filter Low-Quality Transcripts:** Remove poorly transcribed or noisy samples to avoid misleading the model.
3. **Feature Extraction:** Introduce different feature extraction techniques such as Log-Mel feature extraction

Evaluate and Iterate

1. **Compare with Task 2a Baseline:** If the baseline model performs well, identify differences in data processing and training setup.
2. **Perform Hyperparameter Tuning:** Optuna could be used to carry out Hyperparameter Tuning via GridSearch
3. **Train on a Smaller Subset First:** Fine-tune on a reduced dataset to estimate the performance of the model before using the full training dataset
4. **Log Intermediate Outputs:** Store and analyze intermediate representations to detect errors.

Conclusion

The fine-tuned model exhibited significant issues, producing blank transcriptions. This report outlined potential causes and provided a structured approach to diagnosing and improving the model. By refining preprocessing, adjusting training strategies, enhancing data quality, and iterating systematically, the model's accuracy can be significantly improved for the cv-valid-dev dataset.