Question 1 Report: Speech Enhancement

Anuj Rajan Lalla
Roll Number: B22AI061
CSL7770: Speech Understanding

Programming Assignment 2, Question 1

GitHub Repository: Click here

1 Introduction

- **Objective:** Enhance speech in multi-speaker scenarios by ensuring robust speaker verification.
- **Goals: (I have performed part I and II which was the speaker verification part)**
  - Download VoxCeleb1 (evaluation) and VoxCeleb2 (fine-tuning) datasets.
  - Evaluate a pre-trained speaker verification model (`wavlm-base-plus-sv` was chosen) on VoxCeleb1 using EER, TAR@1%FAR, and Speaker Identification Accuracy.
  - Fine-tune the model using LoRA and ArcFace loss on VoxCeleb2 (first 100 identities for training, remaining 18 for testing).
  - Compare the performance of the pre-trained and fine-tuned models.
- **Motivation:**
  - Reliable speaker verification is essential for effective speech enhancement in multi-speaker environments.
  - Fine-tuning aims to produce more discriminative speaker embeddings and reduce error rates.

2 Dataset Description

- **VoxCeleb1:**
  - **Usage:** Evaluation dataset.
  - **Content:** Cleaned trial pairs for speaker verification.
  - **Audio Format:** WAV files (resampled to 16 kHz as is required by the model from huggingface).
- **VoxCeleb2:**
  - **Usage:** Fine-tuning dataset.
  - **Content:** Audio files for a large set of speaker identities.
  - **Data Split:** First 100 identities (training) and remaining 18 identities (testing).
  - **Audio Format:** Primarily M4A files (converted/resampled to 16 kHz as needed).

3 Methodology

3.1 Pre-trained Model Evaluation

- **Data Preparation:**
  - Loaded the VoxCeleb1 (cleaned) dataset, including the trial pairs file and corresponding audio files from the `vox1` folder.
  - Resampled audio to a uniform 16 kHz to ensure consistency across samples.
- **Model Inference:**
  - Employed the pre-trained `wavlm-base-plus-sv` model from Hugging Face.
  - Extracted speaker embeddings from each audio sample in the trial pairs.
- **Performance Evaluation:**
  - Computed cosine similarity between embeddings to measure speaker similarity.
  - Calculated key metrics: Equal Error Rate (EER), True Acceptance Rate at 1% False Acceptance Rate (TAR@1%FAR), and Speaker Identification Accuracy.

- Generated visualizations such as ROC curves and similarity score histograms for further analysis.

3.2 Fine-Tuning with LoRA and ArcFace Loss

- **Dataset and Split:**
  - Utilized the VoxCeleb2 dataset, processing audio files (from the `vox2` folder) and associated metadata.
  - Sorted speaker identities in ascending order and designated the first 100 identities for training and the remaining 18 for testing.
- **Model Adaptation:**
  - Applied Low-Rank Adaptation (LoRA) to update select layers (e.g., `q_proj`, `k_proj`, `v_proj`) of the pre-trained model.
  - Integrated ArcFace loss to enhance the discriminative power of the speaker embeddings during fine-tuning.
- **Training and Evaluation:**
  - Fine-tuned the model over multiple epochs with appropriate training parameters (learning rate, optimizer settings, etc.).
  - Evaluated the fine-tuned model on the VoxCeleb1 trial pairs using the same performance metrics (EER, TAR@1%FAR, and Speaker Identification Accuracy).
  - Compared the performance of the fine-tuned model against the baseline pre-trained model, analyzing improvements and potential trade-offs.

4 Results and Discussion

4.1 Quantitative Performance

- **Pre-trained Model:**
  - Equal Error Rate (EER): 5.23%
  - TAR@1%FAR: 74.45%
  - Speaker Identification Accuracy: 94.77%
- **Fine-tuned Model:**
  - Equal Error Rate (EER): 4.94%

– TAR@1%FAR: 78.29%
– Speaker Identification Accuracy: 95.06%

4.2 Training Details and Procedure

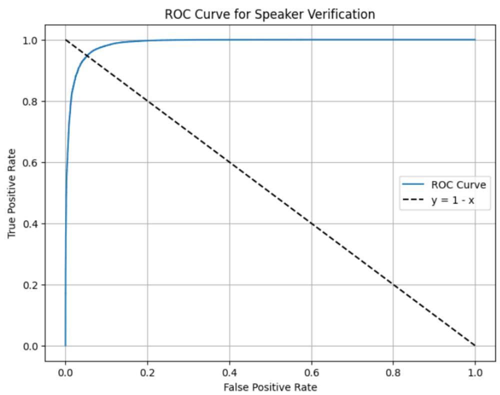
- Fine-tuning was performed on the VoxCeleb2 dataset over 3 epochs.
- Each epoch consisted of 65 iterations.
- A steady decrease in training loss was observed, indicating effective convergence.

Epoch	Iterations	Average Loss	Duration (mm:ss)
1/3	65	11.2873	14:14
2/3	65	10.5755	13:24
3/3	65	10.1388	13:18

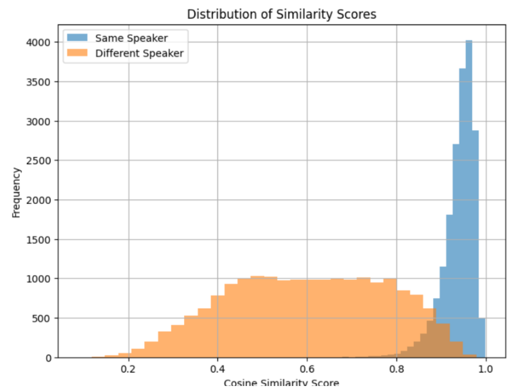
Table 1: Training performance over 3 epochs during fine-tuning with LoRA and ArcFace loss.

4.3 Visualizations and Analysis

- **Pre-trained Model Visualizations:**
  – **Figure 1a**: ROC curve for the pre-trained model, illustrating the trade-off between false acceptances and true acceptances.
  – **Figure 1b**: Histogram of cosine similarity scores for same and different speaker pairs.
  – These figures indicate that the pre-trained model achieves an EER of 5.23% and a TAR@1%FAR of 74.45%.
- **Fine-tuned Model Visualizations:**
  – **Figure 2a**: ROC curve for the fine-tuned model, showing improved separation between genuine and impostor scores.
  – **Figure 2b**: Histogram of cosine similarity scores for the fine-tuned model.
  – The fine-tuned model exhibits enhanced performance with an EER of 4.94% and a TAR@1%FAR of 78.29%.

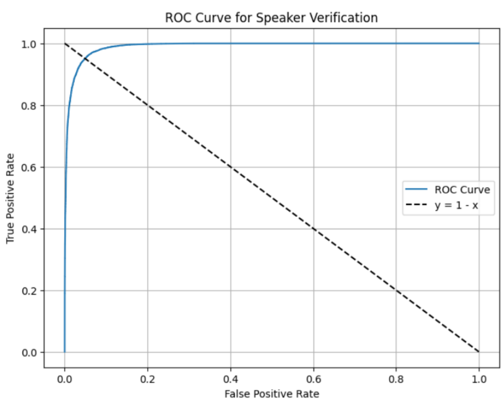


(a) ROC curve for the pre-trained model.

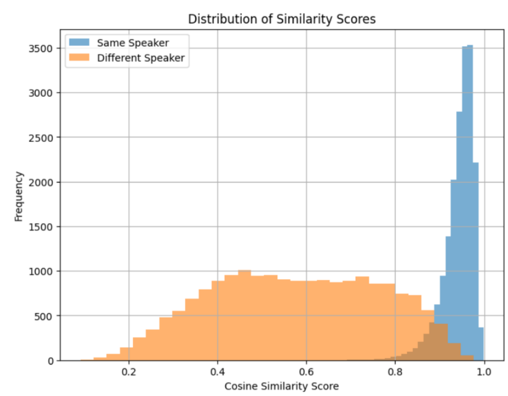


(b) Histogram of cosine similarity scores for the pre-trained model.

Figure 1: Visualization of the pre-trained model's performance.



(a) ROC curve for the fine-tuned model.



(b) Histogram of cosine similarity scores for the fine-tuned model.

Figure 2: Visualization of the fine-tuned model's performance.

4.4 Discussion and Observations

- The reduction in EER from 5.23% to 4.94% suggests an improved balance between false acceptances and rejections.
- The increase in TAR@1%FAR from 74.45% to 78.29% indicates that the fine-tuned model is more robust under low false acceptance conditions.
- The slight improvement in Speaker Identification Accuracy (from 94.77% to 95.06%) further validates the effectiveness of the fine-tuning strategy.

- The training loss consistently decreased over the epochs, confirming effective model convergence.
- The ROC curves and histograms show that, after fine-tuning, there is a clearer separation in the score distributions. Specifically, the histogram for the fine-tuned model shows a notable reduction in the frequency of high cosine similarity scores for different-speaker pairs, which indicates that the model is better at distinguishing between speakers. (Eg for score 0.8 for different speakers frequency goes down from 1000 in pre-trained to somewhere around 800 for finetuned)

5 Conclusion

- The experiments demonstrate that fine-tuning the pre-trained `wavlm-base-plus-sv` model using LoRA and ArcFace loss improves speaker verification performance.
- Key metrics improved, with the EER reducing from 5.23% to 4.94%, TAR@1%FAR increasing from 74.45% to 78.29%, and Speaker Identification Accuracy slightly rising from 94.77% to 95.06%.
- The analysis of ROC curves and similarity score histograms confirms a better separation between same- and different-speaker pairs after fine-tuning.
- The decrease in the frequency of high similarity scores for different-speaker pairs further validates the improved discriminative capability of the fine-tuned model.

References

[1] Huggingface transformers. `https://github.com/huggingface/transformers`. Library for state-of-the-art Natural Language Processing models, used here for model loading and inference.

[2] Peft: Parameter-efficient fine-tuning. `https://github.com/huggingface/peft`. Used for Low-Rank Adaptation (LoRA) during model fine-tuning.

[3] pydub: Manipulate audio with a simple and easy high-level interface. https://github.com/jiaaro/pydub. Used for processing M4A audio files.

[4] Pysoundfile. https://github.com/bastibe/PySoundFile. Used for reading and writing sound files.

[5] Python libraries for audio and machine learning. Librosa, Joblib, Matplotlib, tqdm, NumPy, Scikit-learn, and PyTorch. These libraries were used for audio processing, feature extraction, visualization, and model building. For more details, refer to their official websites: https://librosa.org, https://joblib.readthedocs.io, https://matplotlib.org, https://github.com/tqdm/tqdm, https://numpy.org, https://scikit-learn.org, and https://pytorch.org.

[6] Unispeech: Downstreams for speaker verification. https://github.com/microsoft/UniSpeech/tree/main/downstreams/speaker_verification. GitHub repository for the speaker verification implementation.

[7] Voxceleb dataset. https://mm.kaist.ac.kr/datasets/voxceleb/. Dataset for speaker verification.

[8] Wavlm base plus sv. https://huggingface.co/microsoft/wavlm-base-plus-sv. Pre-trained speaker verification model from Microsoft.