# Speaker Separation and Verification

Neermita Bhattacharya

Indian Institute of Technology, Jodhpur, Jodhpur 342037, India
b22cs092@iitj.ac.in

## Abstract

Speaker verification, separation and speech enhancement are critical tasks in audio processing, enabling applications such as voice authentication, speaker diarization, and noise-robust speech recognition. Speaker verification identifies individuals based on voice characteristics, while speech enhancement improves audio quality in noisy or multi-speaker environments from separated audio. Recent advances in self-supervised learning (e.g., Wav2Vec2, HuBERT) have significantly improved speaker verification, while transformer-based models (e.g., SepFormer) have enhanced speech separation capabilities.

For section (I), two datasets, VoxCeleb1 and VoxCeleb2 were downloaded and extracted. The pre-trained model of unispeech-sat-base-plus-sv (available here), was utilized for speaker verification in section (II). This model can be used to derive meaningful embeddings for speaker identification/verification tasks. A list of trial pairs - VoxCeleb1 (5000) was utilized to study the results of this pre-trained model. Speaker Verification Accuracy of 95.50%, Equal Error Rate (EER) of 4.12% and TAR@1%FAR of 79.45% were achieved by pre-trained unispeech. The goal was to find whether finetuning the model further on the VoxCeleb2 dataset (100 for training, 18 for tetsing) using Low-Rank Adaptation (LoRA) and ArcFace loss could help increase the accuracy. The results indicate a higher performance. Speaker Verification Accuracy of 98.33%, an Equal Error Rate (EER) of 1.78%, and a TAR@1%FAR of 95.78% are observed after finetuning.

Section (III) part (A) mainly focused on speaker separation using a pretrained model, Sepformer (available here). A function was created to automate the mixing of different utterances from the VoxCeleb2 dataset. The first 50 speakers were utilized to construct the training set and the next 50 for the test set. A multi-speaker dataset is then created consisting of 500 training samples (random pairs from the 50 speakers in train) and 100 testing samples (random pairs from the 50 speakers in test). This simulates real-world overlapping speech. The SepFormer model is then applied to separate and enhance individual speakers, with quality assessed via metrics averaged over all speakers such as Signal-to-Distortion Ratio (4.92), Signal to Interference Ratio (20.25), Signal to Artefacts Ratio (5.46), and Perceptual Evaluation of Speech Quality (1.61).

Part (B) of this section dealt with using the finetuned speaker verification model to identify which separated audio corresponded to which original speaker. The overall Rank-1 identification accuracy among all the speakers in the test set was 88.0%.

Finally, section (III) required us to create a novel method utilizing both concepts of Speaker Verification and Separation in a single pipeline. An overall Rank-1 accuracy of 91% for speaker verification from the separated audios was achieved. All code files and reports are available at github.com/Neermita18/Speech-Understanding-PA2.

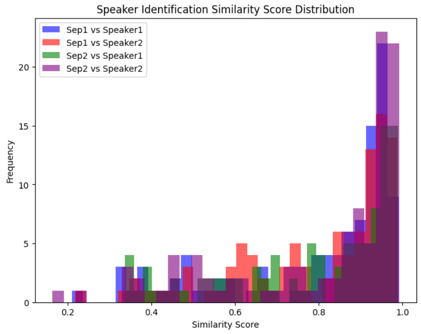


Figure 1: Speaker Identification Similarity Score Distribution

# Contents

# 1 Question 1

## 1.1 Introduction

Speaker verification identifies whether a speech sample matches a claimed speaker using voice characteristics like pitch and tone. It is used in authentication and security, with models like Wav2Vec2, Unispeech, Hubert improving accuracy. Speaker separation extracts individual voices from mixed audio, useful for transcription and meetings. Deep learning models like SepFormer enhance separation, especially in overlapping speech or noisy environments. Unispeech-sat-base-plus-sv [3] is a self-supervised learning model fine-tuned for speaker verification, leveraging phonetic and linguistic supervision (PLS) to enhance speaker discrimination. SepFormer [2] is a transformer-based speech separation model designed for blind source separation, using a dual-path transformer architecture to extract individual speakers from a mixed audio signal. It is typically optimized with SI-SDR loss for high-quality separation.

### 1.1.1 Speaker Verification

Speaker verification determines if a speech sample belongs to a claimed speaker. It is widely used in biometric authentication, security, and virtual assistants. The process extracts speaker-specific features and compares them to stored references.

Verification can be text-dependent, requiring a fixed phrase, or text-independent, allowing any speech sample. Text-independent systems are more flexible but face greater variability.

Several factors affect accuracy, including acoustic features like pitch and tone. Background noise and recording conditions can distort speech, making verification harder.

Channel variability from different microphones or telephony systems can cause mismatches, reducing performance. Speaker variability, such as changes due to aging or emotions, also impacts accuracy.

Deep learning models like Wav2Vec2 and HuBERT improve verification by learning robust speech representations from large datasets. Diverse training data further enhances generalization.

### 1.1.2 Speaker Separation

Speaker separation isolates individual voices from multi-speaker audio. It is crucial for transcription, voice assistants, and meetings where multiple people speak simultaneously.

The number of overlapping speakers affects separation difficulty. More voices increase complexity, especially with long-duration overlaps. Speaker similarity also plays a role—similar voices are harder to distinguish.

Background noise and reverberation can degrade speech, making it difficult for models to extract clean signals. Training on noisy data helps models generalize better.

Advanced models like SepFormer use transformers to capture long-term dependencies, improving separation. Speaker embeddings from verification models can further guide separation for better results.

Diverse and realistic training data improve model robustness, ensuring clear speech extraction in real-world environments.

### 1.1.3 Low-Rank Adaptation (LoRA)

Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning technique for large neural networks. Instead of updating all model weights, LoRA introduces trainable low-rank matrices to approximate weight updates, significantly reducing the number of parameters to be trained.

Given a weight matrix $W_0$, LoRA models the update as:

$$W = W_0 + \Delta W, \quad \text{where} \quad \Delta W = AB$$

where:

- $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ are learnable low-rank matrices.
- $r$ is much smaller than $d$, reducing the number of trainable parameters.

In the code, LoRA is applied to the attention layers (`q_proj`, `k_proj`, `v_proj`), the output projection, and dense layers. The rank $r = 8$ and the scaling factor $\alpha = 16$ control the expressiveness of the adaptation. A dropout of 0.1 is used to prevent overfitting. These settings allow efficient fine-tuning while keeping most of the original transformer model frozen.

### 1.1.4 ArcFace Loss

ArcFace loss is an additive angular margin loss used in classification tasks to enhance feature discrimination. It was studied upon [1] here. It modifies the standard softmax loss by adding an angular margin $m$ to the target class logits:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j\neq y_i}e^{s\cos\theta_j}}$$

where:

- $s$ is a scaling factor for numerical stability.
- $\theta_{y_i}$ is the angle between the feature vector and class weight.
- $m$ is the margin added to enforce greater separation between classes.

In the code, the weight matrix is initialized using Xavier uniform initialization, and cosine similarities are computed after normalizing both inputs and weights. The margin $m = 0.4$ and scaling $s = 30.0$ enforce stronger feature discrimination. The logits are then adjusted accordingly before being passed to a cross-entropy loss function.

## 1.2 Section I: Dataset Collection

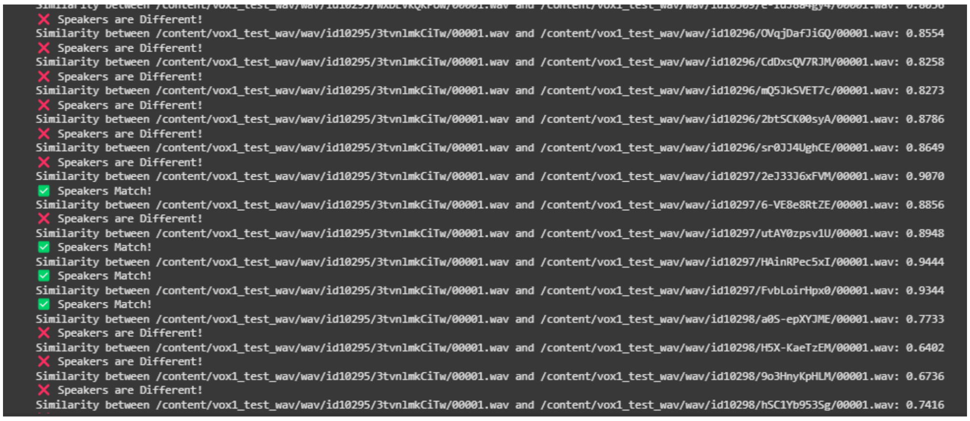
VoxCeleb1 and VoxCeleb2 were used for training and testing for all tasks.

- **Dataset**: VoxCeleb1 (evaluation), VoxCeleb2 (fine-tuning)
- **Model**: UniSpeech-SAT + LoRA (r=8) + ArcFace Loss
- **Training**: First 100 identities from VoxCeleb2
- **Evaluation**: VoxCeleb1 trial pairs (5000 pairs used), or VoxCeleb2 last 18 speakers for Section (II), or last 50 speakers for Section (III).

## 1.3 Section II: Speaker Verification/Identification using LoRA and ArcFace Loss

The list of pairs from VoxCeleb1 was utilized as the testing file. It consisted of the score (0/1) depending on whether the speakers were the same or not, and the files of the two speakers. In the beginning, the VoxCeleb1 dataset was used to verify the accuracy of the unispeech pre-trained speaker verification model. As expected, for a threshold of 0.89 on the cosine similarities obtained from the embedding pairs, the performance is shown to be legitimate.

Next, the list of pairs VoxCeleb1 was used (5000 pairs) to test the performances of both the pre-trained as well as the finetuned unispeech model.



```
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10296/OVqjDafJiGQ/00001.wav: 0.8554
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10296/CdDxsQV7RJM/00001.wav: 0.8258
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10296/mQ5JkSVET7c/00001.wav: 0.8273
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10296/2btSCK00syA/00001.wav: 0.8786
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10296/sr0JJ4UghCE/00001.wav: 0.8649
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10297/2eJ33J6xFVM/00001.wav: 0.9070
✅ Speakers Match!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10297/6-VE8e8RtZE/00001.wav: 0.8856
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10297/utAY0zpsv1U/00001.wav: 0.8948
✅ Speakers Match!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10297/HAinRPec5xI/00001.wav: 0.9444
✅ Speakers Match!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10297/FvbLoirHpx0/00001.wav: 0.9344
✅ Speakers Match!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10298/a0S-epXYJME/00001.wav: 0.7733
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10298/H5X-KaeTzEM/00001.wav: 0.6402
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10298/9o3HnyKpHLM/00001.wav: 0.6736
❌ Speakers are Different!
Similarity between /content/vox1_test_wav/wav/id10295/3tvnlmkCiTw/00001.wav and /content/vox1_test_wav/wav/id10298/hSC1Yb953Sg/00001.wav: 0.7416
```

Figure 2: Speaker Verification Results for Threshold = 0.89

### 1.3.1 Equal Error Rate (EER)

- The EER is the point where the False Acceptance Rate (FAR) and False Rejection Rate (FRR) are equal.
- It provides a single value to compare different verification systems: lower EER means better performance.
- The EER is found by plotting the Receiver Operating Characteristic (ROC) curve and identifying the intersection of FAR and FRR.

### 1.3.2 True Acceptance Rate (TAR) at 1% False Acceptance Rate (FAR)

- TAR (True Acceptance Rate) is the percentage of correctly accepted genuine users.
- FAR (False Acceptance Rate) is the percentage of impostors incorrectly accepted.
- ”TAR at 1% FAR” means measuring how well the system correctly identifies genuine users while allowing only 1% of impostors to be incorrectly accepted.
- A higher TAR at 1% FAR indicates a more reliable system.

### 1.3.3 Identification Accuracy

- Used in closed-set speaker identification, where a system chooses the correct identity from a set of known speakers.
- Accuracy = Correct Identifications / Total Attempts ×100
- Unlike verification, identification does not require a threshold; it assigns the most likely identity based on feature matching.

### 1.3.4 Results

Since the list of pairs from VoxCeleb1 (cleaned) contained (0/1) labels, embeddings of the pair of audio samples were retrieved from the pre-trained/finetuned unispeech-sat model, the cosine similarity was estimated, and a threshold of 0.85 was used to check whether the samples were from the same speaker or not. This enabled us to do a 0/1 classification and report the Speaker Identification Accuracy.

- **Pre-trained Model**:
  - EER: 4.12%
  - TAR@1%FAR: 79.45%
  - Identification Accuracy: 95.50%
- **Fine-tuned Model**:
  - EER: **1.78%**
  - TAR@1%FAR: **95.78%**
  - Identification Accuracy: **98.33%**

**Observations:**

- LoRA adaptation significantly reduced EER while maintaining parameter efficiency.
- ArcFace loss improved discriminability of similar voices.

## 1.4 Section III: Speaker Separation

Speaker Separation is employed in problem settings more commonly known as cocktail-party problems, where multiple speakers and background noise might be present, and the task is to separate out clear and enhanced voices of the original speakers.

### 1.4.1 Creating a Mixed Dataset

- Mix Audio (mix_audio) – Loads two audio files, normalizes, adjusts SNR, mixes them, and saves the output.
- Convert .m4a to .wav (convert_m4a_to__wav) – Converts .m4a files to .wav using torchaudio.
- Get Random Audio (get_random_audio) – Selects a random audio file from a given speaker's folder.
- Prepare Speaker Data – Splits speakers into train (50) and test (50) groups.
- Generate Training Data (500 pairs) – Randomly selects two speakers, mixes their audio, and stores metadata.
- Generate Testing Data (100 pairs) – Similar to training, but with the test speakers.
- Finally, we store training and testing metadata in train_metadata.csv and test_metadata.csv. They are stored in the format: 'speaker1_id', 'speaker2_id', 'mixed_path', 'source1_path', 'source2_path']

### 1.4.2 Speaker Separation using pretrained Sepformer

The Sepformer model was loaded from speechbrain.inference.separation. Specific instructions were given on HuggingFace which were utilized to run inference using the model. Evaluation Metrics for Separation are discussed below:

**Signal-to-Distortion Ratio (SDR)**
The Signal-to-Distortion Ratio (SDR) measures the overall quality of the separated signal. A higher SDR indicates better separation with less distortion. It is defined as:

$$SDR = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{noise}} + e_{\text{interf}} + e_{\text{artif}}\|^2}$$

where:

- $s_{\text{target}}$ is the true source signal.
- $e_{\text{noise}}$, $e_{\text{interf}}$, and $e_{\text{artif}}$ are the noise, interference, and artifacts errors, respectively.

**Signal-to-Interference Ratio (SIR)**
The Signal-to-Interference Ratio (SIR) quantifies how well the model removes interference from other sources. A higher SIR means better suppression of interfering signals. It is given by:

$$SIR = 10\log_{10}\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}}\|^2}$$

where:

- $s_{\text{target}}$ is the true source signal.
- $e_{\text{interf}}$ is the interference from other sources.

**Signal-to-Artifacts Ratio (SAR)**
The Signal-to-Artifacts Ratio (SAR) evaluates the distortion introduced by the separation model. A higher SAR value means fewer artifacts in the separated signal. It is calculated as:

$$SAR = 10\log_{10}\frac{\|s_{\text{target}} + e_{\text{interf}}\|^2}{\|e_{\text{artif}}\|^2}$$

where:

- $e_{\text{artif}}$ represents artifacts introduced by the model.

**Perceptual Evaluation of Speech Quality (PESQ)**
The Perceptual Evaluation of Speech Quality (PESQ) metric measures the perceptual quality of the separated signal compared to the original clean speech. It produces a score in the range of -0.5 to 4.5, where higher values indicate better perceptual quality. The PESQ score is computed using a complex perceptual model, approximated as:

$$PESQ = a_1 \cdot \text{distortion} + a_2 \cdot \text{perceived quality}$$

where:

- $a_1$ and $a_2$ are weighting factors.

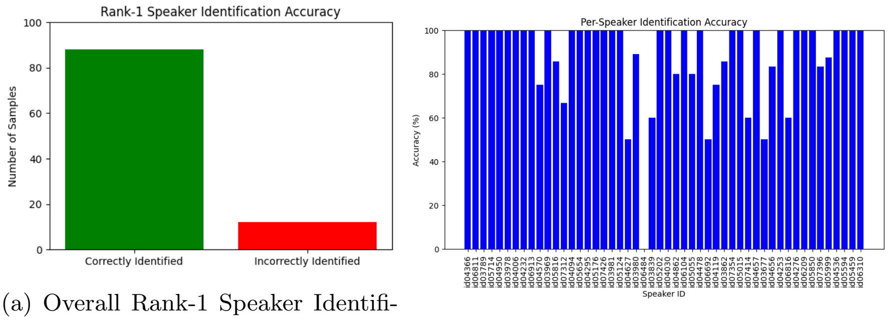
**Methods used for Speaker Separation:**

- evaluate_separation(): Ensures reference and estimated sources have the same length. Computes SDR (Signal to Distortion Ratio), SIR (Signal to Interference Ratio), SAR (Signal to Artifacts Ratio) using bss_eval_sources.
- calculate_pesq(): Ensures input audio is in the valid range and format. Computes PESQ (Perceptual Evaluation of Speech Quality) for reference and processed audio.
- trim_or_pad_audio(): Trims or pads audio to match the reference length.
- Processing Test Files: Iterates over test metadata. Loads mixed, original, and speaker data. Resamples mixed audio to 8000 Hz. Saves temporary mixed file for model input.
- Speech Separation: Uses a pre-trained model to separate sources. Saves separated speech files in separated_audio folder.
- Evaluation of Separation: Loads and resamples both original and separated sources. Adjusts lengths using trim_or_pad_audio(). Computes SDR, SIR, SAR using evaluate_separation() and PESQ for each source using calculate_pesq().
- All results saved in separation_evaluation_results.csv. We also group results by speaker to analyze performance differences.

### 1.4.3 Speaker Identification on Separated Audio

The performance of the separated audios by the Sepformer model can be assessed by evaluating the cosine similarities between the original pairs and the separated pairs. Further, speaker identification/verification could also be done based on the similarities found.

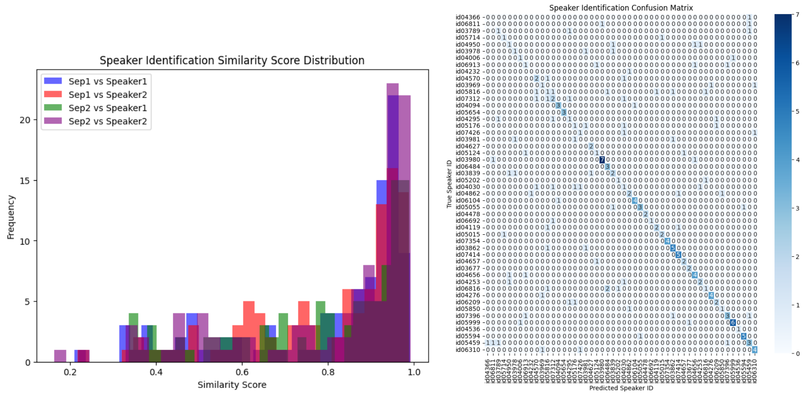
- For this task, the finetuned speaker verification model, unispeech-sat, was used during inference. The performances of both the simple pre-trained unispeech model and the finetuned unispeech-sat are reported in Section 1.5.
- Speaker embedding extraction was done. We extracted speaker embeddings for both original and separated speech sources. Uses extract_embedding(source_path) for each file.
- Cosine Similarity Computation: Computes similarity between original and separated speaker embeddings using cosine similarity. Forms a similarity matrix to determine the best matches.
- Speaker Assignment: Assigns each separated source to the closest original speaker based on similarity scores. Determines correctness by checking if speaker assignments match expected pairs.
- Evaluation Process: Iterates through test_metadata.csv, which contains mixed and original speaker paths. Separates mixed speech and matches identified speakers to ground truth labels.
- Accuracy Calculation: Rank-1 accuracy was calculated as the percentage of correctly identified speaker pairs. An overall accuracy of 88% was achieved.
- Per-speaker accuracy: Computes identification accuracy for each speaker separately.



(a) Overall Rank-1 Speaker Identification Accuracy

(b) Per-Speaker Identification Accuracy

Figure 3: Identification Accuracies



(a) Similarity Scores in Speaker Identification

(b) Confusion Matrix of Speaker Identification

Figure 4: Confusion Matrix and Similarity Score Distribution in Speaker Identification

## 1.5 Section IV: A Novel Pipeline Combining Speaker Separation and Verification

### 1.5.1 Outline

The proposed pipeline integrates speaker separation (SepFormer) and speaker verification (UniSpeech-SAT). SepFormer extracts individual speakers from mixed audio. UniSpeech-SAT verifies speaker identity. Each model is fine-tuned differently due to their distinct objectives.

### 1.5.2 Finetuning Sepformer for Speaker Separation

- Optimize **separation quality** while preserving **speaker identity**.
- Ensure minimal distortion and high signal clarity.

**Loss Functions**:

**Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) Loss**: Measures how well the predicted signal matches the target while being invariant to scale.

$$\mathcal{L}_{\text{SI-SDR}} = -10 \log_{10} \frac{\|\alpha s_{\text{target}}\|^2}{\|\hat{s} - \alpha s_{\text{target}}\|^2} \tag{1}$$

where:

- $s_{\text{target}}$ is the clean reference speech.
- $\hat{s}$ is the separated speech.

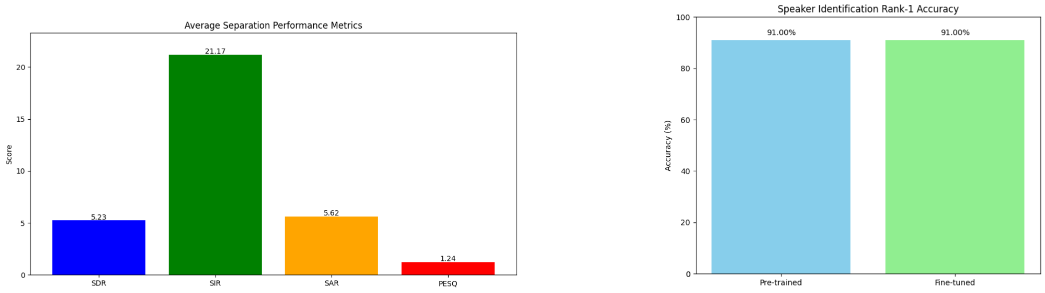
- $\alpha$ is a scale factor.

**Cosine Similarity Loss**: Ensures separated speech embeddings remain close to true speaker embeddings.

$$\mathcal{L}_{\text{cosine}} = 1 - \frac{\langle \mathbf{E}_{\text{true}}, \mathbf{E}_{\text{pred}} \rangle}{\|\mathbf{E}_{\text{true}}\| \|\mathbf{E}_{\text{pred}}\|} \tag{2}$$

where:

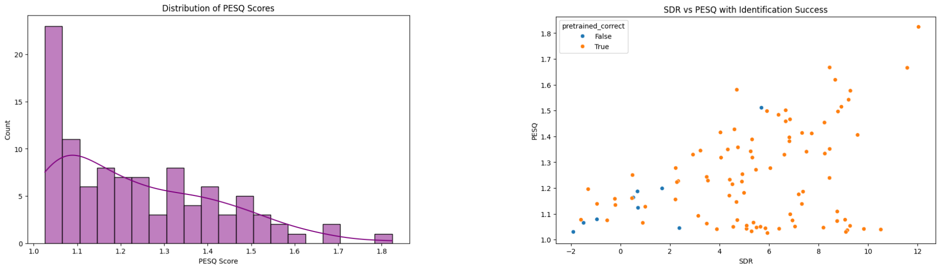
- $\mathbf{E}_{\text{true}}$ and $\mathbf{E}_{\text{pred}}$ are embeddings of true and predicted speech.



(a) Separation metrics

(b) Rank-1 Speaker Identification Accuracy

Figure 5: Separation metrics and Identification accuracies



(a) Overall PESQ scores

(b) PESQ vs SDR plot

Figure 6: PESQ scores

### 1.5.3 Finetuning Unispeech-sat for Speaker Verification/Identification

As mentioned above, Unispeech-sat can be finetuned with LoRA layers and ArcFace loss.

- **Low-Rank Adaptation (LoRA)** introduces trainable low-rank matrices. Allows **efficient fine-tuning** while keeping most pre-trained weights frozen.
- **ArcFace Loss** enhances speaker separability in embedding space.

### 1.5.4 Final Pipeline Setup

- **Pretrained SepFormer** → Separates speakers from mixed speech.
- **Fine-Tuned UniSpeech-SAT** → Verifies speaker identity.

### 1.5.5 Pipeline Flow

1. **Input:** Mixed speech audio.
2. **Step 1:** SepFormer separates individual speakers.
3. **Step 2:** UniSpeech-SAT verifies speaker identity.
4. **Output:** Separated and verified speakers.

### 1.5.6 Conclusion

- SepFormer was **not fine-tuned** due to training challenges.
- UniSpeech-SAT was **fine-tuned with LoRA & ArcFace loss** for better verification.
- An impressive overall accuracy of 91% was achieved by both the pre-trained and finetuned Unispeech-sat + Sepformer pipeline.

## References

[1] Jiankang Deng, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):5962–5979, October 2022. ISSN 1939-3539. doi:10.1109/tpami.2021.3087709. URL http://dx.doi.org/10.1109/TPAMI.2021.3087709.

[2] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation, 2021. URL https://arxiv.org/abs/2010.13154.

[3] Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang. Unispeech: Unified speech representation learning with labeled and unlabeled data, 2021. URL https://arxiv.org/abs/2101.07597.