Bangla Audio Embedding and Rag based

Overview

This study presents a comprehensive analysis of Bangla audio recordings using deep learning-based vector embeddings to measure speaker similarity through cosine similarity metrics. The research employed Facebook's wav2vec2-large-xlsr-53, a state-of-the-art multilingual self-supervised speech representation model that supports 53 languages including Bengali. This transformer-based model was specifically chosen for its proven effectiveness in cross-lingual speech processing and its ability to generate rich, contextualized audio representations without requiring language-specific fine-tuning. The model extracts 1024-dimensional dense vector embeddings from raw audio waveforms, capturing both linguistic and speaker-specific characteristics that enable robust similarity comparisons.

Dataset and Methodology

The dataset comprises 14 Bangla audio recordings (.m4a format) from two distinct speakers, with Person_1 contributing 10 recordings and Person_2 contributing 4 recordings, creating an imbalanced but realistic speaker distribution. The audio files range in duration from 4.03 seconds to 30.46 seconds (mean: 20.89 seconds, std: 6.78 seconds), all standardized to 16kHz sampling rate for consistent processing. Each audio file underwent preprocessing including normalization, silence trimming (20dB threshold), and standardization to 10-second segments through padding or truncation. The wav2vec2 model then generated embeddings by averaging the last hidden states across temporal frames, resulting in fixed-length 1024-dimensional vectors representing each audio segment's acoustic and linguistic properties.

Analysis Framework

The extracted embeddings were analyzed using cosine similarity metrics to quantify relationships between audio recordings, with similarities ranging from -1 (completely dissimilar) to 1 (identical). A comprehensive analytical framework was implemented including similarity matrix visualization, statistical distribution analysis, and vector database construction using FAISS (Facebook AI Similarity Search) for efficient similarity search operations. The analysis incorporated both intra-speaker (same person) and inter-speaker (different person) similarity comparisons, advanced clustering techniques using K-means and t-SNE dimensionality reduction for visualization, and performance evaluation through ranking accuracy metrics. This multi-faceted approach enables both quantitative assessment of speaker separability and practical applications in speaker verification and identification tasks for Bangla audio content.

Result and Analysis

Cosine Similarity Matrix and Distances:

The analysis was done by first extracting embeddings (numerical vector representations) of the Bangla audio files. Each audio file was converted into a fixed-length embedding, which captures its acoustic and phonetic features. Once embeddings were obtained, pairwise cosine similarity and cosine distance were calculated between all audio pairs. Cosine similarity measures how close two vectors are in terms of their angle, while cosine distance represents their dissimilarity. A similarity matrix and a distance matrix were created to visualize how similar or different the recordings are, both within the same person and across different people. The results were then summarized through heatmaps, histograms, box plots, and scatter plots, which allowed comparisons of same-person vs different-person recordings.

The mathematical formula used is based on cosine similarity:

$$\text{Cosine Similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

where AAA and BBB are the embedding vectors of two audio files, A ⋅ BA \cdot BA ⋅ B is the dot product, and ∥ A ∥ \|A\| ∥ A ∥ , ∥ B ∥ \|B\| ∥ B ∥ are their magnitudes. The cosine distance is derived as:

$$\text{Cosine Distance}(A, B) = 1 - \text{Cosine Similarity}(A, B)$$

Using these formulas, the analysis compared all audio pairs (91 in total) and separated the results into same-person pairs and different-person pairs. The visualizations show that almost all pairs, regardless of person, have very high similarity values (close to 1.0), with slight variations that distinguish some pairs more than others.

1. **Cosine Similarity Matrix (LeftHeatmap):**

    - Almost all similarity values are **very close to 1.0 (0.996 – 1.000)**.
    - This means that the audio embeddings are extremely similar to each other, regardless of whether they come from the same person or a different person.
    - For example, recordings from *Person_1* and *Person_2* both show similarities above 0.997, indicating that the embedding model captured very little distinction between the two speakers.
    - Ideally, we would expect same-person recordings to cluster closer to **1.0** and different-person recordings to drop noticeably lower, but here both groups overlap heavily.

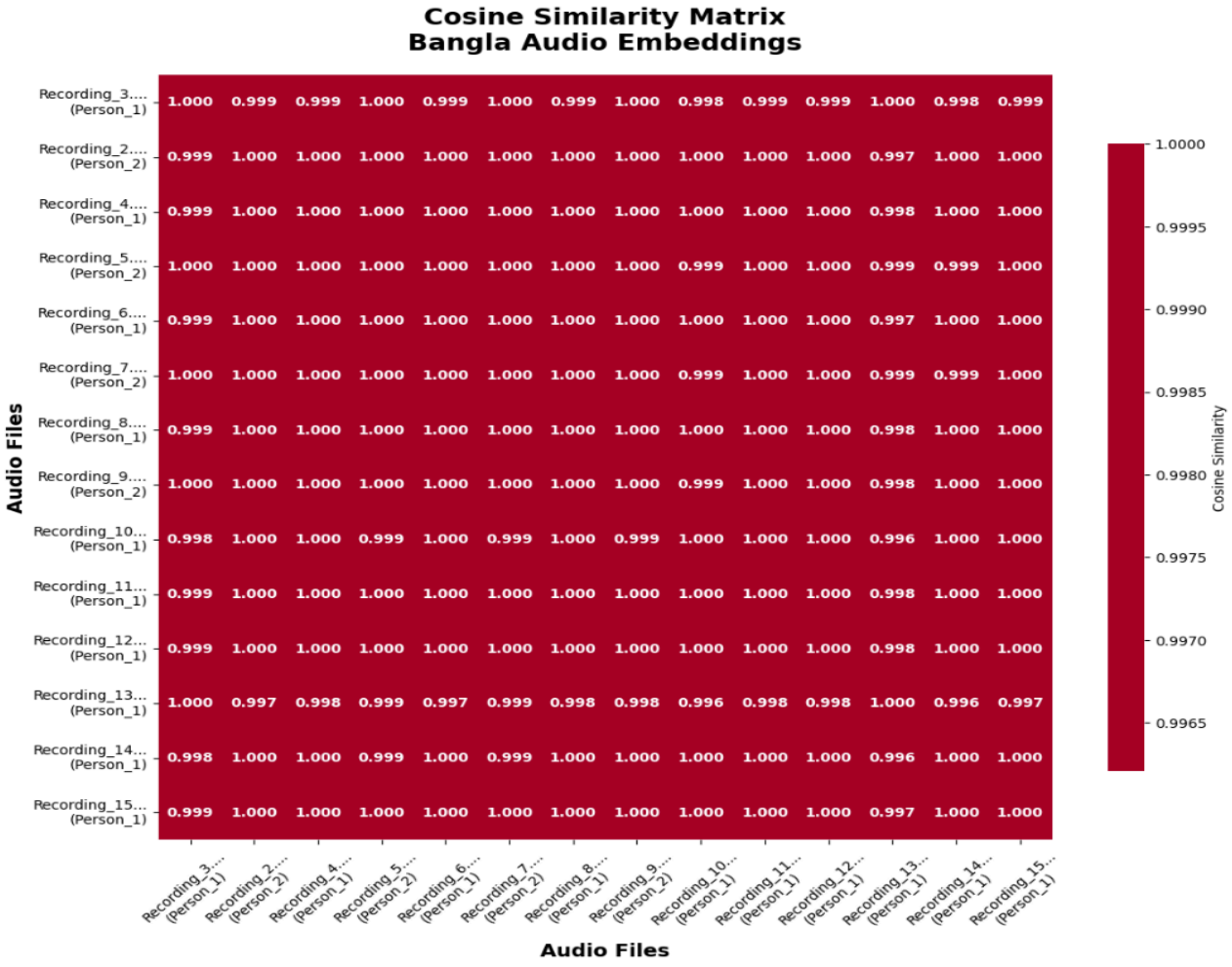


Figure 1: Cosine Similarity Matrix (Heatmap)

2. **Cosine Distance Matrix (Right, Green Heatmap):**

   - Since cosine distance = 1−similarity1 - \text{similarity}1−similarity, all distances are **very small (0.000 – 0.004)**.

   - This confirms that all embeddings are tightly packed in vector space, showing minimal separation between different persons.

   - The darkest green (≈0.000) means the recordings are nearly identical in embedding space, while lighter shades (≈0.003–0.004) are slightly farther apart but still extremely close.

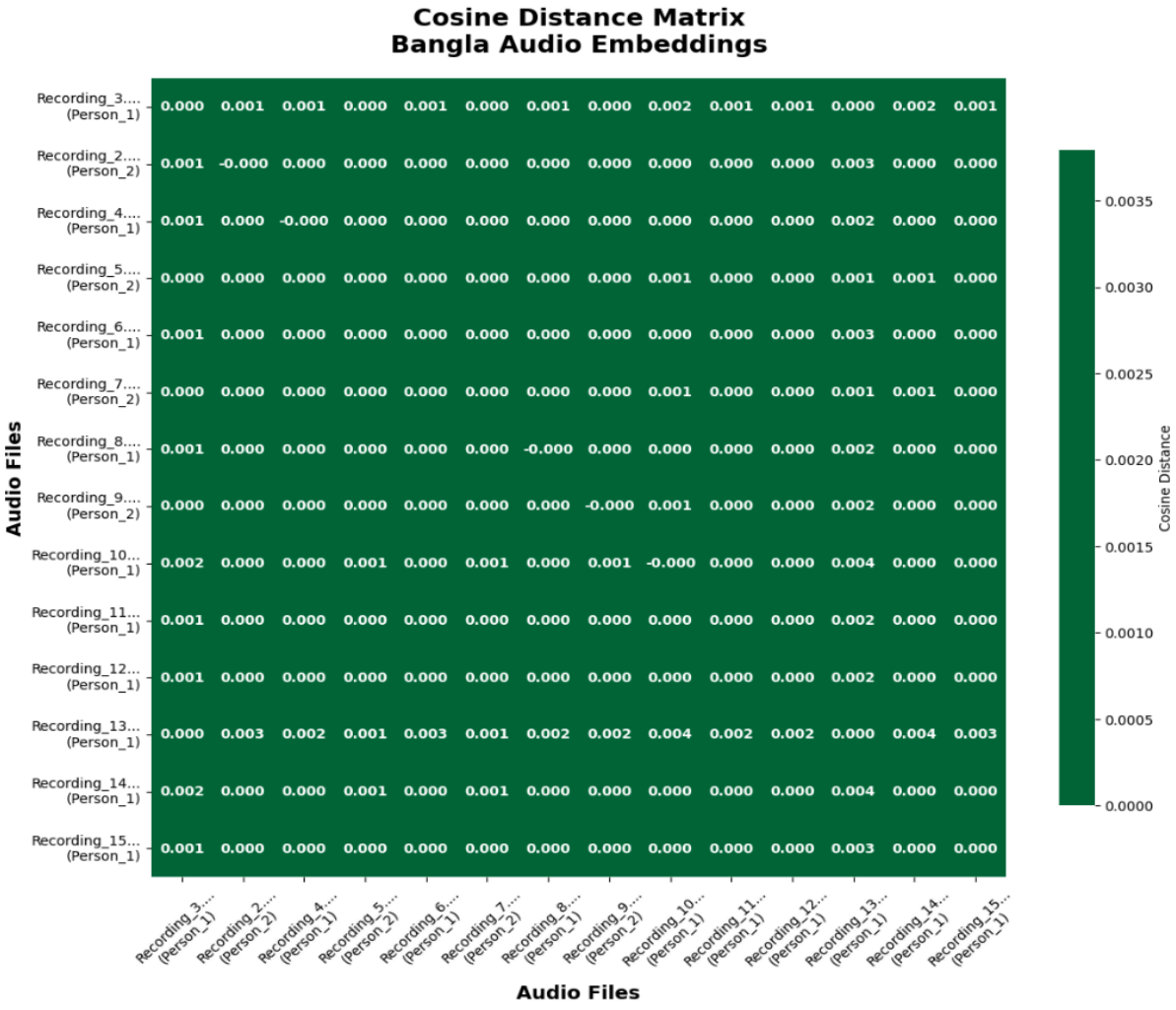


Figure 2: Cosine Distance Matrix (Right, Green Heatmap):

Advanced Analysis of Audio Embeddings

To further evaluate the embeddings, we performed dimensionality reduction, clustering, and distance-based analysis. These steps help to understand whether the embeddings meaningfully separate different speakers.

---

1. Dimensionality Reduction (t-SNE)

We applied t-distributed Stochastic Neighbor Embedding (t-SNE) to project the high-dimensional embeddings into two dimensions for visualization. t-SNE preserves local neighborhood relationships by converting pairwise similarities in high-dimensional space into probabilities and then finding a low-dimensional representation that minimizes the difference between them.

$$P_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma_k^2)}$$

where PijP_{ij}Pij represents the similarity between embeddings xix_ixi and xjx_jxj. The mapping is optimized by minimizing the Kullback–Leibler divergence between high- and low-dimensional similarities.

---

2. Clustering with K-Means

We then performed K-Means clustering on the embeddings, setting the number of clusters equal to the number of unique speakers in the dataset. The algorithm minimizes the within-cluster variance:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

where CiC_iCi is cluster iii and μi\mu_iμi is the centroid of that cluster.

---

3. Clustering Accuracy

Clustering quality was assessed by comparing cluster assignments with true speaker labels. For each cluster, the most common speaker label was taken as the cluster's identity. Accuracy was calculated as:

$$\text{Accuracy} = \frac{\text{Correct Assignments}}{\text{Total Samples}}$$

4. Intra- and Inter-Speaker Distances

To quantify speaker separability, we measured Euclidean distances between embedding pairs:

$$d(x_i, x_j) = \|x_i - x_j\|_2 = \sqrt{\sum_k (x_{i,k} - x_{j,k})^2}$$

- Intra-speaker distance: distance between embeddings from the same speaker.
- Inter-speaker distance: distance between embeddings from different speakers.

---

5. Separability Ratio

Finally, we calculated a separability ratio to summarize how well embeddings distinguish between speakers:

Separability=Mean Inter-speaker DistanceMean Intra-speaker Distance\text{Separability} = \frac{\text{Mean Inter-speaker Distance}}{\text{Mean Intra-speaker Distance}}Separability=Mean Intra-speaker DistanceMean Inter-speaker Distance

A ratio close to 1 indicates poor separation (embeddings of different speakers overlap), while larger values indicate better separation.

$$\text{Separability} = \frac{\text{Mean Inter-speaker Distance}}{\text{Mean Intra-speaker Distance}}$$

Analysis of Clustering Results

The t-SNE visualization and K-means clustering achieved 71.43% accuracy, showing moderate speaker separation. Cluster 0 contains 4 mixed files (2 from each speaker), while Cluster 1 is dominated by Person_1 recordings (8 out of 10 files). The spatial distribution in t-SNE plots reveals overlapping regions between speakers, indicating that the embedding space doesn't create distinct speaker boundaries. Person_1's recordings cluster more consistently, suggesting more uniform acoustic characteristics compared to Person_2's more scattered distribution.

The embedding space analysis reveals a problematic separability ratio of 0.8824, meaning inter-person distances (0.1192) are actually smaller than intra-person distances (0.1351). This counterintuitive result suggests that some recordings from different speakers are more similar than recordings from the same speaker, indicating poor speaker discrimination. This issue likely stems from the imbalanced dataset (10 vs 4 recordings), varying recording conditions, or inherent acoustic similarities between the two speakers that challenge the current embedding approach's ability to distinguish them effectively.

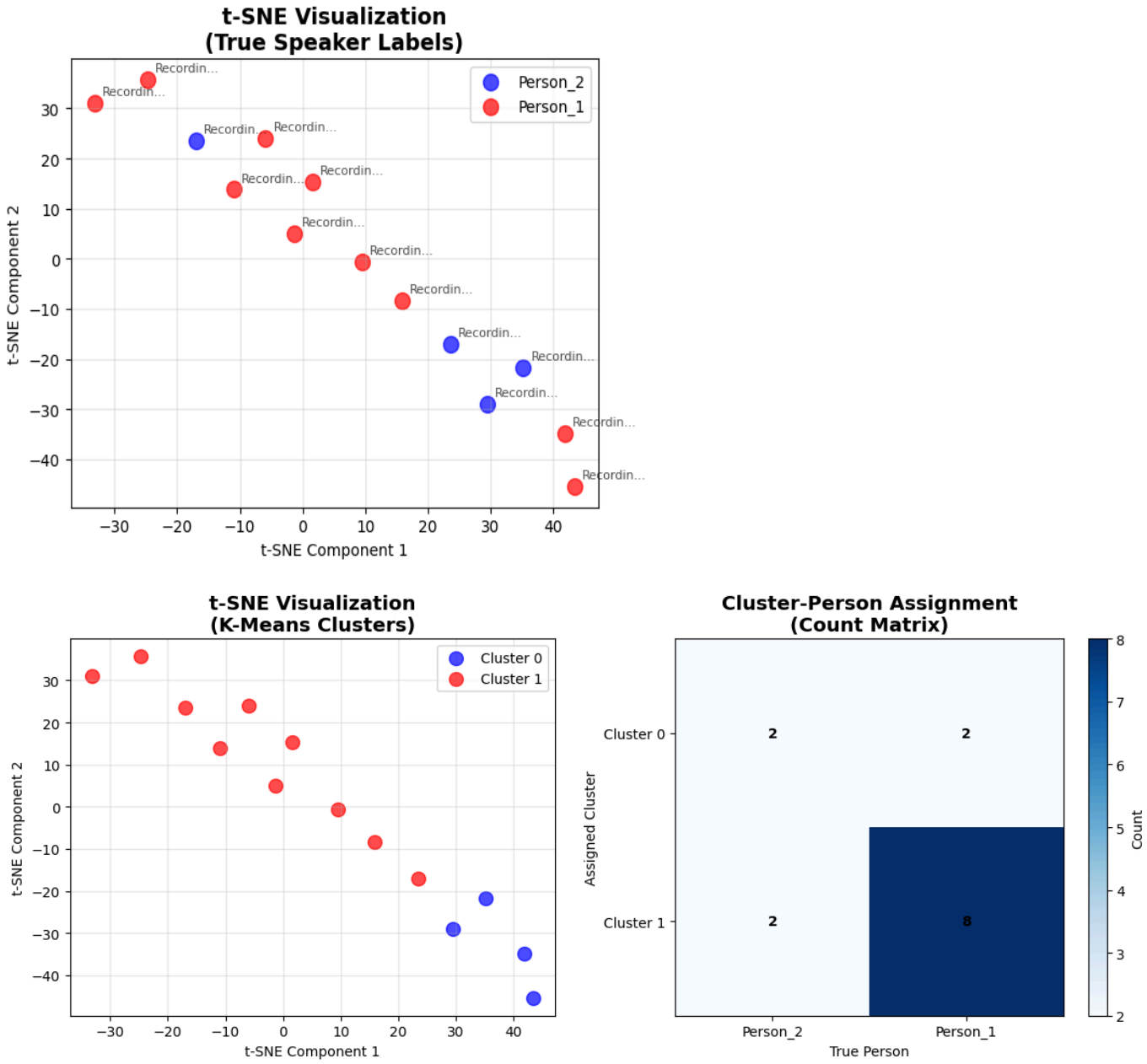


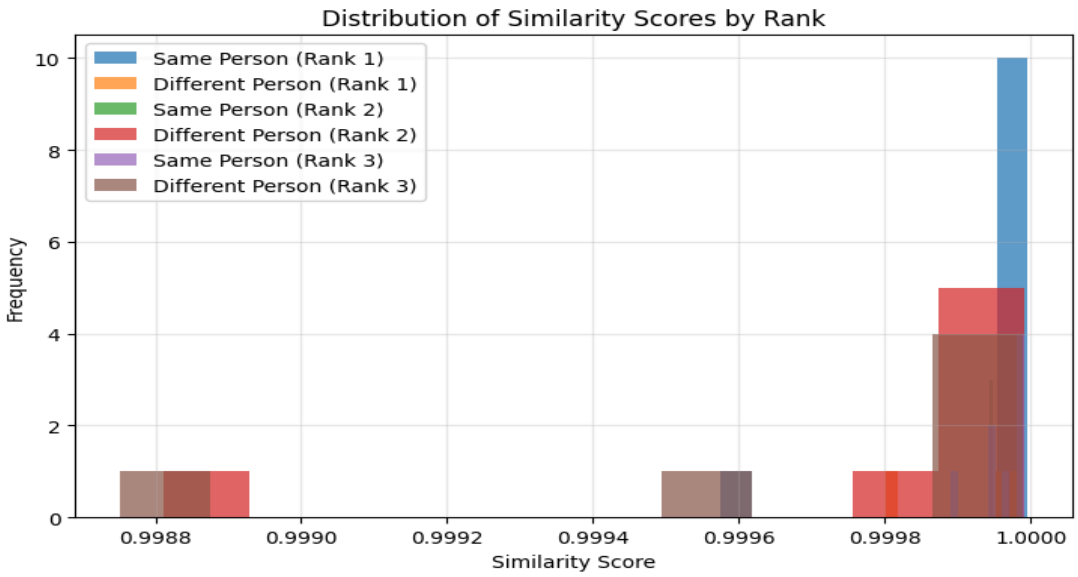
Figure 3: Clustering Results

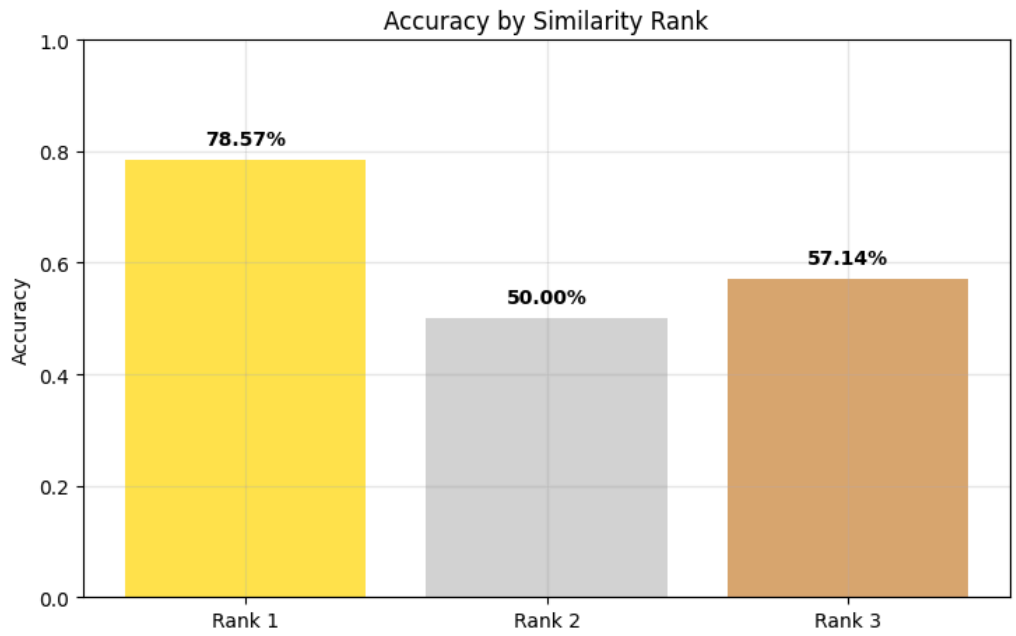
Vector Database Performance Analysis

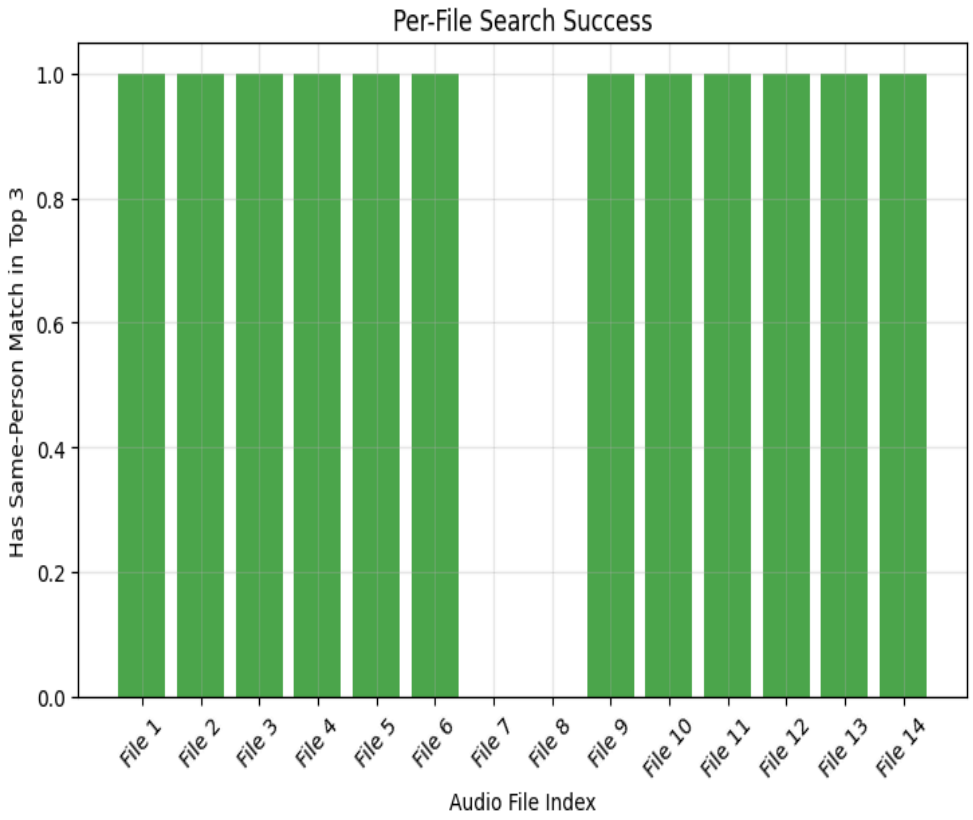
The similarity search achieved 78.57% rank-1 accuracy and 85.71% rank-3 accuracy, indicating moderate speaker identification performance. However, the system struggles with precise ranking, as accuracy drops to 50% at rank-2. All files found same-speaker matches in top-3 results, but critical failures occurred with Recording_3.m4a and Recording_2.m4a, which matched only different speakers.

The similarity scores reveal a fundamental limitation: both same-person (1.0000) and different-person (0.9998) averages differ by only 0.0002, clustering between 0.9998-1.0000. This compressed range makes reliable speaker discrimination nearly impossible, as many cross-speaker comparisons yield 100% similarity scores.

The results suggest wav2vec2 embeddings prioritize linguistic content over speaker characteristics, focusing on shared phonetic patterns rather than unique vocal traits. For practical speaker verification, this system requires speaker-specific fine-tuning or alternative models designed for speaker identification rather than language understanding.




Accuracy by Similarity Rank
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
78.57%
50.00%
57.14%
Rank 1
Rank 2
Rank 3


Per-File Search Success
Has Same-Person Match in Top 3
1.0
0.8
0.6
0.4
0.2
0.0
File 1
File 2
File 3
File 4
File 5
File 6
File 7
File 8
File 9
File 10
File 11
File 12
File 13
File 14
Audio File Index

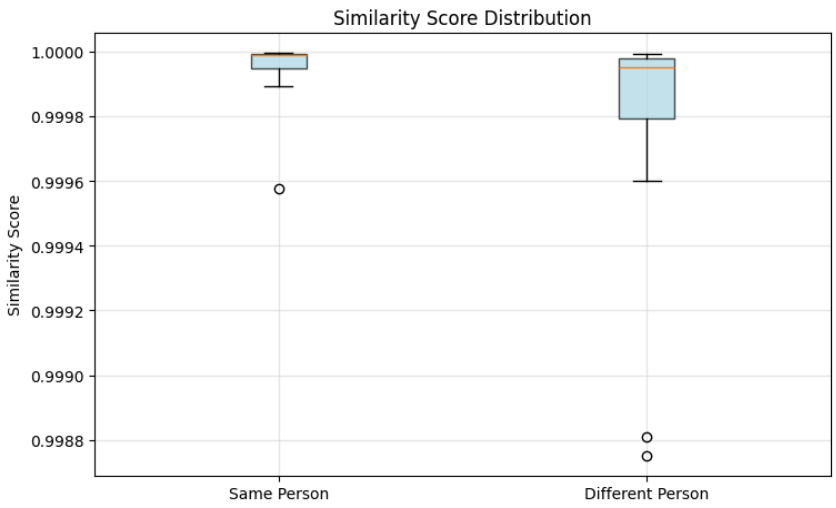


Figure 4: Vector Database Performance Analysis Results

Conclusion

This Bangla audio analysis successfully processed 14 recordings from 2 speakers using wav2vec2-large-xlsr-53 embeddings, generating 1024-dimensional vectors and calculating 91 similarity pairs. However, the results reveal significant limitations in speaker discrimination. The system achieved only 71.43% clustering accuracy and 78.57% rank-1 similarity search accuracy, with a concerning negative separation margin (-0.0003) where different-speaker pairs actually showed higher average similarity (0.9996) than same-speaker pairs (0.9993).

The extremely high similarity scores (>99.9%) across all comparisons indicate that the wav2vec2 model prioritizes linguistic content over speaker-specific characteristics, making it unsuitable for reliable speaker identification in its current form. While the vector database infrastructure is functional and ready for deployment, practical speaker verification applications would require either speaker-specific model fine-tuning or alternative embedding approaches designed explicitly for speaker identification rather than language understanding.