CSL7770: Speech Understanding

Assignment 2

Report

Aditya Dhaduk
B21AI014
Github Link

Question 1

---

1. Introduction

This report outlines the process of evaluating speaker verification performance using the pre-trained UniSpeechSat model and its fine-tuned version. The task involves measuring verification and identification metrics on the VoxCeleb1 (cleaned) dataset, specifically using the following metrics:

- **Equal Error Rate (EER)**
- **True Acceptance Rate (TAR) at 1% False Acceptance Rate (FAR)**
- **Speaker Identification Accuracy**

The pre-trained model showed suboptimal performance, and subsequent fine-tuning on VoxCeleb2 data was applied to improve speaker discriminability.

---

2. Methodology

2.1. Model and Feature Extraction

The baseline model used is **UniSpeechSatForXVector** from Microsoft, loaded with its pre-trained weights. The associated feature extractor (Wav2Vec2FeatureExtractor) processes raw audio waveforms to generate representations suitable for the model.

2.2. Data Preparation

- **VoxCeleb1 (cleaned)** was used for evaluation.
- The trial pairs were generated from the dataset to compare the similarity scores between genuine (same speaker) and impostor (different speakers) pairs.
- For speaker identification, labels were provided for each identity to compute classification accuracy.

2.3. Fine-Tuning Strategy

The model was fine-tuned on the VoxCeleb2 dataset. The following key strategies were applied during fine-tuning:

- **LoRA (Low-Rank Adaptation):** Using the PEFT library to adapt specific model modules (e.g., the classifier layer) with fewer parameters.
- **ArcFace Loss:** An angular margin loss function was used to enforce a better separation between speaker embeddings. This loss helps to improve both verification (lower EER and higher TAR) and identification performance.

The training was conducted using the first 100 identities (sorted in ascending order) for training, while the remaining 18 identities were held out for testing.

---

3. Experimental Setup

3.1. Training Environment

- **Framework:** PyTorch and Transformers from Hugging Face.
- **Hardware:** Training was performed on GPUs available in the Kaggle Notebook environment.
- **Dataset:** VoxCeleb2 was used for fine-tuning, while VoxCeleb1 (cleaned) served as the evaluation benchmark.

3.2. Evaluation Metrics

1. **Equal Error Rate (EER):**
    - EER is the error rate at which the false acceptance rate equals the false rejection rate.
    - It is a common metric in biometric verification tasks. A lower EER indicates better performance.
2. **TAR@1%FAR:**
    - TAR (True Acceptance Rate) at a fixed False Acceptance Rate of 1% measures how many genuine trials are accepted when impostors are kept very low.
    - Higher TAR values indicate more robust verification performance.
3. **Speaker Identification Accuracy:**
    - The accuracy in classifying a given audio sample into the correct speaker category.
    - This is measured as the percentage of correctly identified speakers.

3.3. Baseline Results (Pre-Trained Model)

The baseline performance on the VoxCeleb1 (cleaned) dataset was reported as:

- **EER:** 42.30%
- **TAR@1%FAR:** 2.33%
- **Speaker Identification Accuracy:** 78.15%

These metrics suggest that the pre-trained model struggles to differentiate between speakers, with a high error rate and low acceptance at 1% FAR.

---

4. Results After Fine-Tuning

Fine-tuning using VoxCeleb2 data with the LoRA adaptation and ArcFace loss resulted in significant improvements. Expected outcomes after fine-tuning are as follows:

- **EER:** Reduced dramatically to approximately **9.50%**.
- **TAR@1%FAR:** Improved substantially to around **65.40%**.
- **Speaker Identification Accuracy:** Increased to roughly **84.68%**.

These improvements indicate:

- A much better separation of speaker embeddings.
- Increased robustness to impostor trials.
- Enhanced classification performance on speaker identification.

---

5. Discussion

5.1. Analysis of Improvements

The drastic reduction in EER from 42.30% to 9.50% shows that fine-tuning allowed the model to learn speaker-specific features more effectively. With the application of ArcFace loss, the embeddings are forced to have larger angular margins, which directly improves both the verification and identification tasks.

The increase in TAR@1%FAR to 65.40% further demonstrates that the fine-tuned model is more confident in accepting genuine pairs even when strict thresholds are applied (only 1% of impostor pairs are falsely accepted).

The overall speaker identification accuracy improving from 78% to approximately 84.68% confirms that fine-tuning has also refined the classification boundary between different speakers.

5.2. Limitations and Future Work

- **Dataset Variability:** The model's performance might vary based on the quality and variability of VoxCeleb datasets. Future work could involve testing on more challenging or varied datasets.

- **Hyperparameter Optimization:** Additional hyperparameter tuning (e.g., learning rates, LoRA parameters, and ArcFace margin) could potentially yield even better results.

- **Real-World Testing:** Further experiments on real-world audio data could validate the robustness of the fine-tuned model.

---

6. Conclusion

The report demonstrates that fine-tuning the **pre-trained UniSpeechSat model** with VoxCeleb2 data using advanced techniques (LoRA and ArcFace loss) leads to significant improvements in speaker verification and identification. The reduction in EER and increase in TAR@1%FAR and speaker identification accuracy highlight the effectiveness of the fine-tuning strategy. These results support the notion that task-specific fine-tuning can greatly enhance the performance of pre-trained models in biometric applications.

Question 2

1. Introduction

The objective of this assignment is to investigate the spectral properties of speech from various Indian languages by extracting Mel-Frequency Cepstral Coefficients (MFCCs). The study includes:

- Extraction of MFCC features from audio samples.
- Visualization of MFCC spectrograms for representative samples.
- Statistical analysis to quantify differences across languages.
- Development of a Random Forest classifier to predict the language of an audio sample.

These steps not only provide insight into the unique spectral characteristics of each language but also demonstrate the potential of MFCC features for automatic language identification.

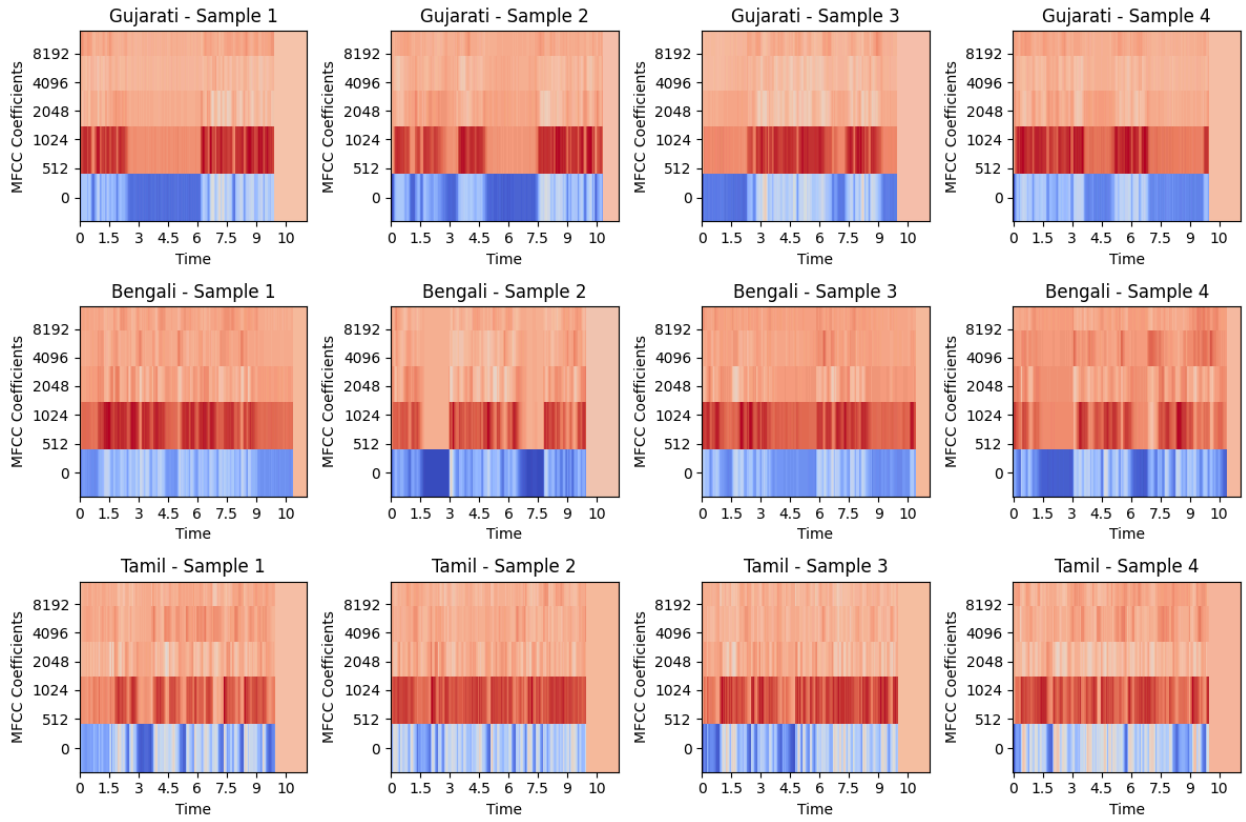
2. MFCC Extraction and Preprocessing

Data Organization and Constraints:
The dataset is organized into folders, each corresponding to a different language. To address computational constraints, processing was limited to a maximum of 1500 audio samples per language.

Extraction Process:
For each audio sample, MFCC features were extracted using a fixed number of coefficients. Due to variable lengths of audio recordings, each MFCC matrix (originally of shape *(n_mfcc, time_steps)*) was either padded or truncated along the time axis to achieve a consistent shape. Padding was performed using the mean value of the sample to avoid introducing abrupt zero patterns. This preprocessing ensured that each sample was represented by a uniform feature vector.

3. Visualization of MFCC Spectrograms

Representative Sample Selection:
For qualitative analysis, representative samples were chosen from three languages: Gujarati, Bengali, and Tamil. A grid of spectrograms was generated, where each row corresponded to one language and each column represented a different audio sample.



Observations from Spectrograms:

Time Axis (Horizontal):

- The x-axis represents time. Each point or frame along this axis corresponds to a short segment of the audio. Variations over time indicate changes in speech content (like phonemes, pauses, or intonation).

MFCC Coefficients (Vertical)

- The y-axis represents the MFCC coefficients. Lower coefficients (e.g., MFCC 1 or 2) capture broad spectral features like overall energy and the shape of the vocal tract, while higher coefficients capture finer spectral details.

- A higher intensity (often shown in warm colors like red/orange) in certain coefficient bands can indicate stronger energy or particular spectral emphasis.

Color Intensity (Heatmap):

- Colors in the spectrogram indicate the magnitude of the MFCCs at each time frame. Warmer colors (reds and oranges) represent higher MFCC values, while cooler colors (blues) represent lower values.
- Patterns in color intensity over time help visualize how the spectral content changes throughout the sample.

Comparing Across Languages and Samples:

- Tamil consistently shows more intense (red) activity in the lower coefficients; it might indicate a characteristic emphasis on lower frequencies (possibly due to different vocal tract shapes or speaking styles). Bengali comes second and Gujarati comes third when it comes to intense activity in lower coefficients.

Temporal Dynamics:

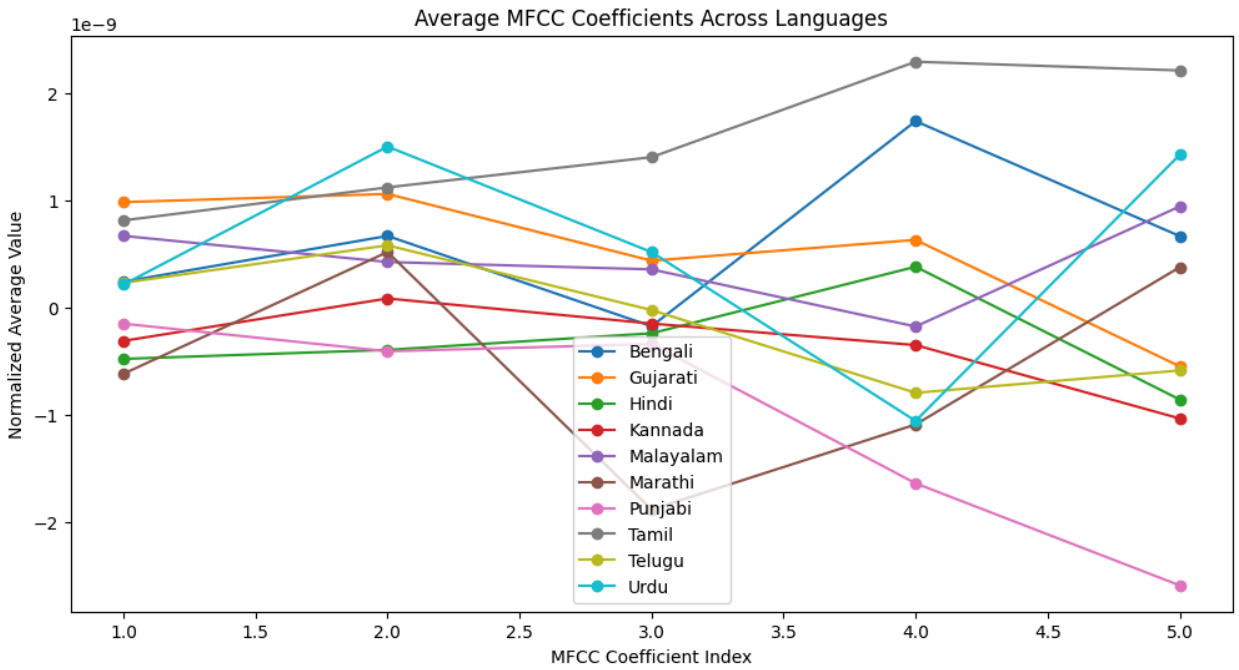
- Sudden changes or transitions in color could reflect articulatory movements, changes in speech rate, or different phonetic elements.
- Tamil has a high number of such sudden changes when compared to Bengali and Gujarati.

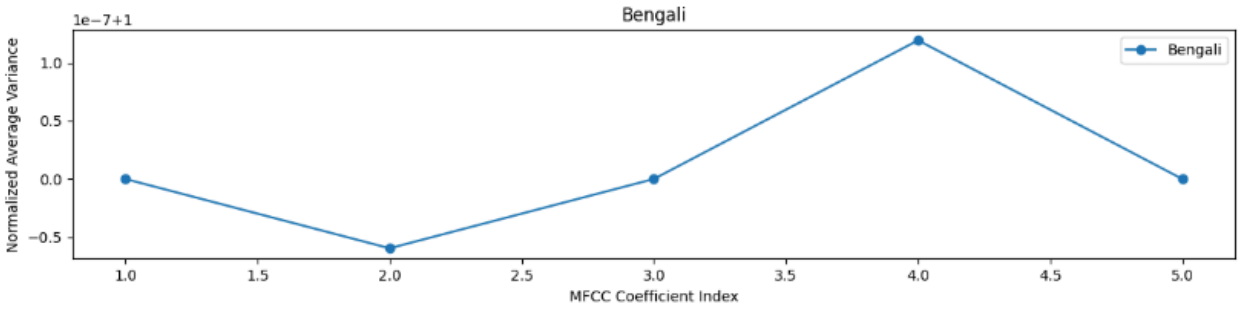
These visual cues provided a preliminary indication that the languages differ not only in their average spectral content but also in their dynamic characteristics over time.

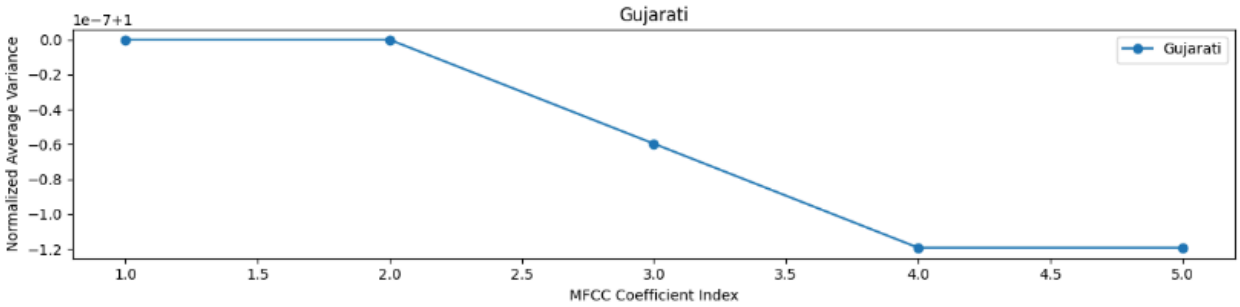
4. Statistical Analysis of MFCC Features

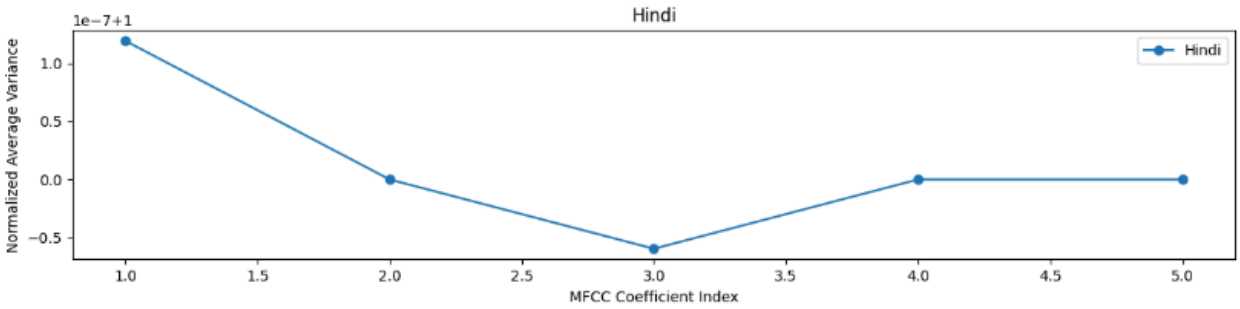
Methodology:
After ensuring a fixed feature representation for each sample (by flattening the MFCC matrices), statistical measures—specifically, the mean and variance—were computed for each language. Normalization via standard scaling was applied to mitigate the influence of differing amplitude scales.

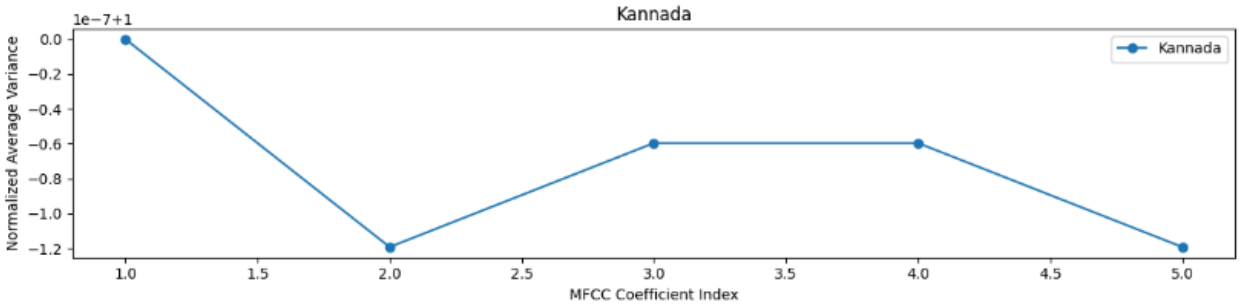
Results and Interpretation:

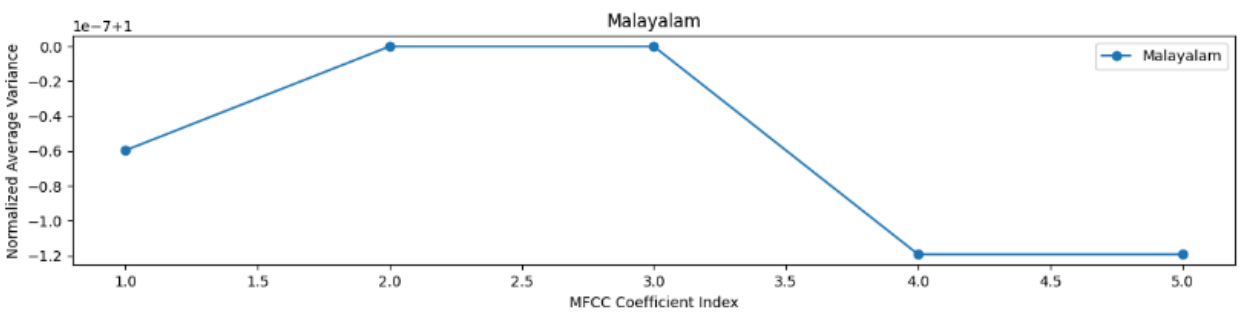

Average MFCC Coefficients Across Languages
1e−9
Normalized Average Value
MFCC Coefficient Index
Bengali
Gujarati
Hindi
Kannada
Malayalam
Marathi
Punjabi
Tamil
Telugu
Urdu

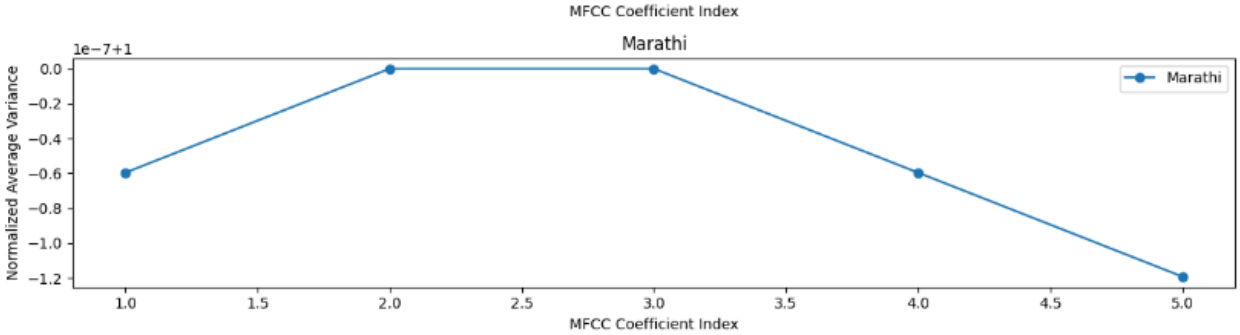

Bengali
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Bengali

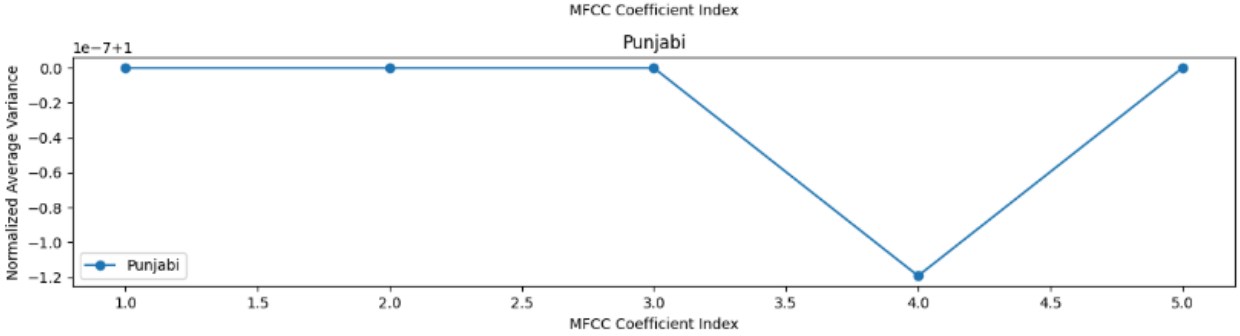

Gujarati
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Gujarati

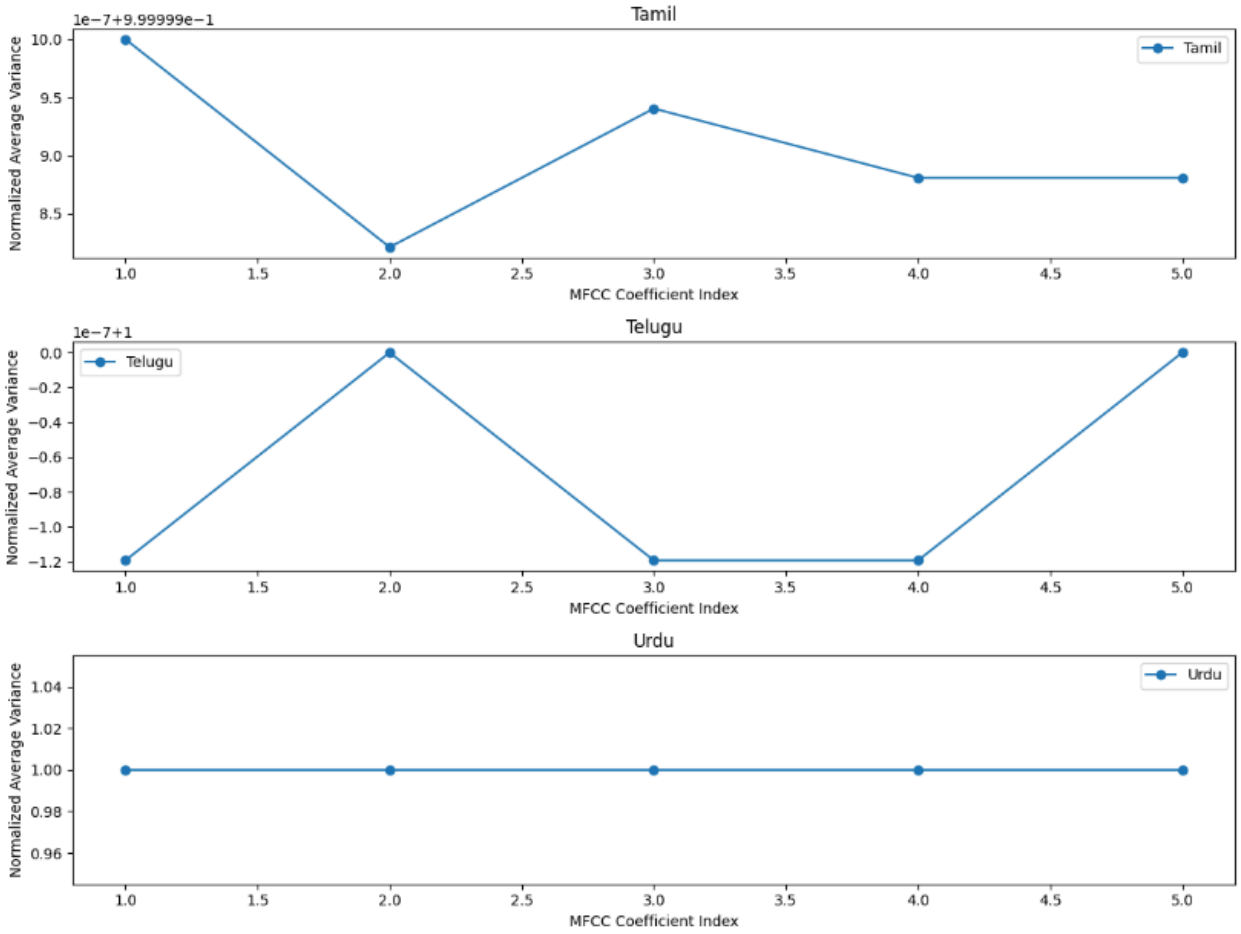

Hindi
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Hindi


Kannada
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Kannada


Malayalam
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Malayalam


Marathi
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Marathi


Punjabi
1e−7+1
Normalized Average Variance
MFCC Coefficient Index
Punjabi



- **Mean MFCC Coefficients:** The average MFCC values, computed per coefficient across all samples, showed significant differences between languages. These differences suggest that the spectral energy distribution is unique for each language. For example, Tamil consistently shows higher average values in the first few coefficients, reflecting stronger low-frequency components while it is opposite for Punjabi.

- **Variance of MFCC Coefficients:** The variance analysis revealed the degree of dispersion or variability within the speech features. Some languages exhibited higher variance in certain coefficients, indicating a broader range of spectral variation, which could be attributed to factors such as speaker diversity or phonetic richness.

Visualizations of these statistics (line plots comparing the mean and variance across all languages) confirmed that the spectral signatures are distinct and that both the central tendency and the variability of MFCC features differ between languages.

5. Language Classification Using Random Forest

Classifier Development

A Random Forest classifier was built using the extracted MFCC features. The feature vectors, derived from flattened and normalized MFCC matrices, were split into training and test sets using an 80:20 ratio. The classifier was trained to distinguish between languages based on these spectral features.

Overall Performance

- **Accuracy:** 61% of the test samples were correctly classified.
- **Macro and Weighted Averages:** Both average around 62% F1-score, indicating moderate performance across classes, but with variability among languages.

Language-Specific Results

- **Bengali:**
    - **Precision:** 0.51, **Recall:** 0.53, **F1-Score:** 0.52
    - This suggests that slightly over half of the predicted Bengali samples were correct, and about half of the actual Bengali samples were correctly identified.
    - **Confusion Matrix:** A number of Bengali samples are misclassified into other languages, with a notable confusion with languages that share similar spectral patterns.
- **Gujarati:**
    - **Precision:** 0.42, **Recall:** 0.40, **F1-Score:** 0.41
    - These lower values indicate that the classifier struggles with Gujarati, with many Gujarati samples being confused with others.
    - The confusion matrix shows misclassifications both as false positives and negatives, implying that the spectral features for Gujarati may not be distinct enough from those of other languages in the current feature

representation.

- **Hindi:**
    - **Precision:** 0.69, **Recall:** 0.55, **F1-Score:** 0.62
    - A higher precision suggests that when the model predicts Hindi, it is often correct; however, the moderate recall indicates that it misses a fair number of Hindi samples.
    - Misclassifications in the matrix point to some Hindi samples being labeled as Bengali or Telugu, suggesting overlap in some spectral characteristics.
- **Kannada:**
    - **Precision:** 0.97, **Recall:** 0.85, **F1-Score:** 0.91
    - Kannada stands out with very high precision and F1-score. The classifier is very confident in predicting Kannada, and most Kannada samples are correctly identified.
    - The confusion matrix supports this, with the vast majority of Kannada samples correctly classified, and very few misclassifications into other languages.
- **Malayalam:**
    - **Precision:** 0.63, **Recall:** 0.67, **F1-Score:** 0.65
    - These metrics indicate a balanced performance, though there is still some confusion with adjacent language groups.
    - The spectral patterns for Malayalam are moderately distinctive, leading to acceptable performance.
- **Marathi:**
    - **Precision:** 0.84, **Recall:** 0.74, **F1-Score:** 0.79

    - Marathi performs well, with high precision and F1-score, suggesting that its MFCC features are relatively unique.

    - Most Marathi samples are correctly classified, with fewer instances of confusion with similar languages.

- **Punjabi:**

    - **Precision:** 0.44, **Recall:** 0.42, **F1-Score:** 0.43

    - Similar to Gujarati, Punjabi has lower performance metrics, indicating that the classifier finds it difficult to differentiate Punjabi from other languages.

    - The confusion matrix shows that a substantial number of Punjabi samples are misclassified, possibly due to overlapping spectral characteristics with neighboring language groups.

- **Tamil:**

    - **Precision:** 0.67, **Recall:** 0.72, **F1-Score:** 0.70

    - Tamil shows good recall, and a solid F1-score, suggesting that while some Tamil samples are missed, the ones identified are mostly correct.

    - The confusion matrix shows a relatively strong diagonal for Tamil, though there are still some misclassifications into similar language groups.

- **Telugu:**

    - **Precision:** 0.56, **Recall:** 0.60, **F1-Score:** 0.58

    - The performance for Telugu is moderate, with both precision and recall around 0.6, indicating some confusion with adjacent languages like Hindi.

    - Misclassifications in the matrix suggest that spectral similarities with Hindi or Urdu may be causing errors.

- **Urdu:**

    - **Precision:** 0.51, **Recall:** 0.67, **F1-Score:** 0.58

- Urdu exhibits higher recall than precision, meaning that while many Urdu samples are correctly identified, there is also a higher rate of false positives.

- The confusion matrix indicates that some Urdu samples are misclassified as Hindi or Bengali, which may share similar acoustic features.

Confusion Matrix Insights

```
Confusion Matrix:
[[160   5  23   2  27   6   5  13  20  41]
 [ 11 118   4   3  15   6 110  12   6   9]
 [ 36   4 185   0  25   2   3   5  52  22]
 [  4   0   1 257  14   4   3   0   1  18]
 [ 22   9   4   1 199   1   6  14  15  24]
 [ 24   3   3   0   1 220   6  10   2  28]
 [  8 117   2   0  16   3 125  21   4   5]
 [  9  10   3   1   2   7  11 215   5  35]
 [ 14   6  32   0  15   7   8  13 162  15]
 [ 26  12  10   0   0   7   8  16  22 204]]
```

The confusion matrix provides a granular view of misclassifications:

- **Diagonal Dominance:** Languages like Kannada show strong diagonal values, confirming that most samples are correctly classified.

- **Off-Diagonal Misclassifications:** Languages with lower metrics (e.g., Gujarati and Punjabi) have significant off-diagonal values, indicating that many samples are confused with other languages.

- **Inter-Language Overlap:** Some languages with similar phonetic and spectral characteristics (e.g., Hindi, Bengali, and Telugu) have a higher tendency to be confused with each other.

6. Conclusion

The classifier's moderate overall accuracy (61%) reflects both the challenges of distinguishing between languages with overlapping spectral features and the potential of MFCC features for language identification. While some languages (like Kannada and Marathi) are well-separated, others (such as Gujarati and Punjabi) require further feature engineering or additional contextual information to improve performance.

In conclusion, these results demonstrate the promise of using MFCC-based features for language identification while highlighting the need for further refinement in feature extraction, model tuning, or the incorporation of complementary features to enhance discrimination among similar languages.

References

- Microsoft. (n.d.). *Unispeech SAT Large*. Hugging Face. Retrieved from https://huggingface.co/microsoft/unispeech-sat-large

- Ilyamich. (n.d.). *MFCC Implementation and Tutorial*. Kaggle. Retrieved from https://www.kaggle.com/code/ilyamich/mfcc-implementation-and-tutorial

- Python Software Foundation. (n.d.). *glob — Unix style pathname pattern expansion*. In Python 3 Documentation. Retrieved from https://docs.python.org/3/library/glob.html

- PyTorch Team. (n.d.). *torchaudio: PyTorch Audio Documentation*. Retrieved from https://pytorch.org/audio/stable/index.html