Unsupervised Speaker Learning System

Overview

The system now implements a smart, unsupervised learning approach for speaker identification that automatically discovers and labels speakers from passively collected conversation data. This replaces the need for manual training data collection with an intelligent system that learns from real usage.

Smart Learning Architecture

Three-Tier Fallback System

1. **Unsupervised Clustering** (Primary)
   - Automatically discovers speakers from collected audio features
   - Uses K-means or DBSCAN clustering algorithms
   - No manual labeling required
2. **Advanced ML Identification** (Fallback)
   - Pre-trained Random Forest classifier
   - Requires manual training data
   - Used when unsupervised clustering isn't available
3. **Basic Audio Analysis** (Final Fallback)
   - Simple energy/pitch change detection
   - Generic speaker labels (Speaker A, B, C)
   - Always available as safety net

Data Collection Strategy

Passive Audio Feature Collection

- **Real-time Storage**: Audio features are collected during normal conversation
- **Feature Buffering**: Accumulates features over time for better representation
- **Automatic Association**: Links audio features with transcribed text
- **Dual Database**: Separate collections for conversations and audio features

Feature Engineering

The system extracts 14 comprehensive voice characteristics: 1. **Energy** - Mean absolute amplitude 2. **Pitch Estimate** - Standard deviation of audio 3. **Zero Crossings** - Frequency content indicator 4. **Spectral Centroid** - Brightness measure 5. **Energy Variance** - Stability indicator 6. **Peak Amplitude** - Loudness measure 7. **RMS Energy** - Power measure 8-10. **MFCC Coefficients** - Spectral shape features 11-12. **Formants** - Vocal tract characteristics 13. **Jitter** - Pitch variation 14. **Shimmer** - Amplitude variation

Learning Workflow

1. Data Collection Phase

```
Conversation → Audio Features → Database Storage
     ↓              ↓              ↓
   Text        Feature Vector    Metadata
```

2. Clustering Phase

```
Audio Features → Feature Scaling → Clustering → Speaker Labels
      ↓              ↓              ↓           ↓
   Database      Normalization   K-means/    Speaker_A,
   Collection                    DBSCAN      Speaker_B, etc.
```

3. Identification Phase

```
New Audio → Feature Extraction → Cluster Prediction → Speaker ID
    ↓              ↓                ↓              ↓
  Input       14 Features      Model Predict    Confidence
```

Implementation Details

Enhanced Database Structure

Conversation Collection


```
{
  "document": "Speaker_A (user): Hello, how are you?",
  "metadata": {
    "session_id": "session_20241201_143022_123",
    "speaker": "Speaker_A",
    "role": "user",
    "is_gemma_mode": false,
    "has_audio_features": true,
    "cluster_id": 0,
    "clustered_speaker": "Speaker_A",
    "clustering_confidence": 0.85
  }
}
```


Audio Features Collection


```
{
  "document": "{\"features\": [0.123, 0.456, ...], \"feature_names\": [\"energy\", \"pitch_e
  "metadata": {
    "conversation_id": "session_20241201_143022_123_20241201_143022_123456",
    "session_id": "session_20241201_143022_123",
    "speaker": "Speaker_A",
```

```
    "feature_count": 14
  }
}
```

Clustering Algorithms

K-means Clustering

- **Advantages**: Fast, deterministic, good for well-separated speakers
- **Parameters**: Number of clusters (auto-determined using silhouette score)
- **Use Case**: When you expect a specific number of speakers

DBSCAN Clustering

- **Advantages**: Discovers natural clusters, handles noise
- **Parameters**: Epsilon (neighborhood size), min_samples
- **Use Case**: When number of speakers is unknown

Optimal Cluster Detection

The system automatically determines the best number of clusters using: - **Silhouette Score**: Measures cluster quality (-1 to 1, higher is better) - **Cluster Balance**: Ensures even distribution across speakers - **Sample Density**: Considers data availability for each cluster

Usage Guide

Initial Setup

```
cd program_files
python main.py
```

The system will: 1. Start collecting audio features automatically 2. Use basic speaker detection initially 3. Display "Unknown" speakers until clustering is trained

Training Unsupervised Clustering

```
# After collecting sufficient data (10+ audio samples)
python train_unsupervised_speakers.py
```

This will: 1. Analyze collected audio features 2. Determine optimal number of clusters 3. Perform K-means or DBSCAN clustering 4. Save clustering state for future use

Analyzing Clustering Quality

```
python train_unsupervised_speakers.py analyze
```

Shows: - Cluster distribution and balance - Quality metrics (silhouette score) - Recommendations for improvement

Smart Fallback Logic

Decision Tree

```
Is Enhanced DB Available?
  Yes → Is Clustering Trained?
      Yes → Use Unsupervised Identification
      No → Fallback to Advanced ML
  No → Fallback to Basic Detection
```

Confidence Thresholds

- **Unsupervised**: 0.6 (60% confidence required)
- **Advanced ML**: 0.6 (60% confidence required)
- **Basic**: Always available (no threshold)

Performance Optimization

- **Feature Buffering**: Accumulates 100 frames before averaging
- **Database Updates**: Every 100 frames to reduce I/O
- **Identification Cooldown**: Every 50 frames to balance accuracy/speed

Monitoring and Analytics

Real-time Display

```
 Hello, how are you?
    Speaker_A |   2 voice(s) |   Known: Speaker_A, Speaker_B
    Unsupervised speaker identified: Speaker_A (confidence: 0.85)
```

Database Statistics

```
# Check current status
python -c "from utils.enhanced_conversation_db import EnhancedConversationDB; db = EnhancedC
```

Clustering Metrics

- **Silhouette Score**: Cluster separation quality
- **Cluster Balance**: Distribution evenness
- **Sample Count**: Data sufficiency
- **Confidence Distribution**: Identification reliability

Configuration Options

Clustering Parameters

```
# In EnhancedConversationDB
min_clusters = 2
max_clusters = 10
silhouette_threshold = 0.3
```

Feature Collection

```
# In SpeakerDetector
feature_buffer_size = 100
db_update_cooldown = 100
identification_cooldown = 50
```

Confidence Thresholds

```
identification_confidence_threshold = 0.6
```

Advanced Features

Incremental Learning

- **Continuous Updates**: New data automatically improves clustering
- **Adaptive Thresholds**: Confidence thresholds adjust based on data quality
- **Cluster Evolution**: Speaker profiles update over time

Multi-Session Support

- **Session Persistence**: Clustering state saved between sessions
- **Cross-Session Learning**: Data from multiple sessions improves accuracy
- **Temporal Analysis**: Speaker patterns tracked over time

Quality Assurance

- **Outlier Detection**: Identifies and handles poor quality audio
- **Feature Validation**: Ensures extracted features are meaningful
- **Cluster Stability**: Monitors clustering consistency

Best Practices

Data Collection

1. **Diverse Conversations**: Include different speakers and topics
2. **Audio Quality**: Ensure clear microphone input
3. **Sufficient Duration**: Collect at least 10-20 audio samples per speaker

4. **Natural Speech**: Use normal conversation, not reading

Training

1. **Wait for Data**: Don't train until you have 10+ audio samples
2. **Try Both Methods**: Test K-means and DBSCAN for best results
3. **Monitor Quality**: Use analysis tools to check clustering quality
4. **Retrain Periodically**: Update clustering as new data becomes available

Usage

1. **Start Simple**: Begin with basic detection, let unsupervised learning develop
2. **Monitor Confidence**: Pay attention to confidence scores
3. **Adjust Thresholds**: Modify confidence thresholds based on your needs
4. **Regular Analysis**: Periodically check clustering quality

Future Enhancements

Planned Features

- **Emotion-Aware Clustering**: Separate speakers by emotional state
- **Context-Aware Identification**: Consider conversation context
- **Real-time Adaptation**: Update clusters during conversation
- **Multi-language Support**: Language-specific feature extraction

Advanced Algorithms

- **Hierarchical Clustering**: Multi-level speaker organization
- **Gaussian Mixture Models**: Probabilistic speaker modeling
- **Deep Learning**: Neural network-based speaker identification
- **Transfer Learning**: Leverage pre-trained speaker models

This unsupervised learning approach provides a much more intelligent and adaptive speaker identification system that learns from real usage patterns rather than requiring manual training data.