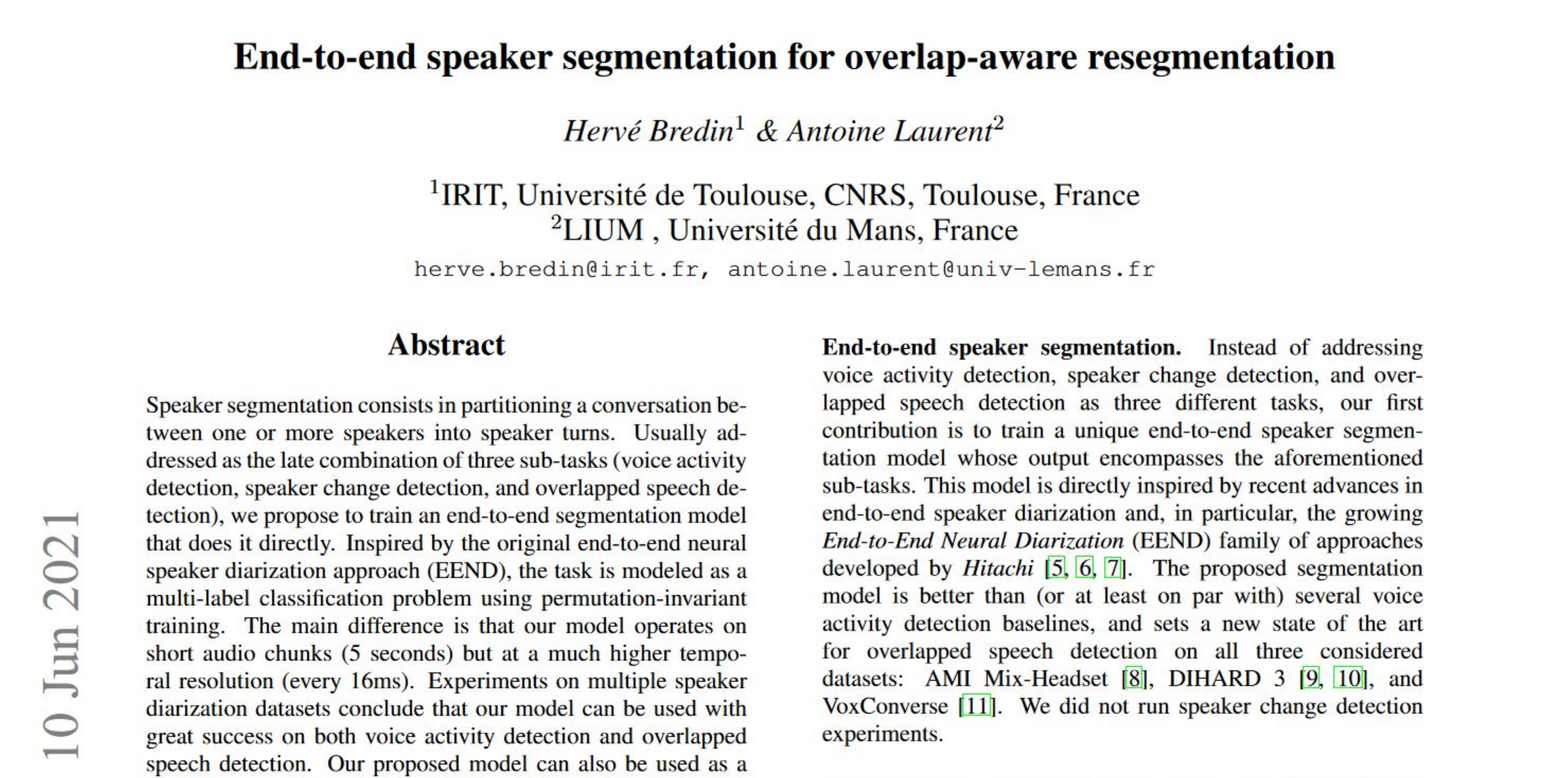
Speaker Change Detection

Hayden Housen

Literature Review



End-to-end speaker segmentation for overlap-aware resegmentation

Hervé Bredin[1] *& Antoine Laurent*[2]

[1]IRIT, Université de Toulouse, CNRS, Toulouse, France
[2]LIUM , Université du Mans, France

herve.bredin@irit.fr, antoine.laurent@univ-lemans.fr

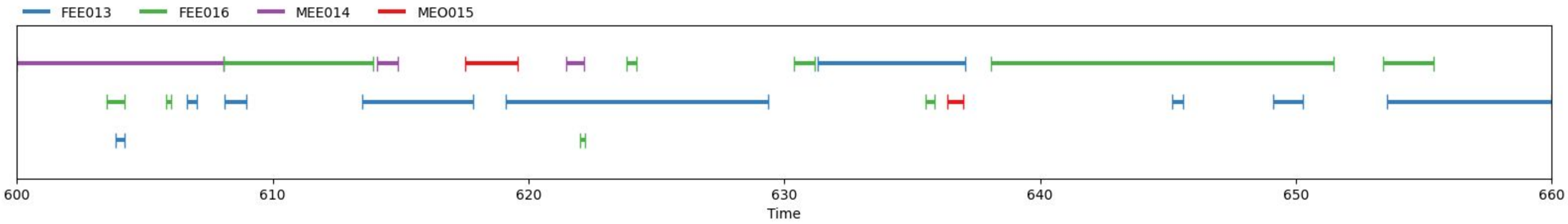
Abstract

Speaker segmentation consists in partitioning a conversation between one or more speakers into speaker turns. Usually addressed as the late combination of three sub-tasks (voice activity detection, speaker change detection, and overlapped speech detection), we propose to train an end-to-end segmentation model that does it directly. Inspired by the original end-to-end neural speaker diarization approach (EEND), the task is modeled as a multi-label classification problem using permutation-invariant training. The main difference is that our model operates on short audio chunks (5 seconds) but at a much higher temporal resolution (every 16ms). Experiments on multiple speaker diarization datasets conclude that our model can be used with great success on both voice activity detection and overlapped speech detection. Our proposed model can also be used as a

End-to-end speaker segmentation. Instead of addressing voice activity detection, speaker change detection, and overlapped speech detection as three different tasks, our first contribution is to train a unique end-to-end speaker segmentation model whose output encompasses the aforementioned sub-tasks. This model is directly inspired by recent advances in end-to-end speaker diarization and, in particular, the growing *End-to-End Neural Diarization* (EEND) family of approaches developed by *Hitachi* [5, 6, 7]. The proposed segmentation model is better than (or at least on par with) several voice activity detection baselines, and sets a new state of the art for overlapped speech detection on all three considered datasets: AMI Mix-Headset [8], DIHARD 3 [9, 10], and VoxConverse [11]. We did not run speaker change detection experiments.

Data

- AMI Meeting Corpus (DIHARD3, VoxConverse)
  - 100 hours of meeting recordings
  - English
  - Recorded in three different rooms with different acoustic properties
  - Mostly non-native speakers
  - Split into train, test, and development sets
  - Groundtruth: RTTM files (one speech turn per line with start time and duration)

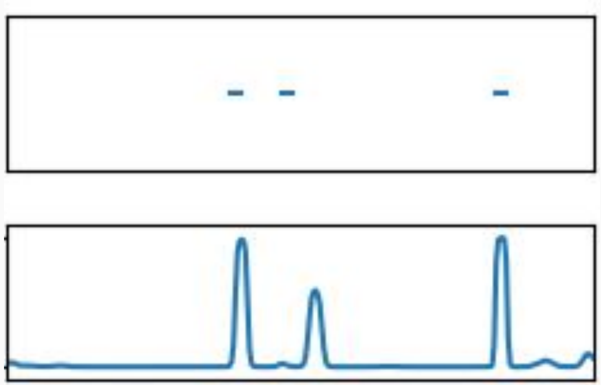


Ideas

- Process raw waveform in chunks of 5s
- Output scores per ≈17ms frame of audio
- **Speaker Change Detection:** scores = $\left\{ \begin{array}{ll} 1 & \text{speaker changed in frame} \\ 0 & \text{otherwise} \end{array} \right\}$
- **Segmentation:** scores = $$\begin{array}{cc} & \text{Frame} \\ \text{Active Speaker} & \begin{array}{c|c|c|c|c|c} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \;\cdots \end{array}$$
- Audio longer than 5s? Predict overlapping 5s chunks □ Aggregate chunks during inference

Idea Visualization

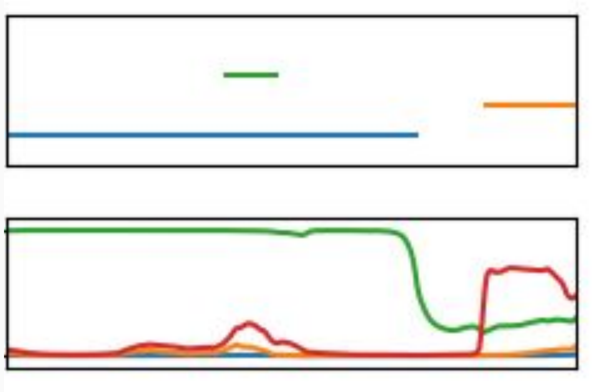
Speaker Change Detection



Ground Truth

Prediction

Segmentation



How to get speaker change points?

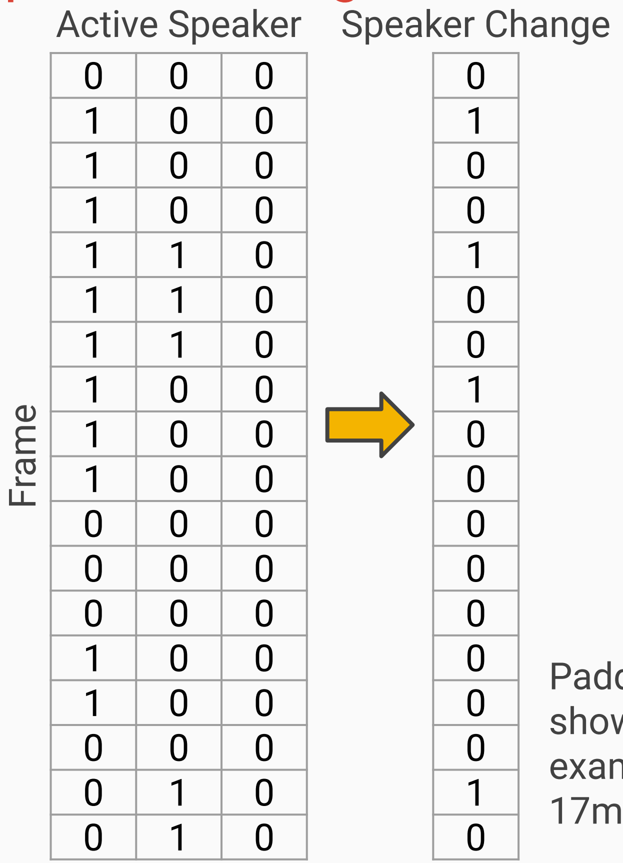
Threshold predictions

New speaker activation passes threshold

Data Preparation

- Load data using pyannote.database
  - Read RTTM files
  - Split into 293 frames
  - Other convenience functions
- Segmentation model outputs 4 speaker activations
  - Determined by scanning entire dataset and counting number speakers talking simultaneously. 99th percentile is 4 speakers.

Speaker Change Detection



Active Speaker			Speaker Change
0	0	0	0
1	0	0	1
1	0	0	0
1	0	0	0
1	1	0	1
1	1	0	0
1	1	0	0
1	0	0	1
1	0	0	0
1	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
1	0	0	0
1	0	0	0
0	0	0	0
0	1	0	1
0	1	0	0

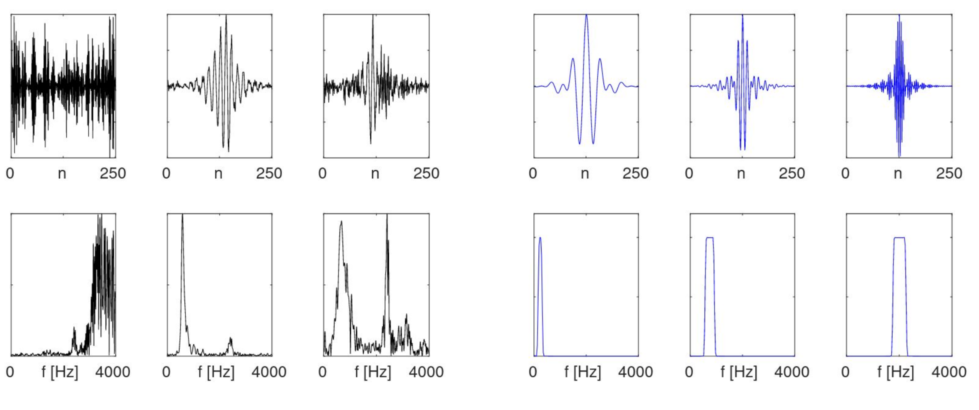
Padding not shown in this example.
17ms□100ms

Model Architecture

SincNet □ LSTM □ FC □ Classifier

Model: SincNet

- SincNet: Features from raw waveform using CNNs instead of hand-crafted features.
- First CNN layer deals with high-dimensional data and may learn strage filters.
- So, use pre-defined band-pass filters and only learn cutoff frequencies.
- Fewer parameters & faster convergence.



(a) CNN Filters

(b) SincNet Filters

Model: Other Components

- LSTM
    - Hidden size = 128
    - 4 layers
    - 50% dropout on first 3 layers
    - Bidirectional
- Linear
    - Hidden size = 128
    - 2 layers
- Classifier
    - Linear outputs number of classes
    - Sigmoid

Model: Pipeline

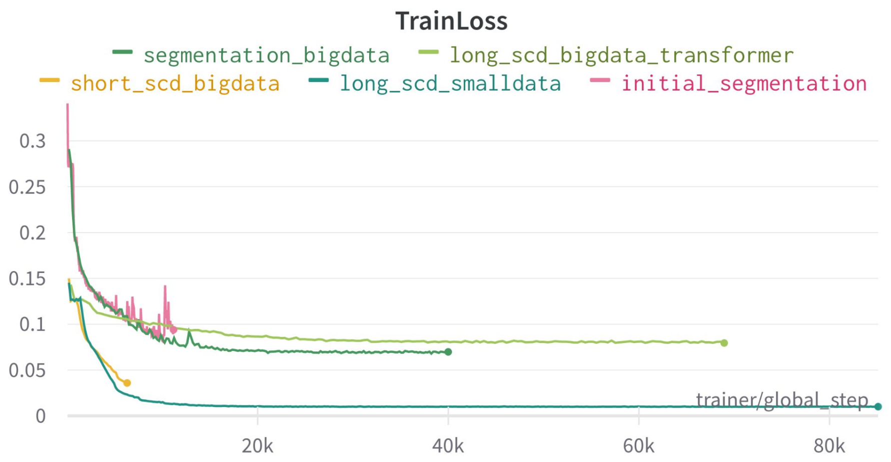
- Waveform: (channels=1, samples=80,000) [5s of audio at 16,000Hz]
- SincNet: (features=60, frames=293) [1 frame every ≈17ms]
- LSTM: (frames=293, features=256)
- Linear: (frames=293, features=128)
- Classifier: (frames=293, speakers=1 [if SCD] or 4 [if segmentation])

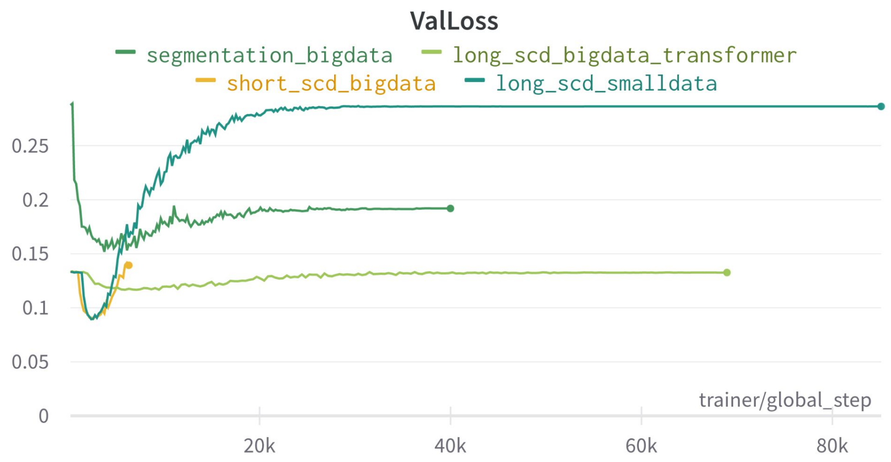
* Batch size not shown

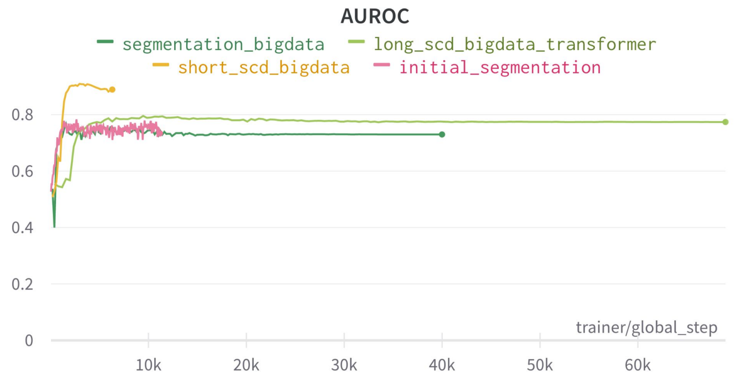
Training

- Loss Function: Binary cross entropy (permutation invariant version for segmentation)
- Adam optimizer with default parameters
- Learning rate initialized at 10^{-3}. Reduced by a factor of 2 when validation loss reaches a plateau for 12 epochs.
- Train DataLoader:
  - Randomly select file (weighted by duration)
  - Randomly select annotated region (weighted by duration)
  - Randomly select one 5s chunk uniformly
- Trained on Tesla T4 and RTX 3070 Mobile

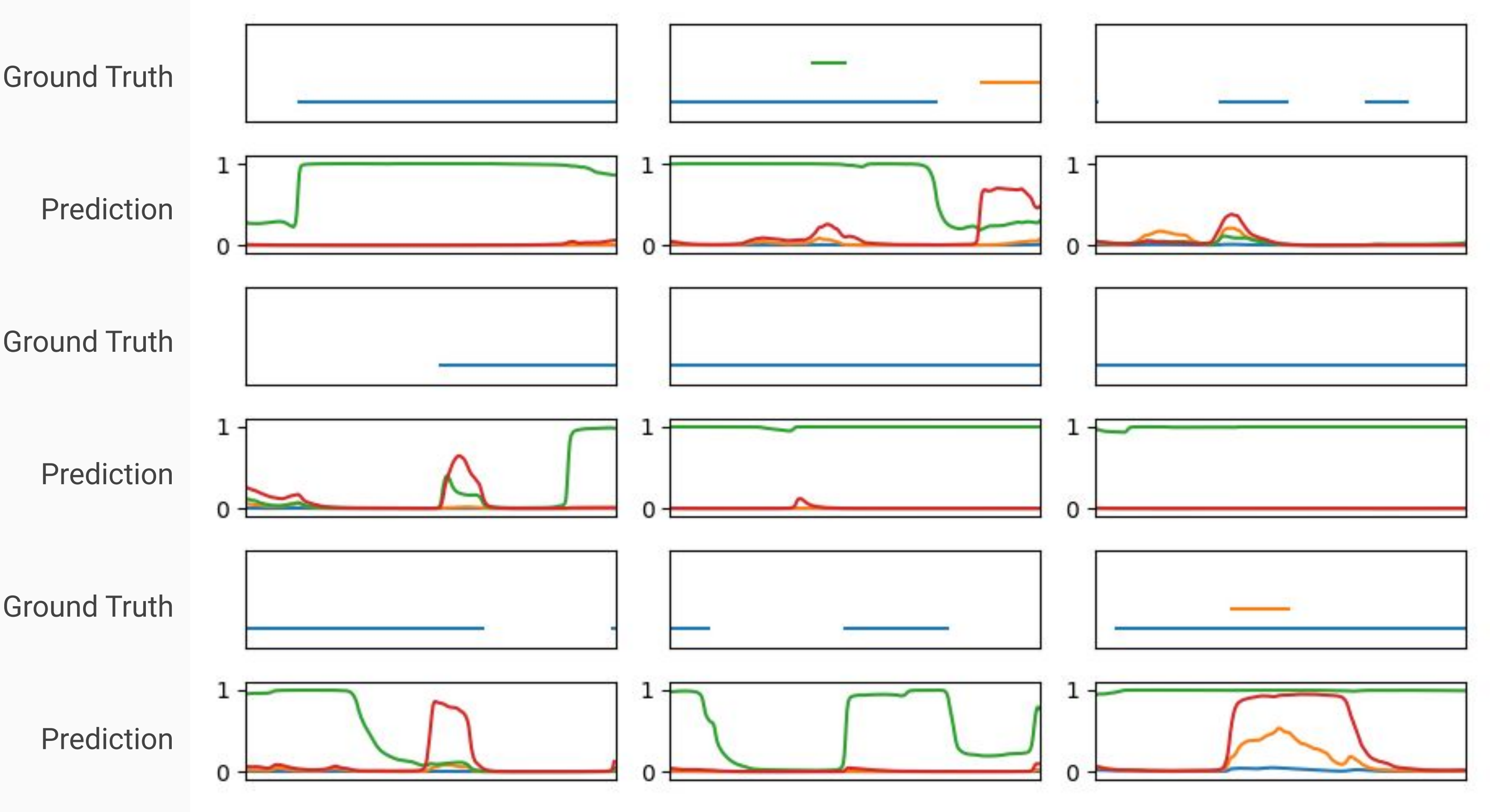
Training Charts



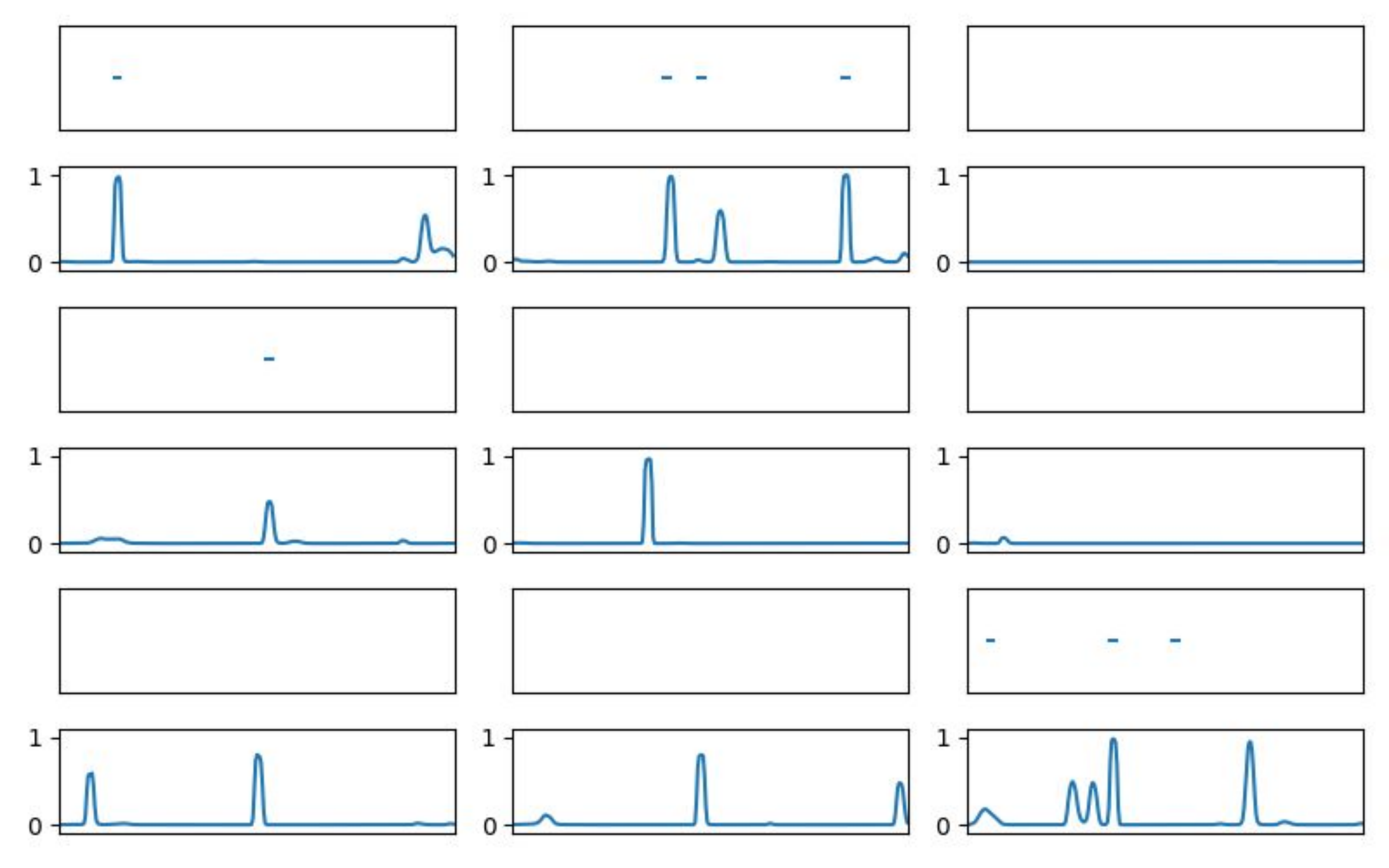




Training: Segmentation



Training: Speaker Change Detection



Ground Truth

Prediction

Ground Truth

Prediction

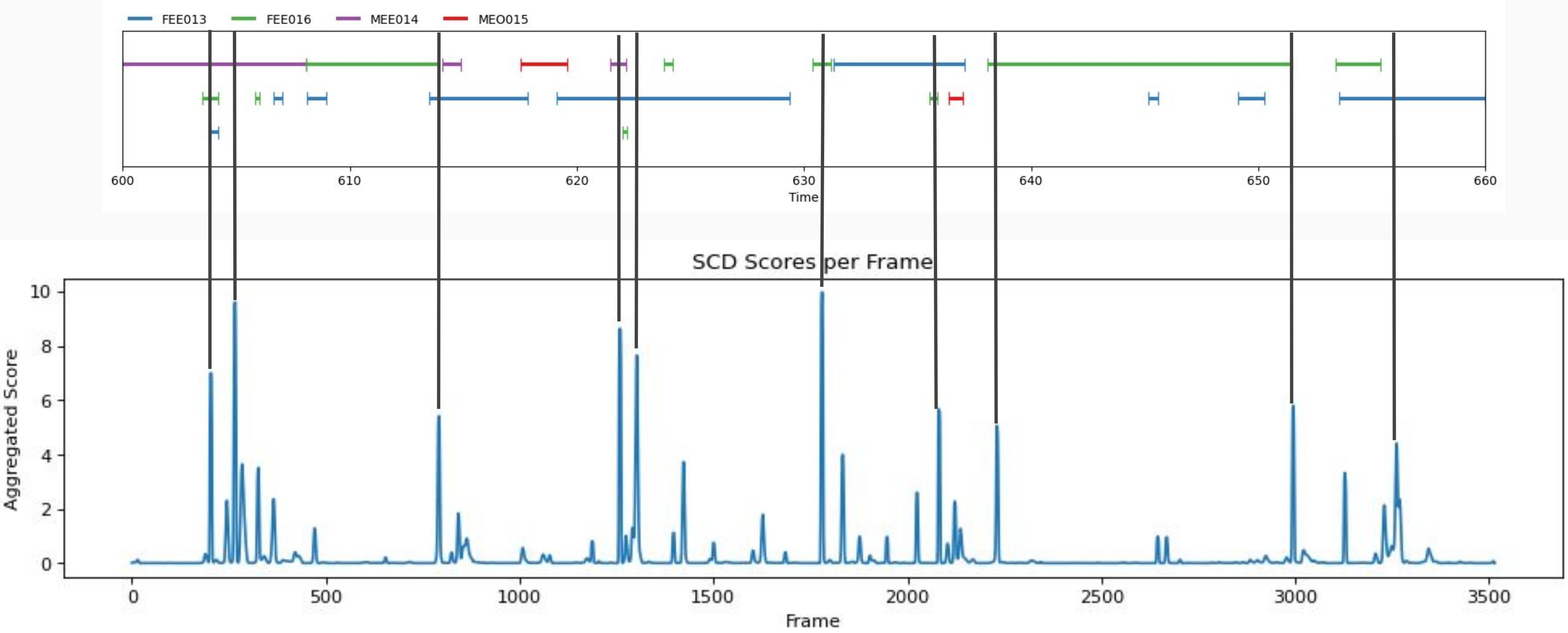
Ground Truth

Prediction

Inference

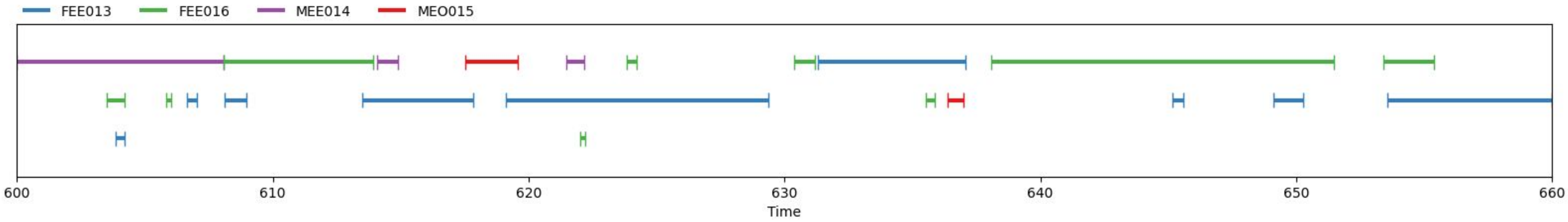
- Model accepts 80,000 samples, 1 minute is 960,000 samples
- Solution: Slide the model across the waveform and aggregate
  - 5s chunks, 0.5 seconds apart
  - Model predicts speaker activations for each chunk (293 output scores per chunk)
  - Make scores discrete (threshold at 0.5)
  - Segmentation only: Model may predict different activation for same speaker in different chunk. Activations that are approximately the same in two chunks get same label (by creating a graph).

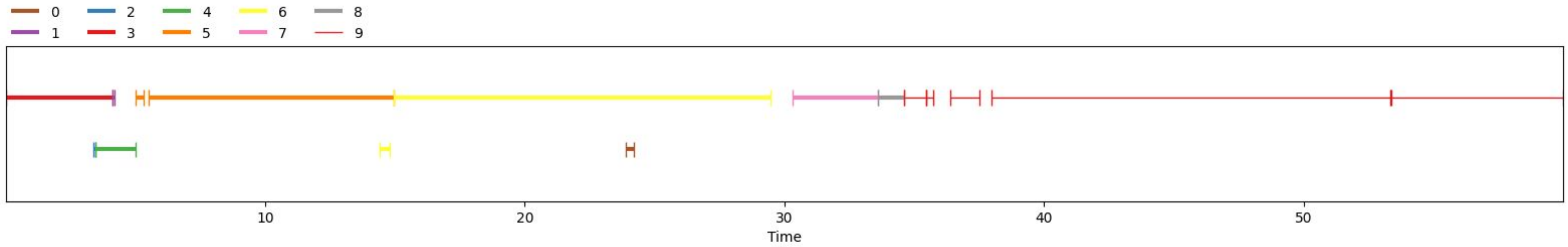
Inference: Speaker Change Detection

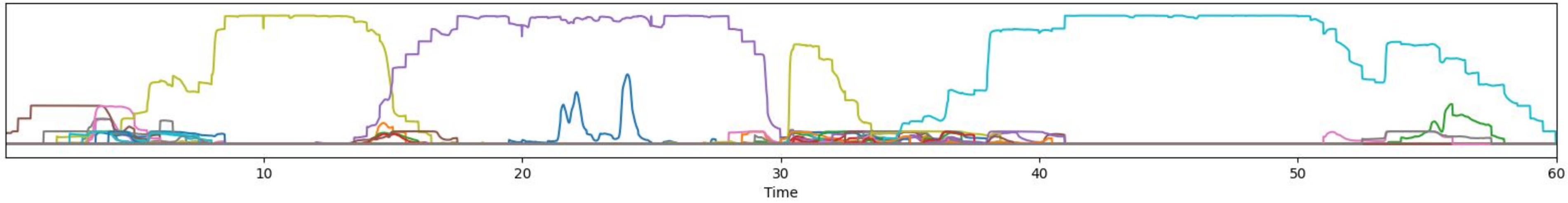


Change Points: 3.5, 4.5, 13.5, 21.5, 22.2, 30.4, 51.1, 55.7

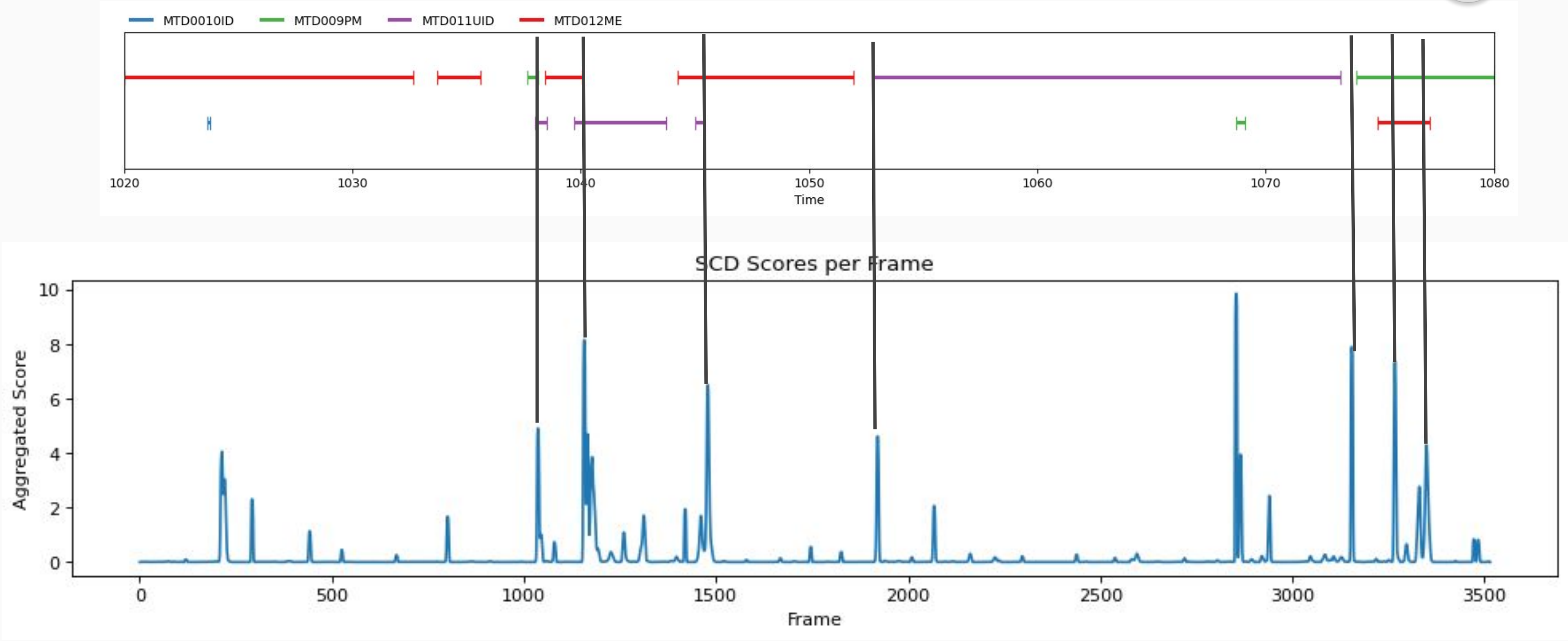
Inference: Segmentation







Similar Sounding Speakers Example



Further Research

- Train longer, larger model
- Data augmentation to address overfitting
- Tested transformers, but performed worse than LSTM.
  - LSTM ROC AUC ≈ 90 & Transformer ROC AUC ≈ 80
- SincNet supposedly better than handcrafted features, but should test MFCCs, FBANKs, etc.
- More advanced inference techniques:
  - Remove short gaps in active speaker output
  - Remove segments only active for short time
  - Separate activate and deactivate thresholds

References

- Primarily implemented “End-to-end speaker segmentation for overlap-aware resegmentation” by Hervé Bredin & Antoine Laurent
  - Code repo: pyannote/pyannote-audio on GitHub
- “Speaker Change Detection in Broadcast TV Using Bidirectional Long Short-Term Memory Networks” by Ruiqing Yin, Hervé Bredin, Claude Barras
- “Speaker Recognition From Raw Waveform With Sincnet” by Mirco Ravanelli, Yoshua Bengio