Speequal

Promoting Fair Conversations

Team

Christian Reimann

Jakob Koscholke

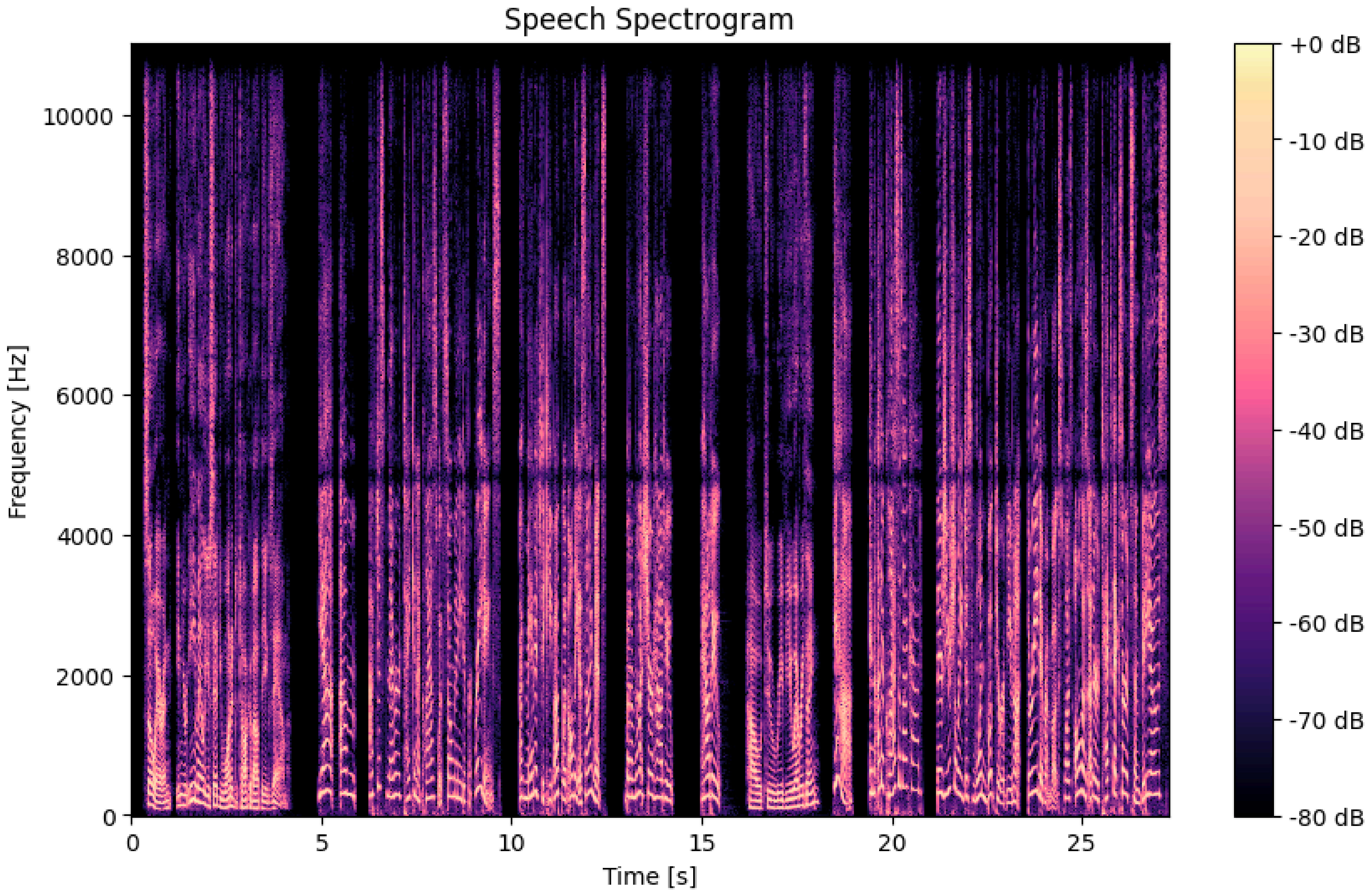
Data Product

1. **What?** App for monitoring **speaker-share**
2. **Why?** Promote **equitable** conversations
3. **Who?** Businesses, politicians, media professionals, teachers, family, friends

Challenges

1. **Zero-shot** problem
2. **Real-time** processing
3. **Model complexity** limits
4. Labelled data is **rare**

Our Data


Speech Spectrogram
Frequency [Hz]
10000
8000
6000
4000
2000
0
Time [s]
0
5
10
15
20
25
+0 dB
-10 dB
-20 dB
-30 dB
-40 dB
-50 dB
-60 dB
-70 dB
-80 dB

Credit where Credit is Due

arXiv:2109.06483v1 [eess.AS] 14 Sep 2021

OVERLAP-AWARE LOW-LATENCY ONLINE SPEAKER DIARIZATION BASED ON END-TO-END LOCAL SEGMENTATION

Juan M. Coria[1], *Hervé Bredin*[2], *Sahar Ghannay*[1], *Sophie Rosset*[1]

[1]Université Paris-Saclay CNRS, LISN, Orsay, France
[2]IRIT, Université de Toulouse, CNRS, Toulouse, France
[1]{juan-manuel.coria, sahar.ghannay, sophie.rosset}@lisn.upsaclay.fr
[2]herve.bredin@irit.fr

ABSTRACT

We propose to address online speaker diarization as a combination of incremental clustering and local diarization applied to a rolling buffer updated every 500ms. Every single step of the proposed pipeline is designed to take full advantage of the strong ability of a recently proposed end-to-end overlap-aware segmentation to detect and separate overlapping speakers. In particular, we propose a modified version of the statistics pooling layer (initially introduced in the x-vector architecture) to give less weight to frames where the segmentation model predicts simultaneous speakers. Furthermore, we derive *cannot-link* constraints from the initial segmentation step to prevent two local speakers from being wrongfully merged during the incremental clustering step. Finally, we show how the latency of the proposed approach can be adjusted between 500ms and 5s to match the requirements of a particular use case, and we provide a systematic analysis of the influence of latency on the overall performance (on AMI, DIHARD and VoxConverse).

Index Terms— speaker diarization, low latency, overlapped speech detection, speaker embedding

1. INTRODUCTION

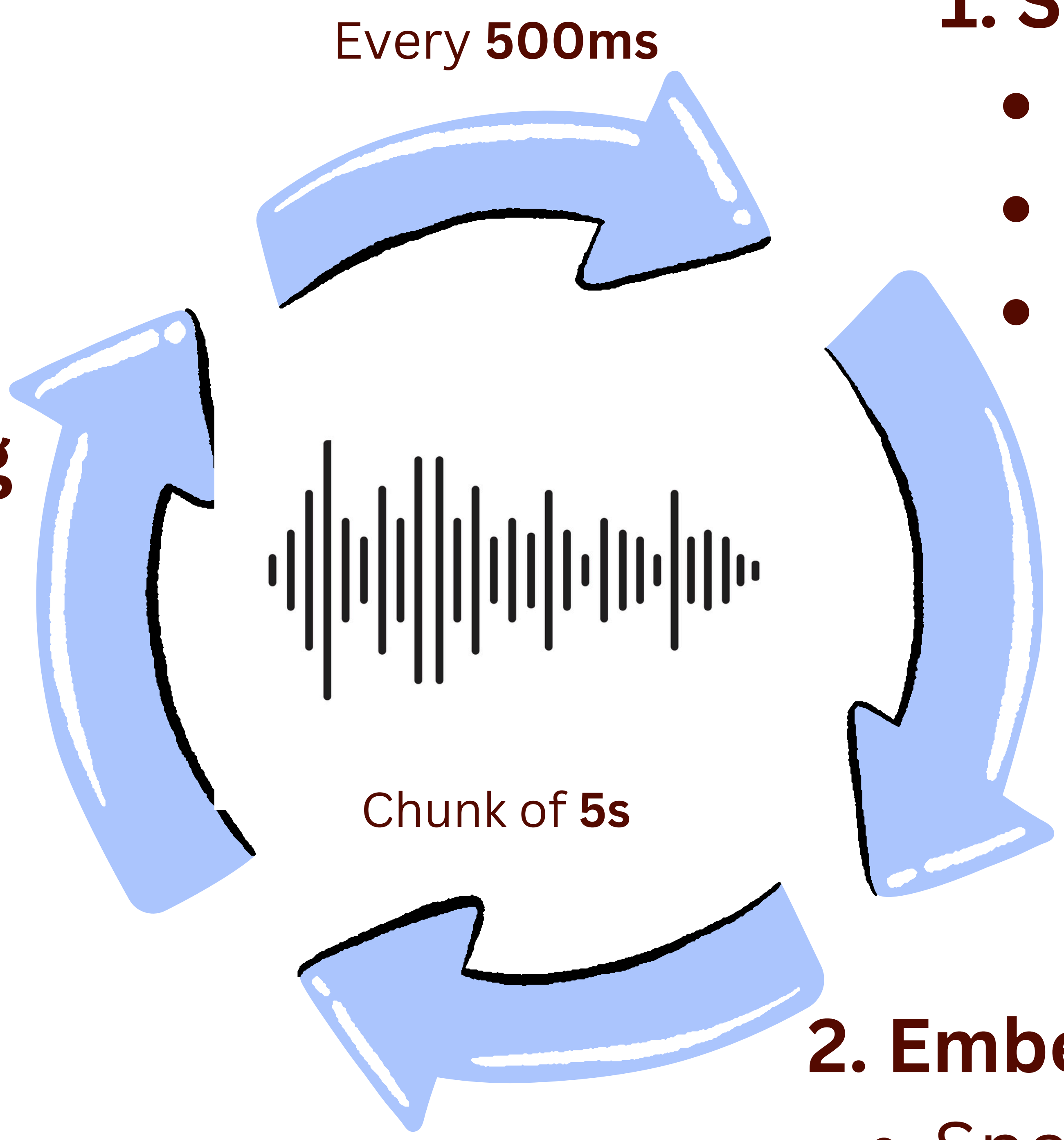
Speaker diarization aims at answering the question "who spoke when", effectively partitioning an audio sequence into segments with a particular speaker identity. Most dependable diarization approaches consist of a cascade of several steps [1, 2]: voice activity detection to discard *non-speech* regions, speaker embedding [3, 4] to obtain discriminative speaker representations, and clustering [2, 5, 6] to group speech segments by speaker identity. The main limitation of this family of *multi-stage* approaches relates to how they handle overlapped speech (which is known to be one of the main sources of errors): either they simply ignore the problem or they address it *a posteriori* as a final post-processing step based on a dedicated overlapped speech detection module [7, 8, 9, 10]. A new family of approaches have recently emerged, rethinking speaker diarization completely. Dubbed end-to-end diarization (EEND), the main idea of this approach is to train a single neural network – in a permutation-invariant manner – that ingests the audio recording and directly outputs the overlap-aware diarization output [11, 12]. We propose to meet half-way between *multi-stage* and *overlap-aware end-to-end* diarization and design a multi-stage pipeline where overlapped speech is a first-class citizen in every single step: from segmentation to incremental clustering. In particular, our first contribution (discussed in Section 2.2.1) is a modified version of the statistics pooling layer (initially introduced in the x-vector architecture) to give less weight to frames where the intial segmentation step predicts simultaneous speakers.

Despite being competitive with *multi-stage* approaches, the main limitation of the *overlap-aware end-to-end* approaches is the strong assumption that the number of speakers is upper bounded or even known *a priori*. While reasonable for some particular use cases (*e.g.* one-to-one phone conversations), this assumption does not hold in many other situations (*e.g.* physical meetings or conference calls). One solution to this problem is to augment *end-to-end* approaches with mechanisms to automatically estimate the number of speakers. For instance, EEND-EDA [13] extends EEND [11, 12] with a recurrent Encoder-Decoder network to generate a variable number of *Attractors* – similar to speaker centroids. *Multi-stage* approaches usually do not suffer from this limitation as they rely on a clustering step for which a growing number of techniques exist to accurately estimate the number of speakers [14]. We propose to combine *the best of both worlds* [15] by first applying the end-to-end approach on audio chunks small enough to reasonably estimate an upper bound on the local number of speakers and, then only, apply global constrained clustering on top of the resulting local speakers. As discussed in Section 2.2.2, we say that cluster-

This work was granted access to the HPC resources of IDRIS under the allocation AD011012177 made by GENCI, and was partly funded by the French National Research Agency (ANR) through the PLUMCOT project (ANR-16-CE92-0025). Thanks to Antoine Laurent for running and sharing the VBx offline speaker diarization *topline*.

Diarization Cycle

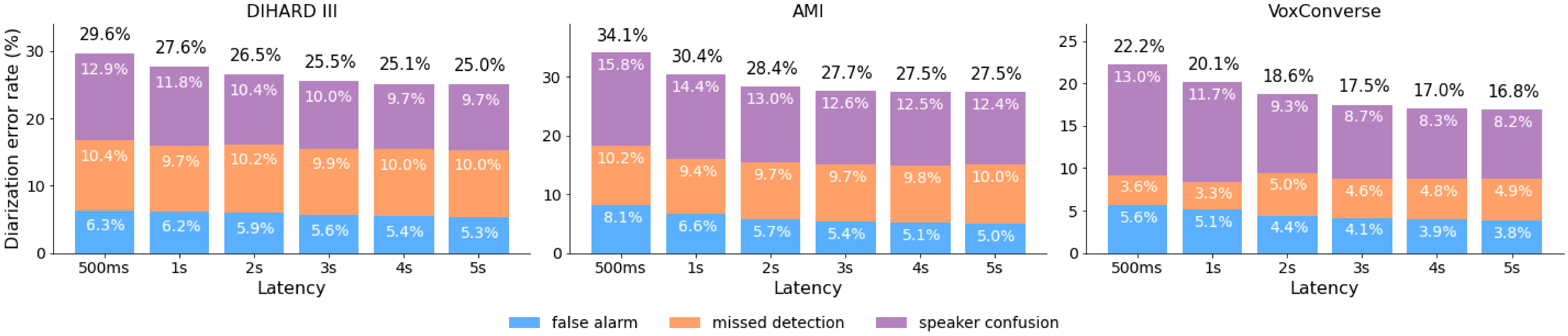
1. Segmentation
- Voice Activity Detection
- Speech Overlap Detection
- Speaker Change Detection



3. Incremental Clustering
- Similarity Detection
- Adding New Speakers
- Cluster Update

2. Embedding Extraction
- Speaker Embedding Extraction

Error vs. Latency



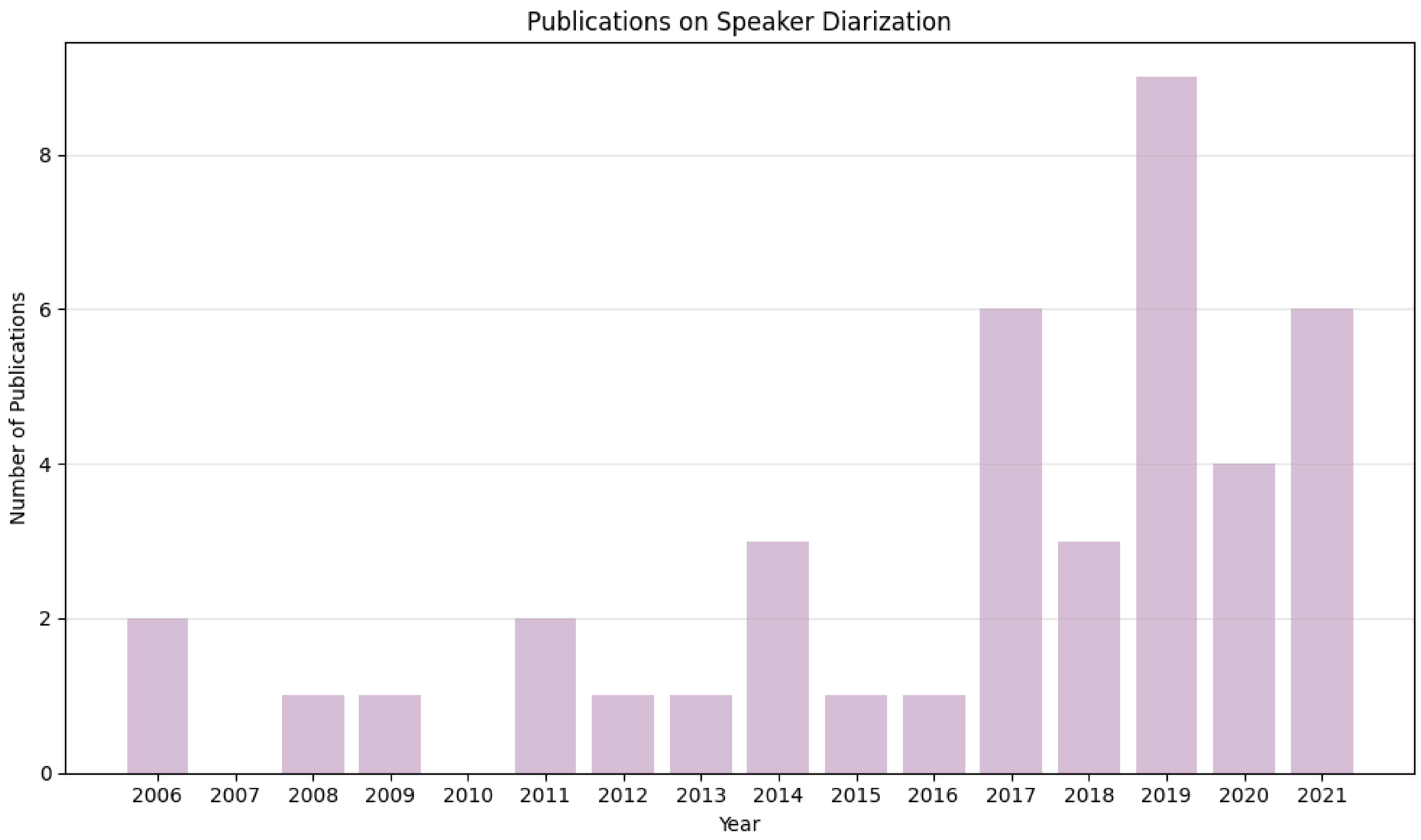
Live Demo

Open Challenges

1. Eliminate **indeterministic** behavior
2. **Hyperparameter**-optimization
3. Stream audio from **websocket**
4. Quantify **speech-overlap**

Further Reading

https://github.com/wq2012/awesome-diarization



Thanks!

TDNN for x-vectors

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	1500Tx3000
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512xN

Table 1. The embedding DNN architecture. x-vectors are extracted at layer *segment6*, before the nonlinearity. The N in the softmax layer corresponds to the number of training speakers.

Training Data

- Extracted from YouTube
- Over 7,000 speakers
- 1 million utterances
- 2,000 h material

Bias in VoxCeleb