DIART SYSTEM FOR THE EGO4D AUDIO-ONLY DIARIZATION CHALLENGE

Juan M. Coria and Sahar Ghannay

Université Paris-Saclay CNRS, LISN, Orsay, France

{lastname}@lisn.fr

ABSTRACT

In this report we describe the system from the *diart* team for the Ego4D Audio-only Diarization Challenge, consisting on determining "who speaks when" in short audio recordings. This system is the direct application of our *diart* python library for speaker diarization in real-time fine-tuned to the Ego4D dataset, and consisting of three modules: an end-to-end speaker diarization model, a speaker an overlap-aware embedding model based on x-vector, and an incremental clustering algorithm with adaptive speaker centroids.

Index Terms— diart, end-to-end speaker diarization, speaker embedding, real time

1. INTRODUCTION

The goal of the Ego4D [1] Audio-only Diarization Challenge is to determine *who speaks when* given a short audio recording extracted from an egocentric video clip. Most speaker diarization systems consist of a combination of modules addressing different related subtasks [2], like voice activity detection, speaker embedding [3, 4], speaker clustering [5], and overlapped speech detection [6]. However, recent work on end-to-end models trained with a permutation-invariant loss [7, 8] simplifies the approach by training a single model to receive an audio recording and output speaker activity probabilities through time.

Unfortunately, a disadvantage of both types of approaches is that they usually address *offline* speaker diarization, where the entire recording is needed beforehand to perform inference. Indeed, their latency and computational cost make them unsuitable for *online* speaker diarization. The system we describe in this report is based on previous work addressing the online scenario with low latency [9], where inference is applied progressively as the conversation takes place.

2. THE DIART SYSTEM

As mentioned previously, our system is based on previous work [9] and consists of multiple interconnected modules that allow the system to run online. We start by describing how end-to-end speaker diarization is leveraged for online decoding in Section 2.1, then we describe *overlap-aware* speaker embeddings in Section 2.2, and finally the online clustering algorithm in Section 2.3.

2.1. Speaker segmentation

First, we extract 5s audio chunks with a 500ms shift and feed them to a speaker segmentation model (*i.e. end-to-end speaker diarization*) [10] sequentially. This model, available in the huggingface [11] space[1], was trained on DIHARD 3 [12], AMI [13] and VoxConverse [14] on 5s chunks with a resolution of 16ms, and using a permutation-invariant loss based on the binary cross-entropy, hence allowing multiple speakers to be active at the same time frame.

Since we aim at online decoding, our system has access to a limited set of previous chunks, but any access to future chunks is prohibited. For each chunk, we obtain what we call a *local* diarization output. A naïve solution to obtain the diarization of the whole recording is simply to concatenate the all local outputs. However, given the nature of permutation-invariant training [7], two speakers in different chunks might activate the same index in the respective local outputs, or the same speaker might activate different indices. To solve this problem, we use speaker embedding and online clustering as a speaker tracking mechanism.

2.2. Overlap-aware speaker embeddings

In order to disambiguate local speakers and track them across audio chunks, we rely on *overlap-aware* speaker embeddings and online clustering. We use a pre-trained speaker embedding model based on the canonical x-vector architecture [4], also available in the huggingface space[2]. It has been trained using additive angular margin loss [15] on VoxCeleb1 [16] and VoxCeleb2 [17], reaching an equal error rate of 2.8% on the test set of VoxCeleb1. We modify this model at inference time to extract per-speaker embeddings from the same audio chunk by focusing on specific audio regions. Indeed, a 5s window limits embedding extraction for multiple speakers. For this reason, we modify the statistics pooling module in x-vector to compute weighted mean and weighted standard deviation features using their local segmentation. Moreover,

[1] as pyannote/segmentation@Interspeech2021

[2] as pyannote/embedding

in order to extract embeddings from the cleanest speech available, we decrease the local segmentation probability scores in low-confidence and high overlap-likelihood regions using the following equation:

$$\mathbf{w}_f = \left(\mathbf{s}_f \cdot \underset{k}{\operatorname{softmax}} \left(\beta \cdot \mathbf{s}_f\right)\right)^\gamma \tag{1}$$

where $\mathbf{w}_f$ are the weights used for speaker embedding extraction at frame f, $\mathbf{s}_f$ are the respective local segmentation scores at frame f, and k represents speakers. Note that β and γ are hyper-parameters that we manually set to 10 and 3 based on preliminary experiments.

2.3. Online clustering

We use a simple online clustering algorithm where embeddings are assigned a *global* speaker based on continually updated speaker centroids that are initialized to the first available embeddings. At each chunk, the system determines speaker assignments using the cosine distance between local speaker embeddings and centroids. However, if this distance is higher than a threshold δ_{new}, the local speaker is considered to be a new global speaker and a new centroid is created. Moreover, an additional threshold ρ_{update} on speech duration is used to decide which embeddings can be used to update centroids. Then, local segmentation scores are permuted according to the assignments and binarized using a threshold τ_{active}. Finally, the output of the system for a given chunk is the average first 500ms of the permuted and binarized scores. As in [9], this average is computed over all past outputs that contain the 500ms window, and it corresponds to a latency of 5s. In order to build the output for the entire recording, we simply concatenate all the 500ms output windows.

3. EXPERIMENTS AND RESULTS

Our results on the Ego4D v1 validation set are shown in Table 1. For the Ego4D system, we fine-tune all three hyper-parameters τ_{active}, ρ_{update} and δ_{new} on the v1 validation set. We evaluate using the diarization error rate (DER) computed with `pyannote.metrics` [18] without forgiveness collar and including all overlapping speech. On the Ego4D test set, our system achieves a DER of 53.5, outperforming the baseline system (based on VBx [2] and the Kaldi VAD [19]) by 11.8%.

System	DER	False alarm	Missed detection	Spk Confusion
VoxConverse	68.0	4.2	46.5	17.3
AMI	58.0	7.0	35.5	15.5
DIHARD III	57.8	6.0	37.6	14.3
Ego4D	55.8	7.0	35.3	13.5

Table 1. Results of our system with hyper-parameters tuned on different datasets. DER stands for Diarization Error Rate.

4. REPRODUCIBILITY

Our results can be reproduced using version 0.5.1 of the *diart* python library for real-time speaker diarization[3].

5. CONCLUSION AND LIMITATIONS

In this report, we describe the diart submission for the Ego4D Audio-only Diarization Challenge. With online decoding and a latency of 5s, our system is capable of outperforming the offline baseline by 11.8%.
Limitations: the system relies on multiple models that need to be trained beforehand, as well as on a considerable number of hyper-parameters that need to be tuned. In the future, we would like to work on alleviating these costs.

6. REFERENCES

[1] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al., "Ego4D: Around the World in 3,000 Hours of Egocentric Video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18995–19012.

[2] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget, "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, pp. 101254, 2022.

[3] Srikanth Madikeri, Ivan Himawan, Petr Motlicek, and Marc Ferras, "Integrating online i-vector extractor with information bottleneck based speaker diarization system," in *Proc. Interspeech 2015*, 2015.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[5] Mireia Diez, Federico Landini, Lukáš Burget, Johan Rohdin, Anna Silnova, Kateřina Žmolíková, Ondřej Novotný, Karel Veselý, Ondřej Glembek, Oldřich Plchot, Ladislav Mošner, and Pavel Matějka, "BUT System for DIHARD Speech Diarization Challenge 2018," in *Proc. Interspeech 2018*, 2018, pp. 2798–2802.

[6] Latané Bullock, Hervé Bredin, and Leibny Paola Garcia-Perera, "Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection," in *Proc. ICASSP 2020*, 2020.

[3] available at github.com/juanmc2005/StreamingSpeakerDiarization

[7] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe, "End-to-End Neural Speaker Diarization with Permutation-Free Objectives," in *Proc. Interspeech 2019*, 2019, pp. 4300–4304.

[8] Shota Horiguchi, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu, "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors," in *Proc. Interspeech 2020*, 2020, pp. 269–273.

[9] Juan M. Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset, "Overlap-Aware Low-Latency Online Speaker Diarization Based on End-to-End Local Segmentation," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 1139–1146.

[10] Hervé Bredin and Antoine Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech 2021*, 2021.

[11] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew, "HuggingFace's Transformers: State-of-the-art Natural Language Processing," *ArXiv*, vol. abs/1910.03771, 2019.

[12] Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman, "The Third DIHARD Diarization Challenge," *arXiv preprint arXiv:2012.01477*, 2020.

[13] Jean Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus," *Language Resources and Evaluation*, vol. 41, no. 2, pp. 181–190, 2007.

[14] Joon Son Chung, Jaesung Huh, Arsha Nagrani, Triantafyllos Afouras, and Andrew Zisserman, "Spot the Conversation: Speaker Diarisation in the Wild," in *Proc. Interspeech 2020*, 2020, pp. 299–303.

[15] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4685–4694, 2019.

[16] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[17] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[18] Hervé Bredin, "pyannote.metrics: A Toolkit for Reproducible Evaluation, Diagnostic, and Error Analysis of Speaker Diarization Systems," in *Proc. Interspeech 2017*, 2017, pp. 3587–3591.

[19] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.