Enhanced low-latency speaker spotting using selective cluster enrichment

Jose Patino
Department of Digital Security
EURECOM
Sophia Antipolis, France
patino@eurecom.fr

Héctor Delgado
Department of Digital Security
EURECOM
Sophia Antipolis, France
delgado@eurecom.fr

Nicholas Evans
Department of Digital Security
EURECOM
Sophia Antipolis, France
evans@eurecom.fr

***Abstract*—Low-latency speaker spotting (LLSS) calls for the rapid detection of known speakers within multi-speaker audio streams. While previous work showed the potential to develop efficient LLSS solutions by combining speaker diarization and speaker detection within an online processing framework, it failed to move significantly beyond the traditional definition of diarization. This paper shows that the latter needs rethinking and that a diarization sub-system tailored to the end application, rather than to the minimisation of the diarization error rate, can improve LLSS performance. The proposed selective cluster enrichment algorithm is used to guide the diarization system to better model segments within a multi-speaker audio stream and hence detect more reliably a given target speaker. The LLSS solution reported in this paper shows that target speakers can be detected with a 16% equal error rate after having been active in multi-speaker audio streams for only 15 seconds.**

***Index Terms*—low-latency speaker spotting, speaker detection, speaker diarization**

I. Introduction

Low-latency speaker spotting (LLSS), a form of biometric speaker recognition, was recently defined [1] in order to address the needs of the security and intelligence services to detect known speakers from multi-speaker audio streams *as soon as possible*. LLSS solutions are also relevant to consumer applications involving voice-based personal assistants and speaker-dependent, text-independent wake-up word detection.

Solutions to LLSS call for the closer integration of speaker detection [2] and diarization [3] technologies, traditionally separate tasks, and their combination within an online processing framework. The bulk of speaker detection research, e.g. [4]–[6], concerns the processing of single-speaker audio streams, or two-speaker telephone conversations, e.g. [7]. Many security and intelligence applications, in contrast, may involve audio streams containing multiple speakers. The application of speaker detection to multi-speaker data calls for some form of speaker segmentation and clustering, or speaker diarization. Speaker diarization is an enabling technology; it is rarely the end application itself. Past evaluation campaigns may not have provided the best platform to develop solutions that perform reliably when they are deployed in practical scenarios, including those involving speaker detection. When diarization solutions are combined with some form of speaker detection, then the use of the diarization error rate (DER) to optimise a diarization system will inherently lead to a suboptimal, combined system. Speaker detection and speaker diarization systems should be optimised in unison.

Whereas speaker detection systems are readily adapted to online processing, their performance tends to degrade rapidly when they are applied to speech segments of smaller duration [8]. Speaker diarization can be applied to cluster together short, same-speaker segments, thereby helping to protect performance, yet speaker diarization systems do not lend themselves well to online processing. While specific, online solutions have been proposed, e.g. [9]–[12], most have been developed as an enabling technology and without regard to an eventual end application.

Our first attempt to develop an efficient LLSS solution [1] took the first steps to unite the optimisation of speaker detection and speaker diarization technologies within a common online framework. While that work showed the potential, it failed to move significantly beyond the traditional definition of diarization. This paper aims to redefine the diarization problem such that the solution is more closely married to the core LLSS task. The approach exploits the use of the target speaker model (the one which is pre-trained for the speaker detection task) to guide diarization to cluster more reliably matching segments in the incoming audio stream. The process is referred to as selective cluster enrichment.

The remainder of this paper is organised as follows. Section II summarises the LLSS framework reported originally in [1]. Section III describes the new selective cluster enrichment procedure. Experiments are reported in Section IV. Conclusions are the focus of Section V.

II. Low-latency speaker spotting

The low-latency speaker spotting (LLSS) task was originally defined in [1]. Accordingly, only a brief summary is presented here.

This work was supported through funding from the Agence Nationale de la Recherche (French research funding agency) within the ODESSA (ANR-15-CE39-0010) project.

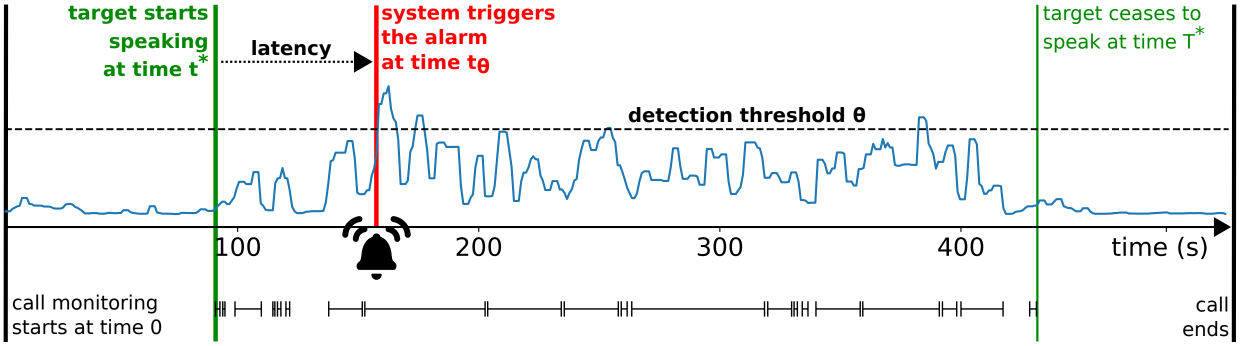

Fig. 1. An illustration of the low-latency speaker spotting (LLSS) task. Original illustration produced by H. Bredin, adapted and reproduced here with permission from [1].

A. Task definition

The LLSS task is illustrated in Figure 1. It shows a 500-second long audio stream which contains the speech of a known, target speaker. The target starts speaking at time t^*, a little under 100 seconds, and is then active during the segments illustrated to the bottom of the figure. The goal of LLSS is to detect the target speaker *as soon as possible* after t^*. In the example illustrated, the speaker is detected at t_θ, implying a detection latency of $t_\theta - t^*$.

A suitable solution to the LLSS problem involves the comparison of a log-likelihood function Λ (blue profile in Figure 1) to a threshold θ according to:

$$\Gamma(t) = \mathbb{1}\left(\max_{\tau\in[0,t]} \Lambda(\tau) - \theta\right) \tag{1}$$

where $\mathbb{1}$ is the Heaviside function which returns 0 for negative values and 1 for positive values and where the log-likelihood ratio is given by:

$$\Lambda(t) = \ln f(a_0^t|H_1) - \ln f(a_0^t|H_0) \tag{2}$$

where a_0^t is the speech from time $t = 0$ to t and $f()$ is a conditional probability density function given two competing hypotheses, namely that before time t the target speaker is either active (H_1) or inactive (H_0). Ideally, Λ should be less than θ for $t < t^*$ and greater than θ for $t \geq t^*$.

B. Metrics

Two LLSS metrics are reported in [1]. The first, referred to as the *absolute latency* refers to the difference:

$$\delta_\theta = \max(t_\theta - t^*, 0) \tag{3}$$

which is in the order of $50s$ for the example illustrated in Figure 1. The second metric reported in [1], and that used everywhere in this paper, is the *speaker latency* which takes into consideration only the time in $t_\theta - t^*$ during which the target speaker is active. The detection threshold θ needs to be set carefully in order to minimise latency while not triggering false alarms before the target is active, or not active in the audio stream at all.

In practice, and for a given database, the average speaker latency will increase monotonically with θ. Different values of θ will furthermore lead to different levels of detection performance. The detection performance, e.g. the detection cost $C_{det}(\theta)$ such as that used with the speaker recognition evaluations administered by NIST, e.g. [13], is typically convex in nature; low values of θ will produce too many false alarms whereas high values of θ will results in too many missed detections. A variable speaker latency may then be optimised according to the measure of detection performance. Alternatively, and as is the case for all work reported here, one can fix an application-dependent speaker latency δ, determine the maximum value of the likelihood function prior to $t^* + \delta$

$$\lambda_\delta = \max_{t\in[0,t^*+\delta]} \Lambda(t), \tag{4}$$

compare λ_δ to threshold θ and then measure detection performance according to some appropriate metric, e.g. $C_{det}(\theta)$ or, more simply, the equal error rate (EER).

III. Selective cluster enrichment

This section describes the adaptation of a diarization subsystem to the operation of a subsequent speaker detection subsystem.

Speaker diarization systems typically entail some form of segmentation and clustering process in order to determine the number of speakers within a multi-speaker audio stream and *who speaks when*. Generally, diarization is performed offline, meaning a diarization algorithm has access to the full audio stream before deriving a diarization hypothesis. In contrast, online diarization can be performed by processing an audio stream in segmental or sequential fashion and by updating the current diarization hypothesis to account for new speech data as it is encountered.

Be them offline or online, speaker diarizaton systems are usually evaluated using the classical DER which combines measures of background noise mistaken for speech, speech mistaken for background noise and speech assigned to the wrong speaker. In practice, one must strike a balance between under and over clustering. When the number of clusters is too

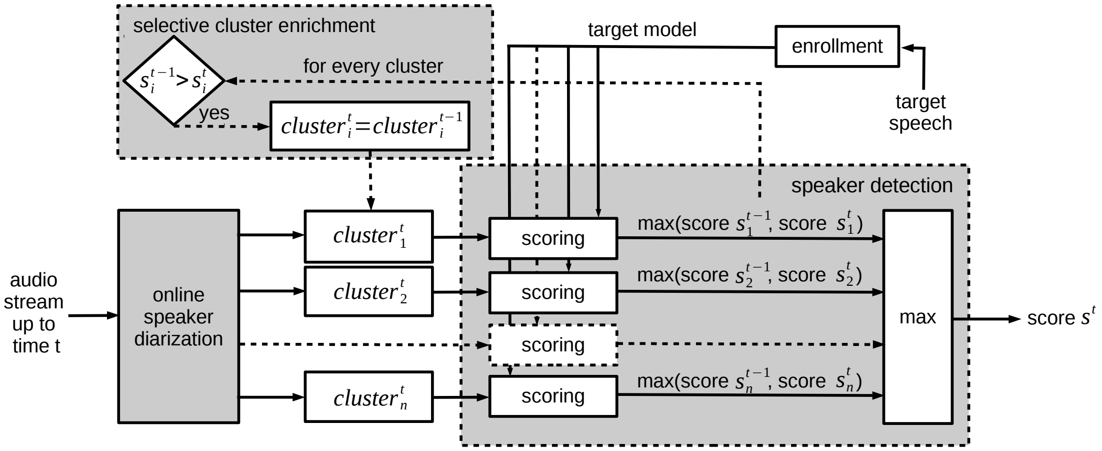
selective cluster enrichment
$s_i^{t-1} > s_i^t$
for every cluster
yes
$cluster_i^t = cluster_i^{t-1}$
target model
enrollment
target speech
speaker detection
audio stream up to time t
online speaker diarization
$cluster_1^t$
$cluster_2^t$
$cluster_n^t$
scoring
scoring
scoring
scoring
max(score s_1^{t-1}, score s_1^t)
max(score s_2^{t-1}, score s_2^t)
max(score s_n^{t-1}, score s_n^t)
max
score s^t

Fig. 2. An illustration of the low-latency speaker spotting solution that combines online speaker diarization with detection. Figure adapted from [1] to incorporate selective cluster enrichment.

high, i.e. greater than the number of speakers, then resulting clusters may have high purity – they are not contaminated excessively by the data of other speakers – by resulting models tend to be poorly trained using insufficient data [14]. In contrast, when the number of clusters is too few, models are comparatively well trained using more data, but purity decreases – inhomogeneous clusters are trained using data from multiple speakers. Somewhere in between, the balance between data quantity and impurity helps to minimise the DER or, as is the goal of the work reported in this paper, to optimise a more application-inspired metric.

The research hypothesis under investigation in the work reported here is that there is potential to guide the clustering process in a way that better balances data quantity and purity in order to improve the reliabilty of a subsequent speaker detection algorithm. This idea is explored within the context of a LLSS task which seeks to detect a particular target speaker for which a model is already trained and available. It seems logical in this case that the diarization process should at least make use of the target speaker model.

The original LLSS approach uses an online speaker diarization process that produces an evolving diarization hypothesis comprising n clusters cluster$_1^t$ ... cluster$_n^t$. Newly arriving data is assigned to the closest cluster in the current diarization hypothesis. The set of clusters are then scored against the target speaker models giving a set of scores s_1^t ... s_n^t. The maximum score among them is then compared to threshold θ in order to derive the detection decision $\Gamma(t)$.

The proposed modification is illustrated in Figure 2. The idea is to consider the target speaker model in the assignment of newly arriving data to one of the clusters in the current diarization hypothesis. The closest matching cluster is derived as before. In contrast to the original approach, though, the newly trained cluster for time t is replaced by the previous cluster for time $t-1$ if the new cluster score s_n^t is less than the previous cluster score s_n^{t-1}. The result is that the closest matching cluster is enriched with newly arriving data only if it improves the match between the cluster and the target speaker model. According to the max operation to the right of Figure 2, the set of scores s_1^t ... s_n^t will then be monotonically increasing with t. As before, the largest of these scores is then compared to threshold θ in order to derive the detection decision $\Gamma(t)$.

Even if the use of the target model at the heart of the diarization process is entirely intuitive, the motivation for the specific way in which it is used is far less intuitive. We attempt now to explain why its use in this way should lead to better LLSS performance. Selective cluster enrichment will have one of two effects. In the case that the closest cluster to newly arriving data match well the target model, then the process will serve only to purify the cluster, increase still further the match with the target model and improve LLSS performance. Other clusters that do not match well the target model can still only be enriched or adapted towards the target speaker model. In the case that the audio stream does not contain speech from the target speaker, then clusters will be either poorly trained using very little data, in which case diarization performance will deteriorate, or they will be adapted successfully towards the target, thereby degrading LLSS performance (since the speakers do not match). The hypothesis is that, even if clusters are inadvertently adapted to the target model, they will rarely be adapted sufficiently well such that the likelihood exceeds the detection threshold. In this case, the benefit of purifying matching clusters will outweigh the penalty of inadvertently adapting non-matching clusters. Accordingly, selective cluster enrichment should help to improve LLSS performance.

IV. Experimental work

This section describes experiments designed to evaluate LLSS performance and the benefit of selective cluster enrichment.

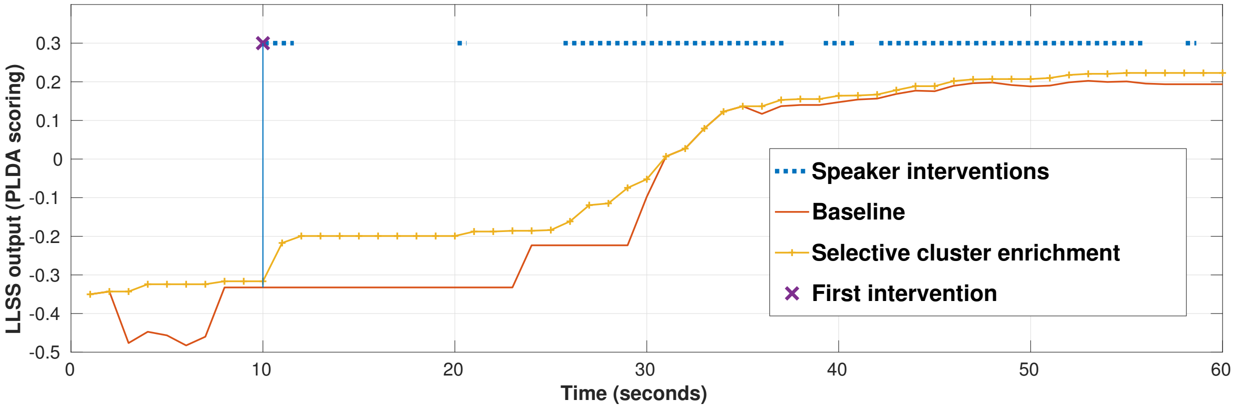

Fig. 3. An illustration of LLSS using PLDA scoring for an arbitrary trial utterance containing a target speaker. Profiles shown for the baseline and proposed solution with selective cluster enrichment.

TABLE I
LLSS PERFORMANCE ILLUSTRATED IN TERMS OF EER FOR DIFFERENT FIXED SPEAKER LATENCIES.

		GMM-UBM				i-vector			
	Latency	3	5	10	15	3	5	10	15
Baseline	Oracle	26.9	17	15.99	15.69	27.49	16.94	15.56	15.24
	Automatic	36.56	26.3	23.28	22.53	32.41	20.2	18.06	17.31
Proposed	Oracle	27.77	17.81	16.32	15.87	29.34	18.33	15.95	15.33
	Automatic	34.32	23.86	20.98	20.22	31.08	19.27	16.61	15.82

A. Database and protocol

The database used for all experimental work reported here is exactly the same as that used in the original LLSS work [1]. It is based upon the Augmented Multi-party Interaction (AMI) meeting corpus [15], a popular, publicly available database that is annotated at the speaker/segment level. All work reported here was performed not using the standard AMI protocols but, instead, protocols specifically designed to support the development and evaluation of LLSS. These have been made publicly available in order to support reproducibility[1]. Standard protocol training, development and evaluation partitions are still respected. Training data is used exclusively for background modelling. Speaker disjoint development and evaluation sets are both partitioned into enrollment and test subsets where enrolment data is used to train target speaker models. Full details of the protocol and its design are available in [1].

B. LLSS implementations

The performance of the selective cluster enrichment algorithm is compared to an LLSS baseline system with both oracle and automatic online diarization. The oracle system simulates the behaviour of an error-less, but still *online* system. The automatic online diarization system relies upon a long short-term memory (LSTM) based voice activity detector (VAD) [16]. Diarization is then performed using i-vectors [6] and an online sequential clustering algorithm. The system uses 19 MFCC coefficients as a frontend, a universal background model (UBM) of 256 components and a T matrix of rank 100, both learned from training data. Two speaker detection algorithms are used: a standard, 256-component Gaussian mixture model with universal background model (GMM-UBM) [4], and an i-vector system [6] with a T matrix of dimension 100 and probabilistic linear discriminant (PLDA) scoring [17]. Full details of both systems are available in [1].

C. Results

Results in Table I illustrate LLSS performance for the baseline and proposed solution, for both oracle and automatic diarization and for both GMM-UBM and i-vector speaker detection algorithms. Performance is expressed in terms of the equal error rate (EER) for various fixed speaker latencies: 3, 5, 10 and 15 seconds. On account of non-target models being poorly trained, results show that performance universally degrades for oracle diarization. However, results universally improve in the case of automatic diarization. This is due to improvements to target model purity stemming from selective cluster enrichment. While it is not the goal of this work to compare GMM-UBM and i-vector algorithms, it is reassuring that selective cluster enrichment improves the performance of both.

These observations confirm the hypotheses presented in Section 3. Further evidence is illustrated in Figure 3 which shows the evolution in PLDA scoring (Equation 3) for the baseline and proposed LLSS solutions for an arbitrary utterance that contains the target speaker during the intervals

[1] github.com/pyannote/pyannote-db-odessa-ami

indicated towards the top. The LLSS output for the proposed system is consistently higher than that of the baseline, showing that selective cluster enrichment serves to improve purity, forcing a monotonic increase in the score.

V. Conclusions

This paper shows how the performance of a low-latency speaker spotting (LLSS) solution can be improved by tailoring the operation of a speaker diarization sub-system to that of the following speaker detection sub-system. The proposed selective cluster enrichment scheme exploits the target speaker model to guide the diarization process in order to enhance the purity of matching clusters in the diarization hypothesis. The works serves to show that the optimisation of a diarization system on its own will never produce optimal results when diarization is only but one components of a more complex toolchain. Selective cluster enrichment will surely degrade the reliability of the diarization hypothesis when assessed with the traditional diarization error rate, but it nonetheless leads to more reliable speaker detection and LLSS performance. Universal improvements observed across two different speaker detection algorithms and a range of different speaker latencies show the potential for still further improvements using additional end-to-end optimisations. The latter is the subject of our ongoing work.

References

[1] J. Patino, R. Yin, H. Delgado, H. Bredin, A. Komaty, G. Wisniewski, C. Barras, N. Evans, and S. Marcel, "Low-latency speaker spotting with online diarization and detection," in *Proc. Odyssey*, June 2018.

[2] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, 2010.

[3] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 20, no. 2, pp. 356–370, Feb. 2012.

[4] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[5] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[6] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech and Language Proc.*, vol. 19, pp. 788–798, 2011.

[7] D. Reynolds, P. Kenny, and F. Castaldo, "A study of new approaches to speaker diarization," in *Proc. Interspeech*, 2009, pp. 1047–1050.

[8] B. Fauve, N. Evans, N. Pearson, J. Bonastre, and J. Mason, "Influence of task duration in text-independent speaker verification," in *Proc. Interspeech*, 2007.

[9] T. Oku, S. Sato, A. Kobayashi, S. Homma, and T. Imai, "Low-latency speaker diarization based on Bayesian information criterion with multiple phoneme classes," in *Proc. IEEE ICASSP*, Mar. 2012, pp. 4189–4192.

[10] W. Zhu and J. Pelecanos, "Online speaker diarization using adapted i-vector transforms," in *Proc. IEEE ICASSP*, March 2016, pp. 5045–5049.

[11] D. Dimitriadis and P. Fousek, "Developing on-line speaker diarization system," in *Proc. Interspeech*, 2017, pp. 2739–2743.

[12] G. Soldi, M. Todisco, H. Delgado, C. Beaugeant, and N. Evans, "Semi-supervised On-line Speaker Diarization for Meeting Data with Incremental Maximum A-posteriori Adaptation," in *Proc. Odyssey*, 2016, pp. 377–384.

[13] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, E. S. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2016 NIST Speaker Recognition Evaluation," *Proc. Interspeech*, pp. 1353–1357, 2017.

[14] N. Evans, S. Bozonnet, D. Wang, C. Fredouille, and R. Troncy, "A comparative study of bottom-up and top-down approaches to speaker diarization," *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 20, no. 2, pp. 382–392, 2012.

[15] J. 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.

[16] R. Yin, H. Bredin, and C. Barras, "Speaker Change Detection in Broadcast TV using Bidirectional Long Short-Term Memory Networks," in *Proc. Interspeech*, August 2017.

[17] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 413–417.