Active Correction for Incremental Speaker Diarization of a Collection with Human in the Loop

Yevhenii Prokopalo, Meysam Shamsi, Loïc Barrault, Sylvain Meignier, Anthony Larcher

applied sciences

MDPI

Article

Active Correction for Incremental Speaker Diarization of a Collection with Human in the Loop

Yevhenii Prokopalo*, Meysam Shamsi, Loïc Barrault, Sylvain Meignier and Anthony Larcher

LIUM, Le Mans Université, Avenue Olivier Messiaen, CEDEX 9, 72085 Le Mans, France; meysam.shamsi@univ-lemans.fr (M.S.); loic.barrault@univ-lemans.fr (L.B.); sylvain.meignier@univ-lemans.fr (S.M.); anthony.larcher@univ-lemans.fr (A.L.)
* Correspondence: Yevhenii.prokopalo@univ-lemans.fr

Abstract: State of the art diarization systems now achieve decent performance but those performances are often not good enough to deploy them without any human supervision. Additionally, most approaches focus on single audio files while many use cases involving multiple recordings with recurrent speakers require the incremental processing of a collection. In this paper, we propose a framework that solicits a human in the loop to correct the clustering by answering simple questions. After defining the nature of the questions for both single file and collection of files, we propose two algorithms to list those questions and associated stopping criteria that are necessary to limit the work load on the human in the loop. Experiments performed on the ALLIES dataset show that a limited interaction with a human expert can lead to considerable improvement of up to 36.5% relative diarization error rate (DER) for single files and 33.29% for a collection.

Keywords: speaker diarization; human assisted learning; evaluation

Citation: Prokopalo, Y.; Shamsi, M.; Barrault, L.; Meignier, S.; Larcher, A. Active Correction for Incremental Speaker Diarization of a Collection with Human in the Loop. *Appl. Sci.* **2022**, *1*, 0. https://doi.org/

Academic Editor: Francesc Alías

Received: 31 December 2021
Accepted: 04 February 2022
Published:

1. Introduction

Speaker diarization answers the question "Who speaks when?" within an audio recording [1,2]. Being important for audio indexing, it is also a pre-processing step for many speech tasks such as speech recognition, spoken language understanding or speaker recognition. For an audio stream that involves multiple speakers, diarization is usually achieved in two steps: (i) a segmentation of the audio stream into segments involving a single acoustic event (speech from one speaker, silence, noise...); (ii) a clustering that groups segments along the stream when they belong to the same class of event. A last step could be added to name the resulting speakers but this step is out of the scope of this paper.

Speaker diarization addresses the case of a single audio recording while it is often required to link the outputs of a speaker diarization system across consecutive audio recordings where speakers might appear in a recurrent manner. This use-case corresponds to the archiving of group meeting recordings or TV and radio shows. We will refer to the first scenario, addressing a single audio recording, as within-show speaker diarization and to the second one, addressing a collection of audio recordings, as cross-show speaker diarization.

When the sequence of shows is finite and the time constraint is not strong, cross-show speaker diarization can be done by concatenating all recordings and considering this task as a global process. However, for the case of long meeting series or TV/Radio shows, it is not reasonable to wait for the end of the collection process before processing the data. In this case, we will consider an incremental cross-show speaker diarization task that consists of processing one show after the other as soon as they are collected. Each show is processed and linked to the previous ones to incrementally extend a database of annotations. In this scenario, we will consider that annotations of a show which has been processed will not be modified once archived.

Modern diarization systems achieve decent performance depending on the type of data they process [3,4] but those performances are often not good enough to deploy such

systems without any human supervision [5,6]. Human assisted learning offers a way to achieve better performance by engaging an interaction between the automatic system and a human expert in order to correct or guide the automatic diarization process [7,8]. Amongst the different modes of human assisted learning, our work focuses on active learning where the automatic system, while processing an incoming sequence of audio recordings, is allowed to ask simple questions to the human expert [9]. In a previous work, we proposed a human-assisted correction process for within-show speaker diarization [10]. Here, we provide more analyses on this work and extend this approach by considering the case of incremental cross-show speaker diarization also including a human active-correction process.

We propose in this study a system architecture depicted in Figure 1. Given a current show and an history of already processed shows, the human assisted speaker diarization (HASD) system first produces an hypothesis based on which a questioning module sends a request to the human expert. The expert's answer is taken into account to correct the current hypothesis and possibly adapt the diarization system. This process iterates until reaching a stopping criteria out of those three: (i) the system has no more questions (ii) the human expert stops answering, or (iii) a maximum interaction cost is reached.

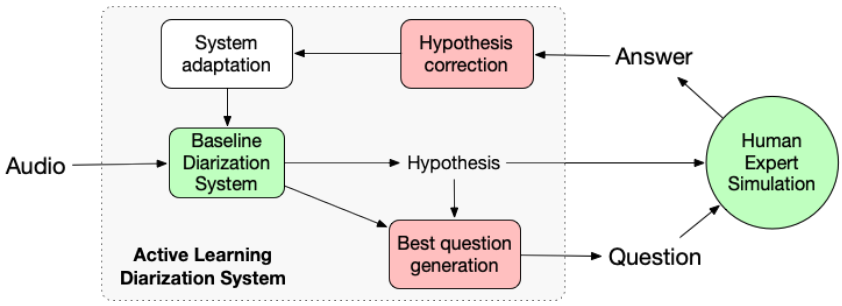


Figure 1. Life-cycle of a human-assisted cross-show speaker diarization system.

In this work, we define a binary question that allows a user/system interaction. For the stage of within-show diarization, we propose two questioning methods with the associated correction module and one method for the stage of cross-show diarization. The scope of this paper does not encompass the integration of both stages nor the system adaptation that will be studied in a future work.

Section 2 describes the related works. Corpora and evaluation protocols, developed in the context of the ALLIES project are detailed in Section 3. Baseline systems for within-show diarization together with their performance are provided in Section 4. We then propose our human-assisted approaches for within-show speaker diarization in Section 5. Section 6 describes the baseline cross-show diarization system while its active counterpart is proposed in Section 7. Finally, the outcomes and the perspectives of this study are discussed in Section 8.

2. Related Works

Within-show speaker diarization is a very active field of research in which deep learning approaches have recently reach the performance of more classic methods based on Hierarchical Agglomerative Clustering (HAC) [1], K-Means or Spectral Clustering [11] or variational-bayesian modeling [12]. Recent neural approaches have shown tremendous improvement for audio recordings involving a limited number of speakers [13–16]; however, the inherent difficulty of speaker permutation, often addressed using a PIT loss (permutation invariant training) does not allow current neural end-to-end systems to perform as well as HAC based approaches when dealing with a large number of speaker per audio file (>7) as explained in [17].

Literature on active learning for speaker diarization is very sparse and existing approaches are complementary to our work more than competitive. In [18], active learning is used to find the initial number of speaker models in a collection of documents. This

information is used to perform within-show speaker diarization without involving the human expert anymore. In [19], multi-modal active learning is proposed to process speech segments according to their length to add missing labels, task that is out of the scope of our study. Ref. [5] proposed an active learning framework to apply different types of corrections together with metrics to evaluate the cost of human-computer interactions.

Unlike the previous cited papers, in our work, one interaction with the human expert can lead to correct a whole cluster of segments (obtained with first of two clustering steps) instead of correcting a single segment only.

In [20], active learning is used to leverage training data and improve a speaker recognition system similar to the one we use for clustering. Active learning based approaches have been developed for other speech processing tasks including speech recognition [21–23], language recognition [24], speech activity detection [25] or speech emotion recognition [26] but are not directly applicable to speaker diarization.

Active learning literature for clustering is much wider [27,28] but mostly focuses on K-means clustering [29,30] or spectral clustering [31,32]. Hierarchical agglomerative clustering, that is used in many speaker diarization systems including our baseline, has also been studied for semi-supervised clustering [33,34]. Those studies propose to use predefined constraints to modify the clustering tree. In our work, instead of modifying the dendrogram, we propose a dynamic approach to update the threshold used to merge and split the clusters.

Regarding evaluation of the active learning process, multiple approaches have been proposed [5,35]. In [5], systems are evaluated by DER together with an estimate of the human work load to correct the hypotheses. We make the choice to use a penalized version of DER described in our previous work [36]. The human correction effort is computed to be in the same unit and thus added to the DER in order to provide a single performance estimator reflecting both the final performance and the cost of interacting with humans.

Cross-show speaker diarization is often considered as the concatenation of two distinct tasks including within show speaker diarization and speaker linking [37–39]. A few studies have considered cross-show speaker diarization as a whole [40,41] while, to our knowledge only [4] considers the case of incremental cross-show speaker diarization.

3. Evaluation and Corpus

Extending diarization to a human-assisted incremental cross-show speaker diarization task requires to extend metrics to take into account both the cost of human interaction and the incremental nature of the task. In this section, we first describe the metrics used to evaluate our baseline systems. Then we give a brief overview of the Penalized DER introduced in [36] and describe the computation of penalized DER for the case of incremental cross-show diarization. Finally, we describe the user simulation module developed to enable reproducible research in the context of human-assisted diarization.

3.1. Baseline Systems Assessment

Within-show speaker diarization results are reported using the weighted diarization error rate (*DER*) [42]. For incremental cross-show speaker diarization, each speaker from the system hypothesis is matched with a speaker from the reference just after processing the first show they appear in the collection. Once a couple of speakers have been associated, this link can not be changed in the future (remember, we do not allow re-processing of past shows). Speaker labels must be kept consistent across shows. The incremental cross-show DER is computed as the diarization error rate on the overall sequence of shows.

3.2. Human Interaction Assessment

Final performance assessment of a human assisted system must take into account the cost of human interaction in order to evaluate the quality of the interaction process. Penalized DER (DER_{pen}), a metric introduced in [10], is used to merge the information about the final performance (after human interaction) with the cost of the interaction

required to reach this result. This metric adds a constant amount of error time, called penalized time (t_{pen}), to the diarization error time, for each question asked to the human expert. Equation (1) defines the DER_{pen}.

$$DER_{pen} = \frac{FA + Miss + Conf + N \cdot t_{pen}}{T_{total}} \quad (1)$$

where FA is false time, $Miss$ is missed time, $Conf$ is confusion time of diarization hypothesis, N is the number of questions asked to the human expert and T_{total} is the total duration of audio files. For all experiments conducted in this paper, t_{pen} is set to 6 seconds. Although fixed empirically, this value has been chosen based on on-going work with human annotators from different companies; this time corresponds to the time required to listen two segments of 3 seconds each. An analysis of the value of t_{pen} has been produced in work to be publish soon. Incremental cross-show penalized DER is computed following the same formula in the incremental framework as explained earlier. We also propose to use the number of corrections over number of questions (CQR) ratio as a questioning performance criteria. It is used to evaluate the early stopping criteria and the question generation module for within-show speaker diarization.

3.3. User Simulation

To enable fair and reproducible benchmarking, a human expert is simulated by using ground truth reference to provide a correct answer to each question. In the context of this study, the user can ask questions of the form: "Have the segments A and B been spoken by the same speaker?". Since segments A and B might not be pure (i.e., they can include speech from several speakers), each segment is first assigned to its dominant speaker in the reference. The dominant speaker of a segment S that spreads between t_{start} and t_{end} is the one with maximum speech duration in the interval $[t_{start}, t_{end}]$ in the reference segmentation. Eventually, the user simulation answers the question by comparing the dominant speakers from segments A and B. This simulation is provided as part of the ALLIES evaluation package (https://git-lium.univ-lemans.fr/Larcher/evallies, accessed on 09-02-2022).

3.4. The ALLIES Corpus

Experiments are performed on the ALLIES dataset (Database and protocols will be made publicly available after the ALLIES challenge), an extension of previously existing corpora [43–45], that includes a collection of 1008 French TV and Radio shows partitioned in three non-overlapping parts whose statistics are provided in Table 1.

Table 1. ALLIES dataset description, all durations are given in hh:mm:ss, The *Recurrent speaker ratio* is the number of the speakers encountered in a show who have already been seen at least once in the past (in a show with older recording date) to the total number of speakers.

Partition	Duration	#speaker	#shows	Recurrent Speaker Ratio
Training	127:15:11	2384	273	34.8%
Dev	100:39:37	1212	362	52.9%
Eval	102:04:51	1284	373	50.6%
Total	329:59:39	5901	1008	49.4%

The training set is used to train the x-vector extractor and the PLDA model while the Development set is used to estimate the optimal clustering threshold. Performance is reported on the *Eval*uation set.

4. Automatic Within-Show Diarization Baseline Systems

This section describes the automatic systems used in this study for within-show speaker diarization and their performance.

4.1. Baseline System Description

To provide a fair study, all experiments are performed with two baseline automatic diarization systems. Both systems perform the diarization in two steps: a segmentation process, that splits the audio stream into (possibly overlapping) segments and a clustering process, that groups the segments into clusters: one cluster per speaker.

The LIUM Voice Activity Detection (*VAD*) system is used to segment the audio stream by discarding non-speech segments (silence, noise, breathing, etc.). This VAD, based on stacked LSTM [3], is implemented in the S4D open-source framework [46]. The output of the network is smoothed by removing non-speech segments shorter than 50ms and speech segments shorter than 25 ms.

The clustering is then performed in four steps: (i) a first hierarchical agglomerative clustering (HAC) is performed on vectors of 13 MFCC using the BIC criteria [46]; (ii) a Viterbi decoding is then used to smooth the segment borders along the audio stream; (iii) *x*-vectors are extracted from each segment and averaged to provide a single x-vector per BIC-HAC cluster; (iv) a second (final) HAC clustering is done by using *x*-vectors. The distance matrix used for this clustering is computed using a PLDA scoring [47].

The only difference between both baseline systems lays in the *x*-vector extractor. The SincNet Diarization System (*SincDS*) uses the SincNet extractor described in Table 2. The dimension of the produced *x*-vectors is 100. The input of the *SincDS* model is 80 dimensional MFCC extracted on a sliding window of 25 ms with a shift of 10 ms. The ResNet Diarization System (*ResDS*) uses a Half-ResNet34 extractor (see Table 3) to produce embeddings of size 256. As input, the *ResDS* model takes 80 dimensional Mel filter bank coefficient vectors extracted every 10 ms on sliding windows of 25 ms. Both MFCC and Mel-spectrogram means and variances are normalized.

Training of both networks is performed using an Adam optimizer with a Cyclic Triangular scheduler and cycles of length 20 steps. The learning rate oscillates between 1e-8 and 1e-3. One epoch corresponds to 100 audio chunks for each training speaker. Batches of 256 chunks are balanced across speakers and data augmentation is performed by randomly applying a single transformation among: noise addition, reverb addition, compression (GSM, ULAW, MP3 or Vorbis coded), phone filtering and pass-band filtering. Time and frequency masking are then applied for each chunk. Both networks are implemented using the SIDEKIT open-source framework [48] while the remaining of the system makes use of S4D [46].

Table 2. Architecture of the SincNet *x*-vector extractor. Dropout is used for all layers except the Linear layers. The activation function for Convolutional and Fully Connected layers is LeakyReLu. (C, F, T, stand for Channels, Features, Time.

Layer Name	**Structure**	**Output ($C \times F \times T$)**
Input	-	$1 \times 80 \times T$
MFCC	SincNet [80, 251, 1] 1D-Conv [60, 5, 1] 1D-Conv [60, 5, 1]	
Conv1D-1	[512, 5, 1]	
Conv1D-2	[512, 3, 2]	
Conv1D-3	[512, 3, 3]	
Conv1D-4	[512, 1, 1]	
Conv1D-5	[1536, 1, 1]	
StatPooling		
Linear-1	[3072, 100]	100
Fully-Connected-1	[100, 512]	512
Fully-Connected-2	[512, 512]	512
Linear-2	[512, 659]	659
SoftMax		

Table 3. Architecture of the *x*-vector Half-ResNet34 extractor with 9.5M trainable parameters. Dropout is used for all layers except the Linear layers. The activation function for Convolutional and Fully Connected layers is LeakyReLu. The Squeeze-and-Excitation layer is abbreviated as SE.

Layer Name	Structure	Output ($C \times F \times T$)
Input	-	$1 \times 80 \times T$
Conv2d	3×3, stride=1	$32 \times 80 \times T$
ResBlock-1	$\begin{bmatrix} 3 \times 3, & 32 \\ 3 \times 3, & 32 \\ \text{SE Layer} \end{bmatrix} \times 3$, stride = 1	$32 \times 80 \times T$
ResBlock-2	$\begin{bmatrix} 3 \times 3, & 64 \\ 3 \times 3, & 64 \\ \text{SE Layer} \end{bmatrix} \times 4$, stride = 2	$64 \times 40 \times T/2$
ResBlock-3	$\begin{bmatrix} 3 \times 3, & 128 \\ 3 \times 3, & 128 \\ \text{SE Layer} \end{bmatrix} \times 6$, stride = 2	$128 \times 20 \times T/4$
ResBlock-4	$\begin{bmatrix} 3 \times 3, & 256 \\ 3 \times 3, & 256 \\ \text{SE Layer} \end{bmatrix} \times 3$, stride = 2	$256 \times 10 \times T/8$
Flatten	-	
Attentive Pooling	-	5120
Dense(Emb)	-	256
AAM-Softmax	-	7205

Applying two consecutive clustering steps makes the application of active correction more complex but removing one of the steps degrades the performance of the baseline system. Thus we chose to keep the two consecutive clustering steps but to only apply active correction at the second clustering step while considering the BIC-HAC clusters as frozen. This choice has the advantage of reducing the correction to a simpler HAC-tree correction process, it also reduces the possibility of having short segments and extracts a more stable *x*-vector from several segments. One drawback is that errors from the BIC-HAC clustering will not be corrected so that the purity of those clusters is thus very important.

4.2. Within-Show Baseline Performance

For each system and data set (Dev and Eval), performance of within-show diarization is given in Table 4. For each system, all parameters and thresholds are optimized to minimize DER on the development set and then used on the evaluation set.

Table 4. Performance of the two baseline within-show diarization systems on both Development (Dev) and Evaluation sets (Eval) when using the reference segmentation (Ref) or an automatic segmentation (VAD). The performance is given as a weighed average of within-show Diarization Error Rate (DER). DER is computed for each show and weighted according to the duration of the shows.

System	Segmentation	Within-Show DER	
		Dev	Eval
SincDS	Ref	17.77	13.38
	VAD	19.07	20.20
ResDS	Ref	14.12	10.63
	VAD	14.97	16.74

From this table, it is noticeable that the *ResDS* system strongly out-performs the *SincDS* system in terms of within-show DER. This is due to the highest quality of its speaker representations (*x*-vectors). Using an automatic segmentation degrades the within-show DER for all

systems but this degradation is more important on the evaluation set than on the development set. For instance, *ResDS* within-show DER increases by a relative 6% on the Dev set but 57% on the Eval set. We observe that our two baseline systems perform better on *Eval* than on *Dev* when using the Reference segmentation but that they face strong degradation when using an automatic segmentation. This can be due to an over optimization of the system hyper parameters on the *Dev* set when using an automatic segmentation or to a mismatch between *Dev* and *Eval* sets but we do not have any clue of that.

5. Human-Assisted Within-Show Diarization

The proposed Human-assisted Speaker Diarization (HASD) system is depicted in Figure 1 and includes four modules: a fully automatic baseline diarization system, a question generation module, a correction module and an adaptation module. This section describes the proposed question generation and hypothesis correction modules. The adaptation module is out of the scope of this work and will be considered in future work.

Given a single audio stream, within-show speaker diarization consists of producing a segmentation hypothesis, i.e., a list of segments and speaker IDs with each segment allocated to a single speaker ID (segments might overlap). Within-show speaker diarization errors can be due to errors in the segment borders or to a wrong label allocation. The latter error being the most harmful in terms of performance [5], this work only focuses on correcting labeling errors.

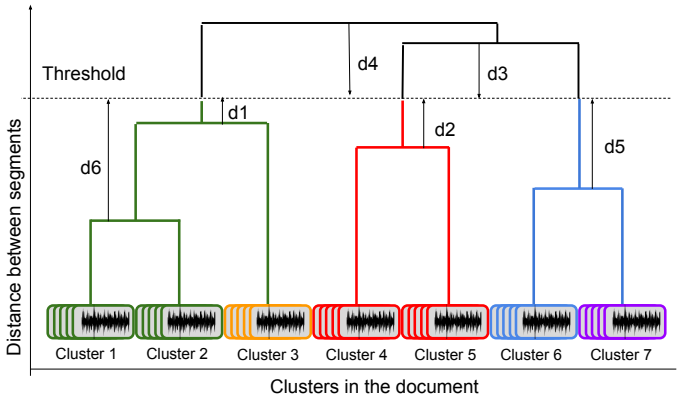
5.1. Within-Show Question Generation Module

When processing a single file, we propose to generate questions by following the architecture of the dendrogram produced by HAC (Figure 2a). HAC is done with no prior on the number of clusters so the threshold is empirically determined on the development set. This threshold separates the dendrogram in two parts (above and below the threshold). From this point, the same question can be asked to the human expert for each node of the dendrogram: *"Do the two branches of the node belong to the same speaker?"*. A "yes" answer from the human expert requires either to merge the two branches of a node above the threshold (merging operation) or, if the node is before the threshold, to leave the branches as they are (no splitting is required here). In case of a "no" answer, a node above the threshold would not be modified (no merging required) and the two branches of a node below the threshold would be separated (split operation).

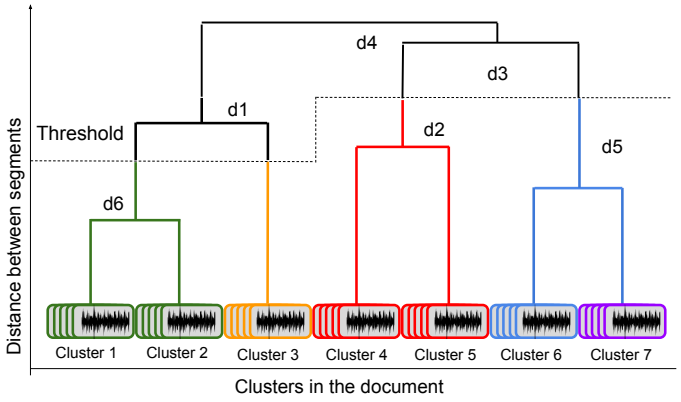
One must now determine which node to ask about and when to stop asking. To do so, we rely on the distance between the threshold and the nodes, referred to as *delta* to differentiate with distance between x-vectors. Examples of those *delta* are labeled $d1$ to $d6$ on Figure 2a. Nodes are ranked in increasing order according to their absolute *delta* value. We propose to ask questions about the nodes in this order, and consider two different stopping criteria.

First, a **two confirmation** ($2c$) criterion illustrated in Figure 2b, in which we assume that if a node above the threshold is confirmed by the human expert to be separated then other nodes above it, with higher *deltas* will not be investigated. Similarly, if one node below the threshold is confirmed by the human expert to be merged, the other nodes, lower in the dendrogram, will not be investigated.

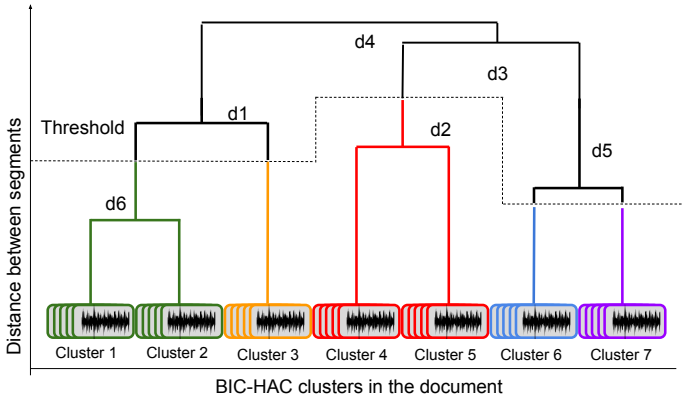
Second, we consider a per-branch strategy (*All*) to explore the tree in more details (see Figure 2c). Initially, a list of all nodes, according to their ranked *delta*, is prepared for investigation. Applying the ***All*** criterion, each time a question gets a confirmation (i.e., the human validates the system decision and no correction is required), the list of ranked nodes to investigate is updated by removing some of the following nodes as follows: (i) for a node lower than the threshold, if the human expert confirms the merge, its children nodes in the two branches will be removed from the list of nodes to investigate. (ii) for a node above the threshold, if the human expert confirms the split, its parents nodes will be removed from the list of nodes to investigate. For example in Figure 2c, $d4$ will not be investigated because $d3$ has been confirmed as a split.



(**a**) Initial HAC



(**b**) Stopping criterion: 2c



(**c**) Stopping criterion: All

Figure 2. The change of HAC dendrogram after interaction with human expert. In (**a**), the nodes located below the threshold (d_1, d_2, d_5 and d_6, ranked by increasing distance to this threshold) will be investigated for splitting correction, and the nodes located above the threshold (d_3 and d_4, ranked by increasing distance to this threshold) will be investigated for merging correction. The d_1 node in (**b**) (and d_1 and d_5 nodes in (**c**) has been modified according to the human correction.

The *2c* criterion relies on a high confidence in the *delta* ranking (the estimation of the distance between x-vectors) and strongly limits the number of questions, while the *All* criterion leads to more questions and thus a finer correction of the dendrogram.

To facilitate the work of the user answering the question, we consider that the HASD system proposes two audio segments (*samples*), for the user to listen to; one for each branch of the current node. Each branch can link several segments, even for nodes located at the very bottom of the tree (remember that, due to the sequential HAC clustering process, leaves of the dendrogram are clusters linked by the BIC-HAC clustering). The system must select the two most representative or informative *samples*. In a previous work [10], we investigated 4 *sample* selection methods:

Longest selects the longest segment from each cluster. It assumes that x-vectors from those segments are more robust and that the gain provided by the correction would lead to higher improvement of DER.
Cluster center selects the closest segment to cluster center assuming this is the best representation of this cluster. The center is selected according to the euclidean distance between segments' x-vectors.
Max / Min selects the couple of segments, one from each branch, with the lowest (max) or highest (min) similarity in terms of PLDA score (distance).
The results of our study [10] showed that Max/Min criterion are not competitive and we thus only focus on **Longest** and **Cluster center** in this work.

5.2. Performance of Within-Show Human Assisted Diarization

Figure 3 illustrates the evolution of DER for an audio file for both stopping criteria. As expected, *All* leads to more questions and achieve a better, lower, final DER than the *2c* criterion. This example shows the necessity of taking into account the cost of human interaction to fairly compare HASD systems.

A first set of experiments is performed to compare the Longest and Cluster center *sample* selection methods for both stopping criteria. Results are presented in Tables 5 and 6. The two sample selection methods are comparable in terms of number of questions asked per hour of audio and CQR. This is visible on the penalized DER which preserves the conclusions drawn by observing the DER.

Comparing the two stopping criteria, we observe that the *All* criterion succeeds to improve the DER more than the *2c criterion* for *SincDS* and *ResDS* models on the Development and Evaluation sets (see Tables 5 and 6). However, by taking into account the cost of question generation (for a t_{pen} empirically set to 6s), it can be concluded that the *2c* criterion presents a better compromise between DER improvement and human interaction cost, except when the longest (or resp. cluster center) segments are the representative of clusters and *ResDS* model is used on the Evaluation (resp. Development) set.

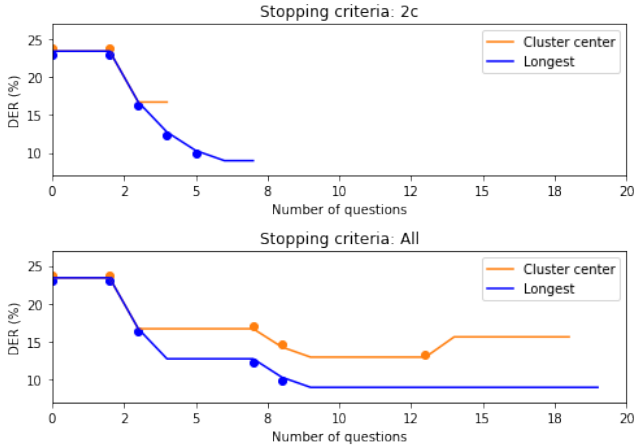


Figure 3. Tracking of the DER corresponding to a single show file (with duration of 42 min) by applying question-correction with different methods. Points in each question indicate that it has resulted in a correction.

Table 5. DER improvement using different stopping criteria and segment selection methods on the Development and Evaluation sets for the *SincDS* model using the reference segmentation.

Method	Stopping Criteria	Dev			Eval		
		DER	**Avg. #Q/h**	DER_{pen}	***DER***	**Avg. #Q/h**	DER_{pen}
Baseline	-	14.12	-	-	10.63	-	-
Baseline	-	17.77	-	-	13.38	-	-
Longest	*2c*	15.29	7.00	16.45	11.06	7.07	12.24
	All	14.29	25.55	18.54	9.96	27.16	14.4
Cluster center	*2c*	15.66	7.03	16.83	10.86	7.12	12.05
	All	15.19	25.60	19.45	9.82	27.17	14.39

Table 6. DER improvement using different stopping criteria and segment selection methods on the Development and Evaluation sets for the *ResDS* model using the reference segmentation.

Method	Stopping Criteria	Dev			Eval		
		DER	**Avg. #Q/h**	DER_{pen}	***DER***	**Avg. #Q/h**	DER_{pen}
Baseline	-	14.12	-	-	10.63	-	-
Longest	**2c**	12.78	21.65	16.39	9.42	24.02	13.42
	All	11.93	27.09	16.45	8.15	29.96	13.14
Cluster center	*2c*	14.56	21.29	18.11	9.42	24.17	13.45
	All	12.71	26.90	17.19	8.58	29.98	13.58

5.3. Analysis

In order to further improve our approach, we analysed the correlation between the benefit of human active correction (in terms of DER reduction or number of questions asked) and the characteristics of the processed audio files (number of speakers, duration of the file...). Based on the Pearson correlation coefficient, no strong correlation has been found (all less than 0.4).

The question generation module is inspired by the idea that closer nodes to the HAC's threshold reflect a lower confidence from the automatic system and thus should be questioned. In order to confirm this hypothesis, the benefits of successive questions for each show, has been checked in an ordinal way. The total DER improvement and CQR (number of corrections over the number of questions) are calculated based on the question order. For both stopping criteria, the first questions lead to larger DER reductions. This confirms that following the *delta* ranking as the sorting method for questioning nodes is a reasonable choice.

The proposed stopping criteria have been compared according to their average CQR per ranked question. Our observation shows that when using the *2c criterion*, successive questions keep contributing to the DER reduction while, for the *All* criterion, the CQR reduces. This demonstrates the expected behaviour where the *All* criterion leads to more questions and thus a finer correction, while the *2c* criterion limits the number of questions to assure higher benefits per question. The choice between these two methods can be guided by the available human resources.

6. Automatic Cross-Show Diarization Baseline System

This section describes the proposed cross-show speaker diarization baseline system and its results on both Development and Evaluation ALLIES partitions.

6.1. Baseline System Description

For incremental cross-show speaker linking, we assume that at time T, a number N of shows, $\{F_1, F_2, ...F_N\}$, has already been processed by the automatic system to produce a

cross-show diarization hypothesis including a set of M detected speakers, referred to as *known*-speakers. A database of x-vectors is built by including one single x-vector per *known*-speaker and per show, resulting in a collection of x-vectors: $\{S_1^1, S_2^1, S_3^1, S_4^2, S_5^2, S_1^3, ...S_M^N\}$, where S_j^i is the x-vector of speaker j observed in show i. Note that a single speaker k might have appeared in several shows F_n, F_m and thus be represented in the x-vector database by several x-vectors $\{S_k^n, S_k^m\}$.

When receiving a new show, F_{N+1}, the within-show speaker diarization is performed. All *new*-speakers, detected in this show, are then compared to the *known-speakers* from the existing database in order to detect recurrent speakers and link them with their previous occurrences in the collection. For each speaker α detected in the current show, all segments attributed to α in this show are processed to extract a collection of x-vectors that are then averaged to obtain one single x-vector S_α^{N+1} representing this speaker. As a result, the current file produces a set $\{S_\alpha^{N+1}, ..., S_\lambda^{N+1}\}$ of x-vectors.

For comparison of *known*-speakers and *new*-speakers, all x-vectors extracted from multiple shows for a same *known*-speaker are averaged to produce one single x-vector for this speaker. A pseudo-distance matrix is computed by comparing the *new*-speakers to the *known*-speakers using a PLDA model. A verification score is computed for each pair of *(known, new)*-speaker. The verification score is then multiplied by -1 and shifted so that the minimum of pseudo-distances is 0.

Values of this modified pseudo-distance matrix are processed in increasing order. Each value is compared to an empirical threshold (defined using a development set). If the pseudo-distance is lower than the threshold, then the corresponding couple of speakers is merged and both merged *known*- and *new*-speakers are removed from the linking process to avoid linking with others. Indeed, merging with another speaker would mean merging two *known*-speakers together or two new speakers together which is forbidden by our initial assumption. At the end of the process, each new speaker S_j^{N+1} merged with a *known*-speaker S_i^k is renamed accordingly (S_i^{N+1}) and her x-vector from the current file is added to the database. The database of *known-speakers* is then updated and used to process new incoming shows.

6.2. Cross-Show Baseline Performance

The proposed baseline cross-show speaker diarization method is applied on top of each of the two within-show baseline systems with the reference segmentation or an automatic segmentation. Performance is given in Table 7. The within-show averaged DER (weighed by the duration of the shows) is given for comparison in the fourth column.

The gap between *SincDS* and *ResDS* is more important that for within-show diarization. This is explained by the higher robustness of *ResDS* x-vectors compared to *SincDS* ones. One could be surprised by the fact that the DER obtained by the *ResDS* system on the *Development* set is lower when using an automatic segmentation than when using the reference. A first analysis shows that an automatic segmentation generates a higher number of speakers in the within show hypothesis. Each of these speakers having thus a shorter speech duration, the clustering errors that occur during the cross-show diarization affects less the final DER when a speaker is misclassified. The side effect of within show diarization errors on the cross-show diarization will be studied in further studies.

Table 7. Performance of the baseline cross-show diarization system applied on top of both within-show diarization baseline systems for both Development (Dev) and Evaluation sets (Eval). The performance is provided when using the reference segmentation (Ref) or an automatic segmentation (VAD) as a weighed average of within-show Diarization Error Rate (DER) and weighed average of incremental cross-show DER (i.e., computed for each show and weighed according to the duration of the shows).

System	Segmentation	Data	DER	
			Within	Cross-Show
SincDS	Ref	Dev	17.77	53.38
		Eval	13.38	51.33
	VAD	Dev	19.07	53.75
		Eval	20.20	53.85
ResDS	Ref	Dev	14.12	32.02
		Eval	10.63	30.43
	VAD	Dev	14.97	27.31
		Eval	16.74	31.13

7. Human Assisted Cross-Show Diarization

Once a within-show speaker annotation has been produced, the resulting speakers are linked to the already existing database of speakers from past shows. After this stage, we do not question the within-show segmentation anymore and only focus on speaker linking between current and past shows.

7.1. Cross-Show Question Generation Module

Similarly to the human-assisted within-show speaker diarization, our method only focuses on speaker linking (clustering) and does not modify the segmentation nor the clustering obtained during the stage of within-show diarization. During the incremental cross-show speaker linking process described in Section 6, the automatic system selects a couple of speakers (S_i, S_α^{N+1}) where S_i appeared in the past and S_α^{N+1} appeared in the current show. The human operator is then asked to listen to one speech sample from each speaker (S_i and S_α) and to answer the question: "Are the two speech samples spoken by the same speaker? "

The human-assisted cross-show diarization correction process differs from the within-show as no clustering tree can be used to define the question. In the cross-show scenario, we decompose the task into two steps:

1. **detection of recurrent speakers**, i.e., detect if a given speaker from the current file has been observed in the past;
2. **human-assisted closed-set identification** of speakers detected as *seen* during the first step. Speakers who have not been categorized as *seen* are simply added to the *known* speaker database.

To detect the recurrent speakers we propose to use a pseudo-distance matrix based on the one described in Section 6.1. The information conveyed in this matrix is the pseudo-distance between couples of x-vectors, the lower the pseudo-distance, the more likely two speakers are the same. This matrix is depicted in Figure 4.

This matrix can be computed in different manners depending on the representation chosen for *known*-speakers. Based on the idea that a *known*-speaker owns many x-vectors in each show where they appear, we consider here three speaker representations:

- The speaker can be represented by the average of her existing x-vectors (corresponding to all segments of this speaker in all already processed shows).

- A second representation, referred to as *Averaging* in the remaining of this study, consists of a set of one x-vector per file in which this speaker appears. The x-vector for a show being the averaged of all x-vector for this speaker in the given show.
- Eventually, we could keep, to represent a speaker, the set of all x-vectors belonging to this speaker (one x-vector per segment). We will later on refer to this last approach as *No averaging*.

Due to a high cross-show variability, we found that the first option is not optimal, probably because TV and radio shows have high acoustic variability for which a single average x-vector computed on all of them would not be a good representative of the speaker; for this reason we only focus in this article on two methods: *Averaging* of x-vectors per show and *No averaging*, where all possible x-vectors are kept for comparison.

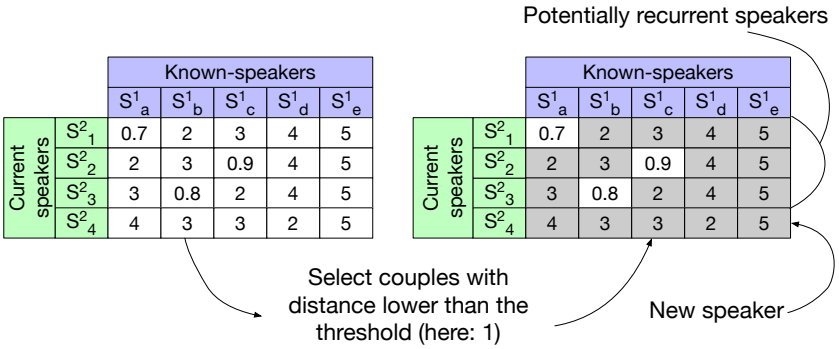


Figure 4. Detection of new and known speakers for the current show by comparing speakers from the current show with known-speakers from the existing speaker database. All speakers from the current show who do not obtain at least one verification score lower than a fixed threshold are labeled as new speakers; others are then considered for human-assisted closed-set identification.

The pseudo-distance matrix is computed by comparing each speaker single x-vector from the current show to all past-speakers *Averaging* and *No averaging* representations. A threshold, set empirically on the development set, is then applied on the pseudo-distances. If a speaker from the current show has no distance below the threshold (see Figure 4) it is categorized as new (never seen in the past). Other speakers are categorized as possibly recurrent and selected for the following closed-set speaker identification phase.

In a second step, human-assisted closed-set identification is applied for all speakers categorized as possibly recurrent. For each possibly recurrent speaker, x-vectors from all *known*-speakers (i.e., *Averaging* or *No averaging* representations of those speakers) are sorted by increasing pseudo-distance (Note that the number of those x-vectors depends on the chosen representation: *averaging* or *no averaging*). By increasing pseudo-distance, a binary question is asked to the human operator to compare the couples of speakers. This question makes use of one single audio segment selected for each speaker. Similarly to the within-show human-assisted process, the human operator is offered two audio segments (one belongs to the current speaker and one belongs to a known-speaker) to listen to: the longest for each speaker. Based on those two segments, the question asked to the human expert is: "Is the speaker from this segment (a known-speaker) the same as the one speaking in this current segment?".

If the operator answers "Yes", the two speakers are linked and the selected *known*-speaker is not proposed anymore to link with any other current speaker. If the operator answers "No", the next x-vector per order of pseudo-distance is considered. For one current speaker, the process ends either when linked with a *known*-speaker or after a number of questions chosen empirically; in the latest case, the current speaker is added to the *known* speaker database.

Cross-show acoustic variability can pollute x-vectors and cause errors in their pseudo-distance sorting. In other words, two segments with a higher acoustic mismatch can have lower pseudo-distance, which misleads the close-set speaker identification. We assume

here that the shows are homogeneous enough so that if the current speaker has appeared in past show (so that actually S_{α}^{N+1} is the same as S_x), then S_x pseudo-distance to S_{α}^{N+1} must be lower than other speakers in the current show. This assumption motives us to keep only the most similar speaker (the one with lower pseudo-distance) in each past show to include in the checking list. In order to find the best match of speakers cross-show (a speaker from current show and a speaker from past shows), two strategies are proposed: (i) *Nearest speaker per show*: in which one single speaker per show is included in the list (with respect to the pseudo-distance order) as candidate for matching with the current speaker. (ii) *All*: in which all speakers for all shows are ranked with respect to their pseudo-distance to the current speaker without limiting of number of speaker per show.

7.2. Performance and Analyses of Cross-Show Human Assisted Diarization

This section reports the performance of our human-assisted cross-show speaker linking for the two baseline systems (*SincDS* and *ResDS*) and the combination of both *Averaging* and *No averaging* speaker representations considering the ranking of *All* speakers or only the *Nearest speaker per show*.

All results are given in Figure 5. A first look at the Figure confirms that the *ResDS* system strongly outperforms the *SincDS* system due to the quality of the produced embeddings. We then observe that all four systems benefit from asking more questions to the human operator (improving final DER), and this for all speaker representations and when considering the entire list of speakers or only the nearest per show. According to the penalized DER (lightest bars), the benefit obtained when using the proposed active correc-tion method is higher than the cost of active correction (except for some cases in the *ResDS* system using VAD). Compared to the baseline systems, asking seven questions per current speaker allows to reduce the incremental cross-show DER by a relative 24% for the *ResDS* system using a VAD segmentation and all *x*-vectors from the *Averaging* representation of past speakers and can be reduced by up to 35% for the case of *ResDS* system using the reference segmentation with the *Averaging* representation of the Nearest speaker per file.

Now comparing the two speaker representations—*Averaging* of *No-averaging*—we find that the *No-averaging* representation always perform (at least slightly) better the *Averaging* representation. This might be explained by the fact that averaging all *x*-vectors from a speaker per show merges robust *x*-vectors extracted from long clean segments with other *x*-vectors extracted from noisy and short segments that degrade the speaker representations. This effect is more clear for the *SincDS* system which embedding's quality is lower than for the *ResDS* system. Reducing the set of *x*-vectors to the nearest speaker per show does not provide any clear benefit.

This experiment also shows that when using *ResDS* system, active correction of the clustering is highly dependent on the quality of the segmentation, which is less the case for SincDS. For *ResDS*, by comparing the performance of active correction when using reference segmentation or an automatic one (the difference of dash line and bars in two below line of Figure 5), it can be observed that the proposed active correction process is more efficient with the reference segmentation. Indeed, when using an automatic segmentation, allowing more questions per speaker leads to an increase of DER_{pen} that ends to be higher than the baseline one. The benefit of active human correction does not compensate for its cost. Remember that the number of questions from 1 to 7 is an upper limit but in many practical cases this number is not reached. Future work will focus on active correction for segmentation.

One could be surprised by the fact that the DER obtained by the *ResDS* system on the Development set is lower when using an automatic segmentation than when using the reference. A first analysis shows that an automatic segmentation using this system generates a higher number of speakers in the within show hypothesis. Each of these speakers having thus a shorter speech duration, the clustering errors that occur during the cross-show diarization affect less the final DER when a speaker is misclassified. The

same effect not being observed with the *SincDS* system underline the fact that side effects of within show diarization errors on the cross-show diarization are complex and will be studied in further studies.

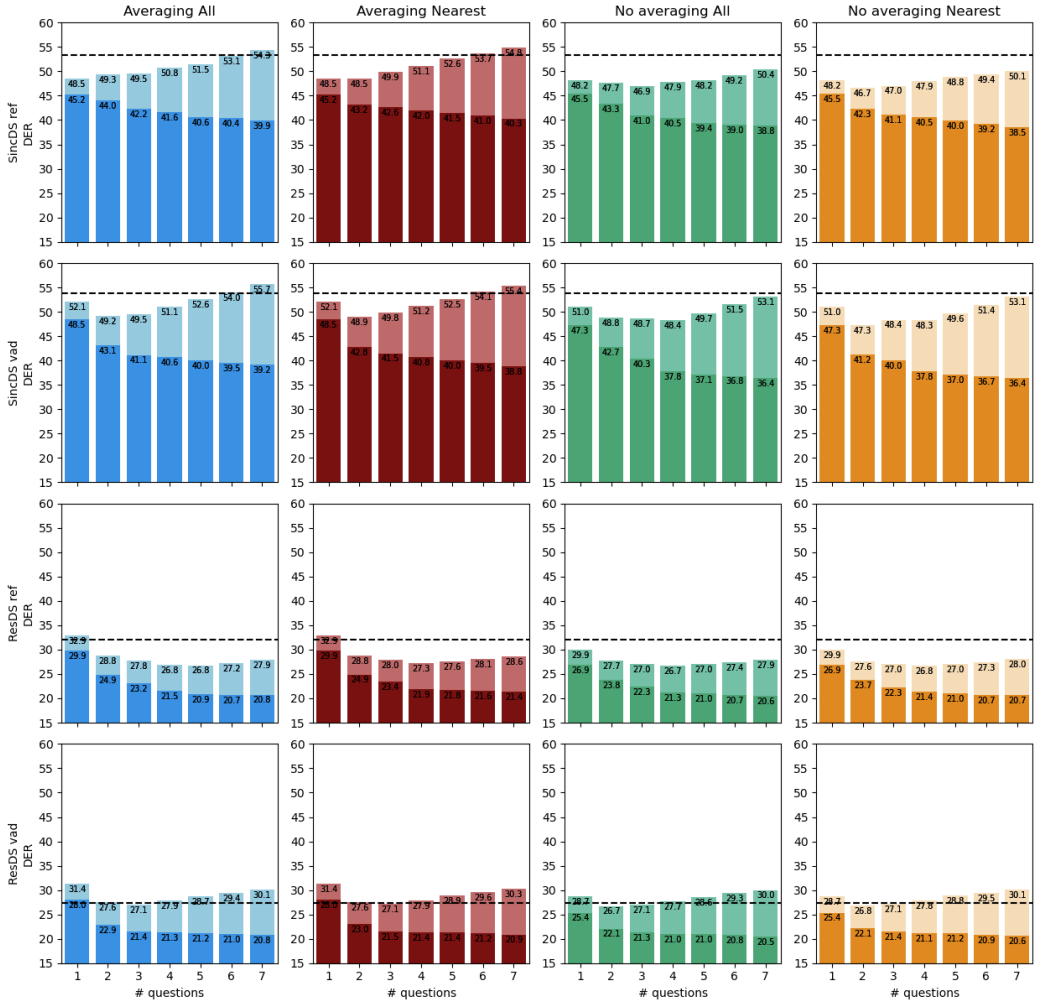


Figure 5. Performance of the proposed human-assisted cross-show diarization approach on the Development partition of the ALLIES corpus. Results are given for both *ResDS* and *SincDS* systems for *Averaging* and *No-Averaging* speaker representations ranking all speakers or only the nearest speaker per show. For each configuration, performance is given for a limit of one to seven questions per current speaker. For each graph, the darkest bars show the DER obtained after human-assisted correction while the lightest bars show the penalized DER. The horizontal dash line on each graph represents the performance of the cross-show baseline described in Section 4.2.

After analysing the results obtained on the *Development* set, we set the optimal configuration for each system using both reference or automatic segmentation, we also set a maximum number of questions per speaker in the current file for each system and perform the corresponding experiment. Results are given in Table 8.

Table 8. Performance of the human assisted cross-show diarization systems on the *Eval* set for the best configuration of each baseline system using either reference or automatic segmentation.

System	**SincNet**		**ResNet**	
Segmentation	**Ref**	**VAD**	**Ref**	**VAD**
Speaker representation	No-averaging			
Ranked speakers	ALL	Nearest per file	All	
Limit of questions per new speaker	2	2	4	2
Baseline DER	51.33	53.85	30.43	31.13
DER	33.78	37.74	20.30	23.88
DER penalized	43.98	46.20	26.84	29.20
Relative improvement of DER	34.19%	29.92%	33.29%	23.29%
Relative improvement of DER pen	14.31 %	14.21%	11.79%	6.19%

The results obtained on the *Evaluation* set and provided in Table 8 confirm observations from the *Dev* set. The quality of the segmentation is essential for the efficiency of the human assisted correction process. We also note that when using the reference segmentation, enabling more questions per speaker during the incremental process leads to better performance even in terms of penalized DER. This means that theoretically allowing more questions might be beneficial without necessarily leading to a higher number of questions actually asked (the automatic system does not need to ask all possible questions).

8. Conclusions

The benefit of human active correction for speaker diarization has been investigated for both within-show diarization and incremental cross-show diarization. This preliminary study has focused on an active correction of HAC clustering errors for the case of within-show diarization and active correction of speaker linking for incremental cross-show diarization.

For within-show diarization, starting from a strong automatic baseline, we proposed two criteria to ask questions to a human expert. Five methods to select samples for auditory tests have been proposed and evaluated using a large and challenging dataset that will be publicly released.

Performance of our human assisted speaker diarization system have been evaluated by using a penalized DER proposed in [36] and shows that it can decrease DER by up to 22.29% relative when applying active correction with the *2c criterion*. This leads to a reduction of 32.07% relative without taking into account the cost of human interaction. The second proposed stopping criterion (*All*) can achieve a relative reduction of 36.51% of DER but requires a higher and less efficient human effort.

For incremental cross-show speaker diarization, we propose a baseline system and an active correction process based on a two step process: detection of previously seen speakers and human-assisted speaker identification. In the best case, our approach reduces the incremental cross-show DER by a relative 33.29% and the Penalized DER by a relative 11.79%.

This preliminary study is very promising and opens large avenues for future studies. More analyses are ongoing to understand and refine the stopping criteria depending on the nature of the processed audio file and its difficulty for diarization systems. Current studies are conducted to improve the question generation module by estimating the quality of the question before soliciting the human expert. We are also developing the adaptation process in order to improve the automatic system using the information provided by the human expert. A limitation of this work comes from the restriction to HAC clustering when many works in the literature have been exploring active learning for other clustering algorithms.

Author Contributions: Conceptualization, Y.P., M.S. and A.L.; methodology, Y.P., M.S., A.L. and S.M.; software, Y.P. and M.S.; validation, Y.P., M.S., A.L. annd L.B.; investigation, Y.P. and M.S.; resources, A.L. and S.M.; data curation, Y.P.; writing—original draft preparation, Y.P., M.S. and A.L. writing—review and editing, A.L. and L.B; visualization, Y.P. and A.L.; supervision, A.L. and L.B.; project administration, A.L.; funding acquisition, A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the CHIST-ERA project ALLIES (ARN-17-CHR2-0004-01) and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101007666, the Agency is not responsible for this results or use that may be made of the information.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The ALLIES Corpus will be made publicly available after the ALLIES Lifelong Learning Human-Assisted Challenge on LIUM's website https://lium.univ-lemans.fr/en/

Conflicts of Interest: The authors declare no conflict of interest

References

1. Anguera, X.; Bozonnet, S.; Evans, N.; Fredouille, C.; Friedland, G.; Vinyals, O. Speaker diarization: A review of recent research. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 356–370.
2. Barras, C.; Zhu, X.; Meignier, S.; Gauvain, J.L. Multistage speaker diarization of broadcast news. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1505–1512.
3. Bredin, H.; Yin, R.; Coria, J.M.; Gelly, G.; Korshunov, P.; Lavechin, M.; Fustes, D.; Titeux, H.; Bouaziz, W.; Gill, M.P. Pyannote. audio: Neural building blocks for speaker diarization. In Proceedings of the ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 7124–7128.
4. Le Lan, G.; Charlet, D.; Larcher, A.; Meignier, S. An Adaptive Method for Cross-Recording Speaker Diarization. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1821–1832.
5. Broux, P.A.; Doukhan, D.; Petitrenaud, S.; Meignier, S.; Carrive, J. Computer-assisted speaker diarization: How to evaluate human corrections. In Proceedings of the LREC 2018, Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018.
6. Ryant, N.; Church, K.; Cieri, C.; Cristia, A.; Du, J.; Ganapathy, S.; Liberman, M. The Second DIHARD Diarization Challenge: Dataset, Task, and Baselines. In Proceedings of the Interspeech 2019, Graz, Austria, 15–19 September 2019; pp. 978–982.
7. Amershi, S.; Cakmak, M.; Knox, W.B.; Kulesza, T. Power to the people: The role of humans in interactive machine learning. *AI Mag.* **2014**, *35*, 105–120.
8. Jiang, L.; Liu, S.; Chen, C. Recent research advances on interactive machine learning. *J. Vis.* **2019**, *22*, 401–417.
9. Wang, M.; Hua, X.S. Active learning in multimedia annotation and retrieval: A survey. *ACM Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 1–21.
10. Prokopalo, Y.; Shamsi, M.; Barrault, L.; Meignier, S.; Larcher, A. Active correction for speaker diarization with human in the loop. In Proceedings of the IberSPEECH 2021, Valladolid, Spain 24–25 March 2021; pp. 260–264.
11. Dawalatabad, N.; Ravanelli, M.; Grondin, F.; Thienpondt, J.; Desplanques, B.; Na, H. ECAPA-TDNN Embeddings for Speaker Diarization. *arXiv* **2021**, arXiv:2104.01466.
12. Landini, F.; Profant, J.; Diez, M.; Burget, L. Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks. *Comput. Speech Lang.* **2022**, *71*, 101254.
13. Park, T.J.; Kanda, N.; Dimitriadis, D.; Han, K.J.; Watanabe, S.; Narayanan, S. A review of speaker diarization: Recent advances with deep learning. *Comput. Speech Lang.* **2022**, *72*, 101317.
14. Fujita, Y.; Kanda, N.; Horiguchi, S.; Xue, Y.; Nagamatsu, K.; Watanabe, S. End-to-end neural speaker diarization with self-attention. In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Singapore, 14–18 December 2019; pp. 296–303.
15. Horiguchi, S.; Yalta, N.; Garcia, P.; Takashima, Y.; Xue, Y.; Raj, D.; Huang, Z.; Fujita, Y.; Watanabe, S.; Khudanpur, S. The hitachi-jhu dihard iii system: Competitive end-to-end neural diarization and x-vector clustering systems combined by dover-lap. *arXiv* **2021**, arXiv:2102.01363.
16. Horiguchi, S.; García, P.; Fujita, Y.; Watanabe, S.; Nagamatsu, K. End-to-end speaker diarization as post-processing. In Proceedings of the ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, QC, Canada, 6–11 June 2021; pp. 7188–7192.
17. Horiguchi, S.; Fujita, Y.; Watanabe, S.; Xue, Y.; Nagamatsu, K. End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors. In Proceedings of the Interspeech 2020, Shanghai, China, 25–29 October 2020; pp. 269–273. doi:10.21437/Interspeech.2020-1022.

18. Yu, C.; Hansen, J.H. Active learning based constrained clustering for speaker diarization. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 2188–2198.
19. Mateusz, B.; Poignant, J.; Besacier, L.; Quénot, G. Active selection with label propagation for minimizing human effort in speaker annotation of tv shows. In Proceedings of the Workshop on Speech, Language and Audio in Multimedia, Penang, Malaysia, 11–12 September 2014.
20. Shum, S.H.; Dehak, N.; Glass, J.R. Limited labels for unlimited data: Active learning for speaker recognition. In Proceedings of the Fifteenth Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
21. Riccardi, G.; Hakkani-Tur, D. Active learning: Theory and applications to automatic speech recognition. *IEEE Trans. Speech Audio Process.* **2005**, *13*, 504–511.
22. Jiaji, H.; Rewon, C.; Vinay, R.; Hairong, L.; Sanjeev, S.; Adam, C. Active Learning for Speech Recognition: The Power of Gradients. In Proceedings of the 30th Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 1–5.
23. Bang, J.; Kim, H.; Yoo, Y.; Ha, J.W. Efficient Active Learning for Automatic Speech Recognition via Augmented Consistency Regularization. *arXiv* **2020**, arXiv:2006.11021.
24. Yilmaz, E.; McLaren, M.; van den Heuvel, H.; van Leeuwen, D.A. Language diarization for semi-supervised bilingual acoustic model training. In Proceedings of the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), Okinawa, Japan, 16–20 December 2017; pp. 91–96.
25. Karakos, D.G.; Novotney, S.; 0002, L.Z.; Schwartz, R.M. Model Adaptation and Active Learning in the BBN Speech Activity Detection System for the DARPA RATS Program. In Proceedings of the INTERSPEECH 2016, San Francisco, CA, USA, 8–12 September 2016; pp. 3678–3682.
26. Abdelwahab, M.; Busso, C. Active learning for speech emotion recognition using deep neural network. In Proceedings of the 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), Cambridge, UK, 3–6 September 2019; pp. 1–7.
27. Settles, B. *Active Learning Literature Survey*; Technical Report; University of Wisconsin-Madison Department of Computer Sciences: 1210 W. Dayton Street Madison, WI 53706-1613 2009.
28. Wang, M.; Min, F.; Zhang, Z.H.; Wu, Y.X. Active learning through density clustering. *Expert Syst. Appl.* **2017**, *85*, 305–317.
29. Basu, S.; Banerjee, A.; Mooney, R.J. Active semi-supervision for pairwise constrained clustering. In Proceedings of the 2004 SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA, 22–24 April 2004; pp. 333–344.
30. Mallapragada, P.K.; Jin, R.; Jain, A.K. Active query selection for semi-supervised clustering. In Proceedings of the 2008 19th International Conference on Pattern Recognition, Tampa, FL, USA, 8-11 December 2008; pp. 1–4.
31. Xu, Q.; desJardins, M.; Wagstaff, K.L. Active constrained clustering by examining spectral eigenvectors. In Proceedings of the International Conference on Discovery Science, Singapore, 8–11 October 2005; pp. 294–307.
32. Wang, X.; Davidson, I. Active spectral clustering. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 14–17 December 2010; pp. 561–568.
33. Miyamoto, S.; Terami, A. Semi-supervised agglomerative hierarchical clustering algorithms with pairwise constraints. In Proceedings of the International Conference on Fuzzy Systems, Barcelona, Spain, 18–23 July 2010; pp. 1–6.
34. Davidson, I.; Ravi, S. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Porto, Portugal, 3–7 October 2005; pp. 59–70.
35. Geoffrois, E. Evaluating interactive system adaptation. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož (Slovenia), 23–28 May 2016; pp. 256–260.
36. Prokopalo, Y.; Meignier, S.; Galibert, O.; Barrault, L.; Larcher, A. Evaluation of Lifelong Learning Systems. In Proceedings of the International Conference on Language Resources and Evaluation, Marseille, France, 11–16 May 2020.
37. Van Leeuwen, D.A. *Speaker Linking in Large Data Sets*. In Procedings of the Speaker Odyssey Workshop, Brno, Czech Republic, 28 June – 1 July 2010;
38. Ferras, M.; Boudard, H. Speaker diarization and linking of large corpora. In Proceedings of the 2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, 2–5 December 2012; pp. 280–285.
39. Sturim, D.E.; Campbell, W.M. Speaker Linking and Applications Using Non-Parametric Hashing Methods. In Proceedings of the INTERSPEECH 2016, San Francisco, CA, USA, 8–12 September 2016; pp. 2170–2174.
40. Karanasou, P.; Gales, M.J.; Lanchantin, P.; Liu, X.; Qian, Y.; Wang, L.; Woodland, P.C.; Zhang, C. Speaker diarisation and longitudinal linking in multi-genre broadcast data. In Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; pp. 660–666.
41. Ferras, M.; Madikeri, S.; Bourlard, H. Speaker diarization and linking of meeting data. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 1935–1945.
42. Galibert, O. Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. In Proceedings of the INTERSPEECH 2013, Lyon, France, 25–29 August 2013; pp. 1131–1134.
43. Giraudel, A.; Carré, M.; Mapelli, V.; Kahn, J.; Galibert, O.; Quintard, L. The REPERE Corpus: A multimodal corpus for person recognition. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey, 23-25 May 2012; pp. 1102–1107.

44. Galliano, S.; Geoffrois, E.; Mostefa, D.; Choukri, K.; Bonastre, J.F.; Gravier, G. The ESTER phase II evaluation campaign for the rich transcription of French broadcast news. In Proceedings of the Ninth European Conference on Speech Communication and Technology, Lisbon, Portugal, 4–8 September 2005.
45. Gravier, G.; Adda, G.; Paulson, N.; Carré, M.; Giraudel, A.; Galibert, O. The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In Proceedings of the International Conference on Language Resources, Evaluation and Corpora, Istanbul, Turkey, 23–25 May 2012.
46. Broux, P.A.; Desnous, F.; Larcher, A.; Petitrenaud, S.; Carrive, J.; Meignier, S. S4D: Speaker Diarization Toolkit in Python. In Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), Hyderabad, India, 2–6 September 2018; pp. 1368–1372.
47. Larcher, A.; Lee, K.A.; Ma, B.; Li, H. Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 7673–7677.
48. Larcher, A.; Lee, K.A.; Meignier, S. An extensible speaker identification sidekit in python. In Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016; pp. 5095–5099.