Human assisted correction for speaker diarization of an incremental collection of documents

Yevhenii Prokopalo

DOCTORAT BRETAGNE LOIRE / MATHSTIC

LIUM
Laboratoire Informatique
Université du Mans

THÈSE DE DOCTORAT DU

LABORATOIRE D'INFORMATIQUE
DE L'UNIVERSITE DU MANS

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Yevhenii PROKOPALO

« Human assisted correction for speaker diarization of an incremental collection of documents »

Thèse présentée et soutenue à Le Mans Université, le 20 Octobre 2022
Unité de recherche : Laboratoire d'Informatique de l'Université du Mans - équipe LST
Thèse N° : 2022LEMA1027

Rapporteurs avant soutenance :

Claude BARRAS	Chercheur, HDR, Vocapia
Corinne FREDOUILLE	Professeur, LIA, Avignon

Composition du Jury :

Président :	Sophie ROSSSET	Examinatrice, Professeur, Université Paris Saclay
Examinateurs :	Hervé BREDIN	Examinateur, Chargé de Recherche, IRIT, Toulouse
	Gaël LE LAN	Examinateur, Chercheur, META FASTAI
Dir. de thèse :	Anthony LARCHER	Professeur, LIUM - Université du Mans
Co-encadrant :	Loïc BARRAULT	Chercheur, META AI

Introduction

The speaker diarization task, also known as speaker segmentation and clustering, consists in determining the number of speakers and when they speak in an audio document or a collection of audio documents. This task is of interest for many companies wishing to index their audiovisual content, improve accessibility and provide annotation for their audio content. Additionally, speaker diarization is used as a pre-processing step for many other speech processing tasks such as speech recognition, speaker and emotion recognition etc.

In order to be valuable, the quality of the annotations of the audio documents has to reach a sufficient level which is, most of the time, not yet achieved by state of the art automatic speaker diarization systems. To achieve the desired performance, many companies employ human annotators to produce manual annotation from scratch or, in order to reduce the cost of the annotation process, ask the human domain expert to correct the output of an automatic diarization system. Nevertheless, human intervention is generally time-consuming and very costly due to the difficulty of the task and to the huge amount of data to process.

Even when correcting an existing automatic annotation, the manual process is extremely long, costly and tedious for several reasons. First, the human domain expert doesn't know which part of the annotation to correct and might thus have to listen to the entire audio document to verify the correctness of the annotations. This process is highly sub-optimal.
The second reason is that an automatic system is likely to perform many errors of the same type that the human domain expert will have to correct one by one across time. This makes the task repetitive and can be very frustrating for the annotator.

This research has been performed in the framework of the European ChistERA project ALLIES, which aims at laying the foundation for development of autonomous intelligent systems sustaining their performance across time. Such unsupervised system should be able to auto-update and perform self-evaluation to be aware of the evolution of its own knowledge acquisition. It should adapt to a changing environment by following a given learning scenario that balances the importance of performance on past and present data to avoid unwanted regression. Such systems could not be developed without adapted metrics and protocols enabling their objective and reproducible evaluation. This evaluation should continuously assess the performance on the given task and quantify the effort required to reach it in terms of unsupervised data collected by the system and of interaction with humans in the case of active-learning. The ALLIES project aims to develop, evaluate and

disseminate those metrics and protocols. Our goal in the project was to apply the concept of human-assisted lifelong learning to the speaker diarization task. More specifically, our work aims to provide an efficient way of interacting between the diarization system and a human domain expert in order to improve the quality of the diarization while limiting the amount of human effort required.

Problematic

To successfully perform the lifelong learning speaker diarization task we had to find solutions to several problems.

The first problem we faced is the absence of a standard definition for human assisted lifelong learning. In the literature there are various definitions, mostly developed for the field of dialog systems. We had to propose an alternative one, which better corresponds to the scope of the Allies project. Another question was the diversity of the different types of interactions between automatic systems and humans, which had no common nomenclature in the literature.

After providing those definitions we faced the absence of the necessary materials required to develop and evaluate human assisted lifelong learning speaker diarization systems. There was no dataset, protocols and metrics which could take into account the specificity of the lifelong learning process. Special attention had also to be paid on the evaluation metric as existing ones didn't take into account the human domain expert or the lifelong learning process.

Eventually, one of the main questions was the development of the human assisted diarization system itself. Such a system requires specific methods and strategies to interact with the human domain expert that are not well developed especially in the field of speaker diarization.

Contributions

This manuscript starts with an overview of speaker diarization, applied to single documents and to a collection of documents. We then propose an analysis of the different definitions of lifelong learning intelligent systems that exist in the literature and provide a nomenclature of different types of interactions between the diarization system and the human expert.

We review the existing material for speaker diarization and propose the ALLIES dataset, protocols and metrics to support the development of human assisted speaker diarization system.

We then describe and evaluate new methods and strategies for human assisted *within-show* speaker diarization that allows to improve the quality of diarization of each single show in a dataset. Eventually we present the methods and strategies developed for human assisted *cross-show* speaker diarization that allows to detect and label recurrent speakers in different shows of dataset across time.

Structure

The proposed thesis is organised in the following way. The first part is dedicated to the analysis of existing works in the domains of speaker diarization and lifelong learning. The first chapter is an overview of existing algorithms and methods used for the speaker diarization task while the second chapter focuses on the question of lifelong learning in different domains and propose an alternative point of view.

The second part of this manuscript is dedicated to protocols, metrics and corpora and contains two chapters. The first of this chapters describes the existing corpora and metrics while the second one describes our contribution in terms of protocols, corpora and metrics for the task of human assisted lifelong-learning speaker diarization.

The final part of this document is dedicated to methods and strategies for active correction in the context of lifelong learning speaker diarization. It contains two chapters, a first one, that describes our contributions and results on active correction for within-show speaker diarization and a second one that describes our contributions and results on active correction for cross-show diarization.

Part I

Towards Human Assisted Lifelong learning speaker diarization

Speaker diarization is a pre-processing step applied before many speech processing applications when recordings contain more than one speaker. Speaker diarization aims at answering the question: "who speaks when?" in an audio recording. The goal of speaker diarization is to find the temporal borders: start and stop, of homogeneous speech segments and to label those segments with speaker identifiers that have to be consistent across the processed audio recording.

Speaker diarization can be applied on a single audio file (or show) and will then be referred to as *within-show* speaker diarization or on a collection of shows, in which case we will use the term *cross-show* speaker diarization. In this part, we will first provide a brief overview of within-show speaker diarization in the beginning of Chapter I. The end of chapter I will then review cross-show speaker diarization in the literature and describe the novelties introduced in our work towards lifelong learning speaker diarization within the ALLIES project. Similarly, Chapter II will discuss the definitions of human assisted learning and lifelong learning in the context of our work with regards to the existing literature.

Chapter I

An overview of speaker diarization

Historically, within show speaker diarization is often performed in two steps referred to as segmentation and clustering. Segmentation aims at detecting borders of homogeneous speech segments while the goal of clustering is to label those segments with speaker identifiers that are kept consistent along the processed audio file (show). Clustering and segmentation often rely on segment comparisons and thus require robust representations of audio segments that can be obtained via statistical or neural models. The first section of this chapter describes the most common acoustic models that are used along the diarization process. The second Section reviews the different tasks that can be combined for segmentation purpose, i.e., voice activity detection, speaker turn detection and overlap detection. The most common clustering approaches used in recent systems are discussed in Section I.3 together with re-segmentation methods that can be used to refine the result of the overall process. Eventually, Section I.4 presents recent end-to-end neural approaches.

I.1 Acoustic modeling

In this section we introduce the main tools used for signal representation and comparison in the speaker diarization domain. The list is not exhaustive but presents the main approaches and models that are shared across the different processing steps. Those tools are introduced considering the duration of audio signal they are used to model.

I.1.1 Acoustic features representation

In general, automatic speech processing systems take as input an audio signal which sampling frequency is set between 4 and 48 kHz. However, raw signals are noisy and convey redundant information. In order to be used by automatic systems, those signals have to be compressed and possibly enhanced, this process is referred to as feature extraction.

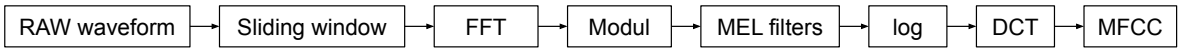

Figure I.1 – Standard Mel Frequency Cepstral Coefficient extraction toolchain.

I.1.1.1 Signal processing based features

During the last decades, many signal processing methods have been developed to tackle this task [1], [2]. The most commons include Linear Predictive Coding (LPC)[3], Perceptual Linear prediction (PLP) [4], [5], constant Q cepstral coefficient(CQCC)[6] but the most popular is still the Mel Frequency Cepstral Coefficients (MFCC) [7], [8].

MFFCs (Mel Frequency Cepstral Coefficients) provide a time frequency representation of the speech signal. The process that produces MFCCs is described in Figure I.1.

MFCC features are computed on a sliding window which duration is usually between 20 and 50ms. Length of this window is set to respect the hypothesis of speech quasi-stationary [8]. To avoid information loss, successive windows are overlapping in order to generally compute MFCCs features with a frequency of 100Hz. The signal is first multiplied by a smoothing window (Hamming usually) in order to avoid artefacts due to the chunking. A pre-emphasis filtering is applied before computing the Fast Fourier Transform. Energy is then computed on frequency bands selected by triangle filters which fundamental frequencies follow the MEL scale. The number of filters determines the number of resulting MFCCs. The logarithm of this values are multiplied by 20. At this stage it is already possible to use the features to feed the automatic speech processing systems, in literature they are referred to as log-filter-bank features [9]. Eventually, a Discrete Cosines Transformation (DCT) is applied in order to compress the information and decorrelate the resulting components. The number of chosen coefficients depends on the resolution of the desired representation, usually in the order of tens. Also it is common to use the first and second derivatives of features to add information about their dynamic.

I.1.1.2 Neural network based features

Artificial neural networks have been used since the 1990s to provide appropriate signal representations for the speech recognition task [10]. Recent developments of deep neural networks have brought new architectures to extract those representations such as bottleneck features [11], [12].

One advantage of those bottleneck features comes from the large temporal context

they are computed on (a few hundreds of milliseconds [12], [13]). Indeed, the input of the network is usually the concatenation of sequence of classical acoustic features. Another advantage is that the resulting bottleneck features can be optimized for a given task (the one the extractor is trained for).

Variants of this bottleneck features have been proposed over the years. Some works propose to train bottleneck extractors optimized on multiple tasks in order for those features to convey a maximum of the speech signal information. This architecture is named Problem Agnostic Speech encoder [14]. With time and the development of deep networks, speech representations are extracted by using more complex architectures of networks like staked bottleneck features [13] and recent work even make use of self-supervised pre-trained models to encode the speech signal into a high dimensional sequence of vectors [15]–[17]. Encoders consist of very deep neural networks often involving transformers fine-tuned for multiple tasks.

One other current trend of research consists of developing end-to-end neural network architecture. Several approaches have been proposed in order to directly process the raw waveform and optimize the entire processing tool-chain for the targeted task [18]. An example of this type of architecture is SincNet [19]. First layers of this network are designed to integrate knowledge about the nature of the speech signal, that allows to train the feature extractor part respectively with the further processing.

I.1.2 Gaussian based models

After feature extraction a speech segment is represented as a sequence of constant length vectors, corresponding to overlapping frames of this segment. A simple method to encapsulate the information from a whole audio segment is to use a Gaussian model $\Theta(\mu, \Sigma)$. Where μ is the average of the features and Σ is their co-variance. Representation with Gaussian model can be successfully applied to short segments but, on longer segments, more complex methods show significantly better performance. The more complex methods are described in the sections I.1.2.2 and I.1.2.3. To compare two segments using those representations, different methods are available such as Generalized Likelihood Ratio (GLR) measure [20], Bayesian information criterion (BIC) [21], Kullback-Leibler divergence [22], [23], Gaussian Divergence [24] or the Probabilistic Linear Discriminant Analysis[25]. GLR, BIC and PLDA are the most widely used and will be described further in section I.1.2.1 and I.1.2.4.

I.1.2.1 Likelihood ratios

Generalise likelihood ratio [20] (GLR) allows to compare two segments x_i and x_j which are represented with Gaussian models $\Theta_i(\mu_i, \Sigma_i)$ and $\Theta_j(\mu_j, \Sigma_j)$. The calculation of the GLR consists of estimating the likelihood ratio between the two hypotheses: H_{same} and H_{dif}. H_{same} declares that x_i and x_j correspond to the same class (same speaker or same non-speech segment) and H_{dif} declares that x_i and x_j correspond to two different class. In the first case, the best representation of the segments would be a Gaussian model $\Theta_{i,j}(\mu_{i,j}, \Sigma_{i,j})$ estimated on union of x_i and x_j, while in the second case, the representation by the two distinct models Θ_i and Θ_j would be more suitable. We therefore want to calculate:

$$GLR(x_i, x_j) = \frac{L(x_i, x_j|\Theta_{i,j}(\mu_{i,j}, \Sigma_{i,j}))}{L(x_i|\Theta_i(\mu_i, \Sigma_i))L(x_j|\Theta_j(\mu_j, \Sigma_j))} \tag{I.1}$$

This GLR measurement makes it possible to define a distance between the sequences x_i and x_j, $d(x_i, x_j) = -log(GLR(x_i, x_j))$, which, with the predefined threshold, allows to decide that segments i and j belong to the same class or not.

The Bayesian information criterion (BIC) [21] is similar to GLR but with a penalization factor based on the complexity of the model. The calculation of BIC measure between segments x_i and x_j, which are represented by n_i and n_j acoustic vectors, is given by the following equations:

$$BIC(i, j) = (n_i + n_j) * log|\Sigma_{i+j}| - n_i * log|\Sigma i| - n_j * log|\Sigma_j| - \lambda P \tag{I.2}$$

$$P = (D + (D(D + 1))/2) + log(n_i + n_j) \tag{I.3}$$

where $|\Sigma_i|$, $|\Sigma_j|$ and $|\Sigma_{i+j}|$ are respectively the determinants of the full covariance matrices of segments x_i, x_j and of the union of the two segments x_{i+j}. P is calculated from the complexity of the model depending on the dimension D of the acoustic vectors and the values of n_i and n_j. λ is a factor regulating the weight of the penalty P. λ is the main parameter of the system to choose, knowing that the greater its value, the more classes are merged.

I.1.2.2 GMM

The approach consisting in modeling speakers using statistical models based on mixtures of Gaussian distributions has been developed in the 1990s. The use of the Gaussian

Mixture Model (GMM) is based on the idea that the distribution which characterizes the set of acoustic vectors of a given speaker is a weighted sum of several Gaussian distributions [26], [27]. Let $X = (x_i), i \in 1..N$ be a set of acoustic vectors and the C components GMM model $\Theta_c = (\omega_c, \mu_c, \Sigma_c), c \in 1..C$ be its density of probability with:

$$p(x_i|\Theta) = \sum_{c=1}^{C} \omega_c p_c(x_i|\mu_c, \Sigma_c) \tag{I.4}$$

where $\omega_c, \mu_c, \Sigma_c$ are parameters of the Gaussian distribution number c, and $\sum_{c=1}^{C} \omega_c = 1$. It is important to notice that this model is defined by the set of mean vectors and covariance matrices of its components as well as the weight ω_c parameter of each of these components in the sum. Parameters of the Gaussian mixture can not be estimated directly and require the use of an Expectation Maximization(EM) algorithm [28], [29]. At each iteration of this estimation process, mean and variance parameters of each of the Gaussian components of the mixture are estimated as in the case of a mono-Gaussian model except that each observation is weighted in order to calculate the mean and the variance of each distribution. The weight of each observation is determined by taking into account all the components of the mixture.

I.1.2.3 I-vectors

Factor Analysis is a probabilistic model that aims at factorizing the speech signal features into factors related to speakers and other variations [30], [31]. The Total Variability paradigm considers a single sub-space that conveys all information from the speech signal. This paradigm assumes that a GMM super-vector, s, (i.e. a vector obtained by concatenating the mean vectors of each distribution from a GMM), is generated following the equation:

$$s = m + Tw \tag{I.5}$$

where s is the utterance super-vector, m is a speaker and channel independent super-vector from the UBM, T is the total variability matrix, and w is the i-vector. An i-vector system can be used as a feature extractor to extract a low-dimensional fixed-size representation vector from a speech utterance.

I.1.2.4 Probabilistic Linear Discriminant Analysis

Probabilistic Linear Discriminant Analysis (PLDA) is a generative model originally proposed for face recognition [32] that has been widely used in speaker recognition and diarization systems [33], [34]. Training of a PLDA model requires a corpus containing several recordings of each speaker. Originally, the vector $w_{l,r}$ associated with recording r of speaker l is expressed as the sum of three terms:

$$w_{l,r} = \mu + Fy_l + Gx_{l,r} + \epsilon \tag{I.6}$$

where μ represents the average of the vector distribution from the learning corpus. The matrix F is a basis of the speaker sub-space, the matrix G a basis of the session-subspace (all the variations of the sessions) and ϵ is an residual noise modeled by a Gaussian distribution with a diagonal covariance matrix. In speaker recognition and diarization, the session subspace is not used and the variability is modeled by using a full residual covariance matrix. y_l is the latent variable dependent on speaker l and $x_{l,r}$ is the latent variable dependent on speaker l and recording session r. The latent variables are assumed to be independent and to follow a Gaussian law. The parameters, $\mu, F, G,$, of the PLDA model are estimated using the EM-ML algorithm [35].

The comparison of two vectors w_i, w_j, is performed by considering two hypotheses H_{same} and H_{dif} as for the cases of GLR and BIC. The score is calculated as the logarithm of the following likelihood ratio:

$$S_p(w_i, w_j) = log\frac{p(w_i, w_j|H_{same})}{p(w_i, w_j|H_{dif})} \tag{I.7}$$

I.1.3 Neural networks for speaker diarization

Since 2014 [36], deep neural networks tend to become state-of-the-art models for speaker recognition and diarization. Various architectures can be used for the different steps of the diarization process including feature extraction ([19], [10]), audio segmentation ([37],[38]) or audio segment representation ([19], [39], [40], [41], [42]). In this section we describe two of the most used architectures for speaker diarization.

I.1.3.1 X-vectors

x-vectors have been proposed to produce robust fixed length representation of speech segments by using deep neural networks. Across time, they have been successfully used for speaker verification[42], language identification [43], emotion recognition [44] and speaker diarization in different architectures [45]. Original x-vector neural network is a Time delay neural network and have the following architecture, depicted in I.2.

The input of the neural network is a sequence of acoustic parameters. First layers of the network are convolutional layers, which regroup the context of multiple consecutive frames and represent them with high-dimensional vectors. At this moment the segment of speech is represented as a sequence of vectors which correspond to highly overlapping chunks of speech. To accumulate information from all those vectors it has been proposed to compute the mean and standard deviation on the output of the last convolutional layer. This operation produces a fixed-length representation of the given speech segment that is then refined and compressed by the next two linear layers that decrease the dimensionality of the representation vector to a few hundreds. The resulting embedding is used to represent the speaker. During the training of the model, a Softmax layer, on top of the architecture, is used to perform classification of the speaker and calculate the loss during the training phase.

Recently other architectures have been proposed in order to use deeper networks often involving ResNet structures [46]–[48], attention mechanisms [49] and possibly processing raw wave forms [19], [40], [50].

I.1.3.2 Bi-LSTM

Architecture based on multiple layers of Bi-LSTMs followed by fully connected layers has been proposed for the task of sequence-to-sequence prediction.

The architecture is organized as depicted in I.3 This network is composed of two Bi-LSTM and a multi-layer perceptron (MLP) whose weights are shared across the sequence. Bi-LSTMs allow to process sequences in forward and backward directions, making use of both past and future contexts. The output of both forward and backward LSTMs are concatenated and fed forward to the next layer. The shared MLP is made of three fully connected feed-forward layers, using *tanh* activation function for the first two layers, and a *sigmoid* activation function for the last layer, in order to output a score between 0 and 1.

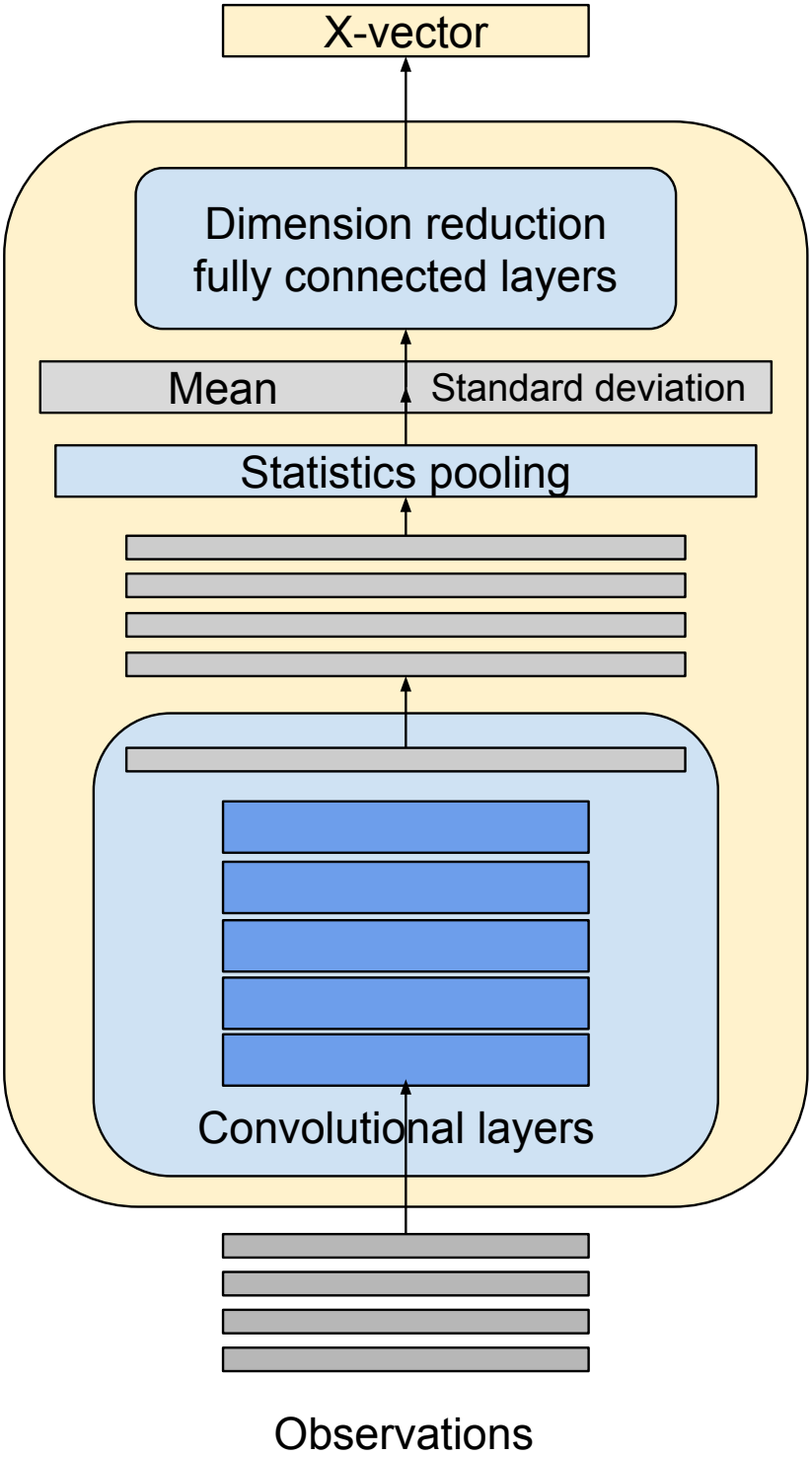

Figure I.2 – Architecture example of an x-vector extractor neural network including 5 convolutional layers, statistic pooling and 3 fully connected layers.

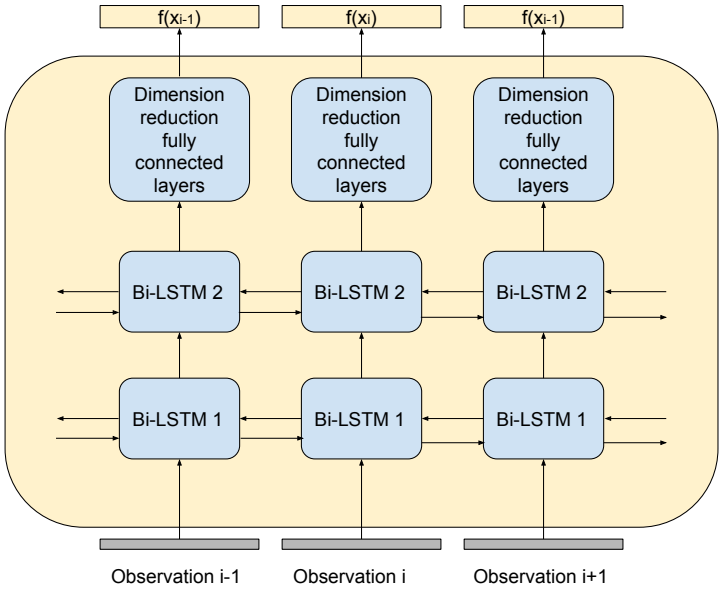

Figure I.3 – Example of Bi-LSTM neural network architecture used for speaker diarization. This architecture includes to layers of bi-LSTM and X fully connected layers.

This architecture has been applied for different tasks [38] by changing the labels in the sequence to predict. It achieves state-of-the-art results for the task of voice activity detection, speaker turn detection, embedding extraction and re-segmentation. Those applications will be described later on.

I.2 Segmentation

As in many machine learning tasks [51],[28], speaker diarization requires a data pre-processing step. For the case of speaker diarization, standard pre-processing steps include acoustic feature extraction, voice activity detection (VAD) and overlapped speech detection. Feature extraction methods are described in section I.1.1 Voice activity detection step aims at separating speech and non-speech parts of the recording in order to avoid miss-classifying non-speech segments as speaker turns. This step allows to facilitate the speaker segmentation step. This tasks is common for almost all domains of speech processing such as speaker and speech recognition [7], [52]–[54].

I.2.1 Voice activity detection

This step aims at separating speech and non-speech parts of the recording. Non-speech segments may contain silence, music and other background noises. Voice activity detection impact diarization in two ways. First, it decreases error of miss and false detection of speech segments. Second, a good VAD helps obtaining good speaker representations as the missed speech segments limit the amount of available data for speaker representation while false detection of speech segments adds noise to the representation of the speakers. It is possible to skip the VAD step and detect the non-speech segments during segmentation but it was observed that VAD during pre-processing can lead to better results [55].

Simple voice activity detection methods make use of GMM and Hidden Markov Models in order to represent each acoustic class (speech,silence, noise, etc.) by a mixture of Gaussian distributions [56]. A Viterbi decoding can then be performed to detect speech and non-speech segments. In addition, discriminant classifiers such as Linear Discriminant Analysis (LDA) [57] or Support Vector Machines (SVM) [58] have been used for the VAD task.

As for feature extraction, the last decade has witnessed the development of neuronal methods for voice activity detection. Amongst architectures that show high performance for this task, one can citep the bidirectional Long Short-Term Memory (Bi-LSTM) networks [59] which architecture is described in Section I.1.3 or Temporal Convolutional Networks (TCN) [60] . As outputs, those networks return sequences of frame-level scores (between 0 and 1). Before the application of the network the input audio recording are windowed (segments of nearly 3 seconds with overlapping of nearly one second). Then neural network is applied. The scores of overlapping segments are averaged. Then the sequence of scores is post-processed using two thresholds for the detection of the beginning and end of speech regions.

I.2.2 Speaker Change Detection

Speaker Change Detection is one of the main steps in speaker diarization task. It aims at finding the boundaries between speech turns of different speakers in a given audio sequence and then split the audio stream into speaker homogeneous segments which will be used for the clustering step.

I.2.2.1 Uniform segmentation

As the aim of speech segmentation consists of producing short homogeneous segments, i.e., segments which audio content entirely belongs to one single class, one straightforward option is to segmentate the audio signal into very short segments in order to maximize their purity. This approach has been shown efficient in the context of the DiHard evaluation [45], [61] where the uniform segmentation is done using a sliding window of 1.5 second duration with an overlap of 0.75s.

This approach is very simple but is not optimal regarding to the following step of the processing toolchain. Indeed, the segmentation step is followed by a speaker representation step such as i-vectors or neural embeddings that have been shown more discriminant when estimated on long duration [62]–[66].

I.2.2.2 Probabilistic approach

Probabilistic approaches can be used to compare consecutive segments and decide whether they were spoken by the same speaker or not. The longer a segment of speech, the more complex the technique of comparing two segments can be, but the lower the probability that it will be homogeneous from a speaker point of view. The result of segmentation into speakers is therefore a compromise between length of segments and homogeneity.

A simple approach consists of comparing short, fixed length segments using two sliding windows that run through the audio document. Many measure have been proposed in the literature to compare two consecutive windows: GLR measure [20], the Kullback-Leibler divergence [22], [23] or the Gaussian Divergence [24].

The most widely used measure is the Bayesian Information Criterion (BIC) [21], which is described in section I.1.2.1 This method requires setting a threshold, that is chosen empirically to obtain homogeneous segments in terms of speakers.

I.2.2.3 Neural network based approaches

Recently, neural networks have been also applied to the segmentation task. It has been formulated as a classification task in which the neural network is used to classify frames including a speaker change [67]. In this article it has been shown that the proposed system can reduce the number of missed changing points, which corresponds to small segments of speech, compared with traditional methods. The best results of this approach was

obtained using bi-directional Long Short-Term Memory Network [68],described in section I.1.3.

I.2.3 Overlap detection

Another difficulty of segmentation step is to detect overlapping segments of speech. It is the possible case when the two or more speakers speak at the same time. Such segments should be accordingly detected and labeled with all speakers speaking simultaneously. Early approaches for speech overlap detection were based on the HMM architecture [69]. In modern sysltes the best results are obtained by neuronal approach such as block-based CNN architectures [70].

I.3 Clustering

Clustering is the most important step in speaker diarization system. Once the segmentation into speakers has been performed, the audio document is divided into homogeneous segments corresponding to different speakers, but resulting segments are often of short duration and are not labelled. The goal of clustering is to group segments corresponding to the same speaker and label them accordingly.

Many algorithms have been developed to perform clustering, based for most of them on the computation of a similarity matrix between the segments. Note that many similarity measures have also been used for speaker diarization: Gaussian / BIC [71], i-vector/PLDA [71], x-vector/PLDA [45]. This section presents some of the most common clustering algorithms used in speaker diarization.

I.3.1 Hierarchical Agglomerative Clustering

After calculating the similarity matrix, classical approaches of unsupervised clustering are possible. The most commonly used is probably the hierarchical agglomerative clustering [20], [72], [73], but there are other methods applied to the diarization task such as k-means (k-means) [74], clustering based on representations in the form of graphs [75], clustering based on the Integer Linear Programming [76], or spectral clustering [77], [78].

The hierarchical agglomerative clustering (HAC) [79] groups successively vectors or classes (here, the speech segments) according to their relative distance. It is an iterative process starting from n initial classes. At each iteration, the two closest classes are grouped

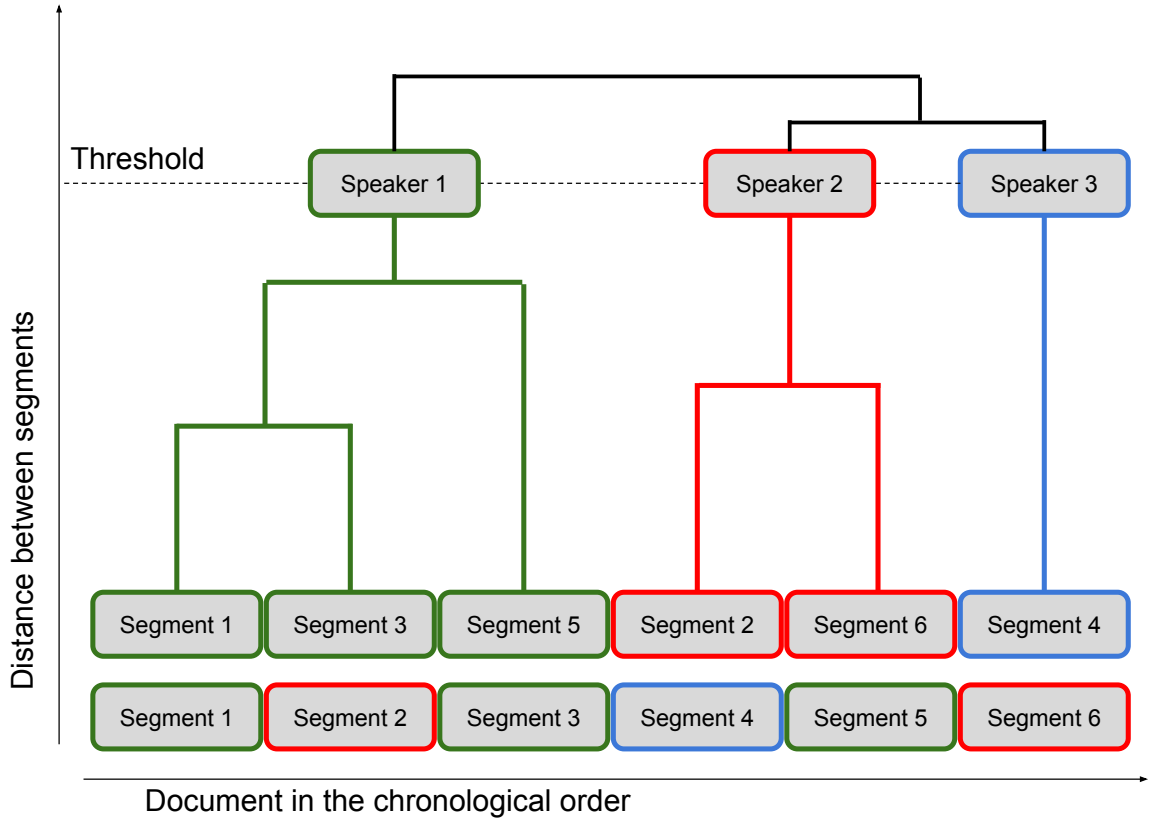

Figure I.4 – Example of dendrogram obtained during a hierarchical agglomerative clustering. Segments are grouped according to their distances until an empirically defined threshold.

into one, and the distance to the other classes is updated. The algorithm stops when there is only one class left or when a stopping criterion is reached (for example, when the dedicating distance of the next grouping reaches a threshold, or when a given number of classes is reached, if we know the total number of speakers of the document). The algorithm is illustrated in I.4

The distance between clusters can be computed with different measures: with the Nearest point algorithm, where the distance between clusters is defined as the distance between nearest points of two clusters, with the Farthest Point Algorithm, where the distance between clusters is defined as the distance between farthest points of two clusters, with the UPGMA algorithm where the distance between clusters is defined as the average distance between all elements of two clusters, etc.

I.3.2 Spectral clustering

An alternative approach for speaker clustering is to use the spectral clustering. Firstly this method was proposed in the work of [78]. Authors applied the N-Jordan-Weiss (NJW) spectral clustering algorithm and reached performance similar to HAC based method with lower computational cost. This approach was then developed in the work of [80], where authors proposed a method to auto-tune the parameters of the clustering algorithm. Another development of this approach was proposed by the [81]. In their paper authors proposed to measure the similarity between speakers segments using bidirectional long short term memory network and apply spectral clustering on top of it.

I.3.3 Variational Bayesian approaches

In [82], [83], the authors present an approach that clusters x-vectors extracted on a sliding window. After automatically estimating the number of speakers in a file, a Variational Bayesian HMM (VB-HMM) is used to cluster the x-vectors by applying an iterative re-segmentation process. This method has shown excellent performance in the latest benchmarking evaluations [84].

I.4 End-to-end speaker diarization

Most of the recent researches are oriented on the use of the deep learning approaches. The most recent researches tries to build completely neural networks pipelines. This section describes such researches.

The authors of [85] propose an unbounded interleaved-state recurrent neural networks (UIS-RNN) which solves both segmentation and clustering tasks. It takes an input sequence of embeddings and returns an output probability for each speaker being present at each time frame.

Another attempt to build an end-to-end neural network system was proposed in the work of [86] by using region proposal networks (RPN). This architecture was first used to detect multiple objects on the two dimensional image [87]. It is composed of three neural network blocks that compute features, performs SAD, speaker embedding extraction and re-segmentation. The proposed pipeline still includes the clustering step which is applied after the speaker embeddings extraction. In this sense, this system can not be considered as fully end-to-end.

[88], [89] propose to combine speech separation, speaker counting and speaker diarization withing one single architecture using the online Recurrent Selective Attention Network (online-RSAN). This system showed interesting results on dataset with no more than 6 speakers per file.

The EEND framework was presented in [90] and offers a pipeline based on a single neural network. This system takes acoustic features as input and produces as output the sequence speech activity for each speaker in the file. The drawback of this approach is the need of defining a maximum number of speakers per file before training. The further extension of this approach: EEND-EDA (encoder-decoder based attractor) proposed in [91] has been developed to solve this limitation. This method uses LSTM-based encoder–decoder on the output of EEND to generate multiple attractors. Each attractor corresponds to a speaker and is used to estimate the activity of this speaker.

Such approaches improved the performance of automatic diarization systems but still benefit from a separated clustering that is difficult to perform in a neural network [92] Most of them have been tested with a limited number of speakers per show.

I.5 Cross-show speaker diarization

In real world tasks it often happens that the a same speaker appears in different audio recordings (TV or radio shows, meetings...). In this work, we don't consider the speaker identification (naming) as part of the within-show diarization process. As a consequence, the speakers detected in a show are labeled with meaningless IDs that are not consistent across recordings. A speaker appearing in several shows will thus be labelled with as many IDs as the number of shows he/she appears in. To guaranty consistency between shows and enable a proper indexing of the audio-visual collection, the automatic system must link appearances of a same speaker within different shows by using the same unique label per speaker across the collection of audio-visual documents. The task of detecting and linking speakers across shows is referred to as cross-show speaker diarization in the literature.

Ideally, cross-show speaker diarization would be applied to the whole dataset at once. The global processing of the audio-visual collection would then enable leveraging all possible resources per speaker in the collection when applying speaker linking across shows. However, such a scenario is only possible when all shows to process are available at the same time. In practice, waiting for the collection to be complete before performing cross-

show speaker diarization will induce a severe latency in the process. Two options are thus available: first, at time T, one can process all available audio-visual documents to perform cross-show speaker diarization as a global process that will be applied again after collecting each show in the future. The second option consists of incrementally processing incoming shows without re-processing previous ones. While the first approach, a global cross-show diarization process, has been studied in the literature [93], [94], the second one, that we will referred to as incremental cross-show speaker diarization, has been less explored [95], [96] and will be further described in the following section.

I.5.1 Global cross-show diarization

This subsection describes the work on global cross-show speaker diarization which is also referred as 'cross-show speaker diarization'. This task was introduced in the literature in the work of [73]. The authors use an ascending hierarchical classification with the CLR measure to group mono-speaker telephone recordings. This approach was extended and developed in other works such as [97] and [71]. Also global cross-show speaker diarization is sometimes referred to as speaker attribution or speaker linking [98].

I.5.2 Incremental cross-show diarization

This subsections describes the specificity of the incremental cross-show clustering, its constraints and differences with the global cross-show clustering.

First of all, during the incremental cross-show clustering, the system does not have access to all data at once. The files are processed one by one and the next file arrives when the processing of the previous one is finished. At this, the system has only access to the information about speakers in the previously processed files, compared to the case of global cross-show diarization where all information is available at once. In other words, when the processing of file N is finished, the processing of file $N+1$ starts. At that time, the system has only access to the information about speaker in files $0..N$. The system compares the speakers in file $N+1$ with the speakers from the previous files and changes the labels of the speakers in file $N+1$ if it finds the same speaker in files $0..N$. When the processing of file $N+1$ is over, the system starts processing file $N+2$ and has access to speaker information from files $0..N+1$. It is important to notice that the system can not modify the labels from files $0..N$ while processing of file $N+1$. It is the second important constraint; after obtaining new information the system can not change its decision.

On the one hand, this restriction makes withing-show diarization more complicated, but on the other hand it can be still be applied when the use of the global cross-show diarization becomes costly.

Such differentiation of global and incremental cross-show diarization is not very common in the literature, but it was used in the works of [95] and [96].

Chapter II

Lifelong learning in the context of the ALLIES project

This research has been done in the framework of the European ChistERA project ALLIES (`https://www.chistera.eu/projects/allies`), which aims to design an autonomous diarization system able to adapt and auto-evaluate itself.

The idea of the ALLIES project is to reach the best possible performance across time, by continuously adapting an autonomous system with the incoming stream of data. The training and adaptation should be performed without any intervention from a machine learning expert. However, a human domain expert can be included in the loop, in an active learning setup, to provide relevant information and adapt the model. The system must then decide what questions to ask to the human domain expert and how to best use their answers. Along this process, the human expert and the system interact in a continuous manner to process the incoming data.

Lifelong learning has been studied for many years [99] but very few attempts have been done to apply this to speaker diarization [95], [100]. This chapter describes the different concepts inherent to lifelong learning, their application in different application domains and more specifically to speaker diarization. This chapter provides our definition of human-assisted lifelong learning, that has been used throughout the ALLIES project and that led the research presented in this document.

II.1 Definition

Modern machine learning (ML) achieves great performance in various domains. Deep learning paradigm proposes to apply the ML algorithms on a well prepared dataset to train the model. Machine learning relies on models trained from massive quantities of examples. Those models are then used to solve real world tasks. Despite their excellent results, example learnt machine learning algorithms suffer from two main drawbacks.

First, real world is changing and machine learning algorithms trained in a one-shot process are facing a flow of constantly evolving incoming data which nature quickly drifts away the one from the data used for the initial training of the system. Classic machine learning methods tackle this issue by increasing the generalization power of automatic systems but generalization might lead to poor performance and is unlikely to cover the diversity of incoming data over a long period of time [95]. One way to avoid this consists of retraining the model. Another option is to develop lifelong learning to extend the paradigm of machine learning and continuously adapt the model to the flow of incoming data.

The second drawback of classic machine learning algorithms comes from the massive quantity of data required for their training or adaptation. Both supervised and unsupervised ML systems require big datasets to be trained correctly. Developing a machine learning system for a new task or just performing a domain adaptation is very costly even if solutions already exist for a similar task. Training supervised ML systems requires large quantity of work by the human experts for data annotation. Lifelong learning can partially answer this second drawback by continuously processing data, avoiding a one-time processing of huge amount of data, but it doesn't reduce the amount of work required from human annotators. Human assisted learning is one way to address this issue by creating an interaction between the automatic system and a human operator in order to provide useful information to adapt the model while reducing the work load on the human annotator by carefully selecting the data to annotate.

In this chapter, we first discuss the definitions of lifelong learning existing in the literature. Then, we provide our definition of human assisted lifelong learning, the one that has been used within the ALLIES project. Finally, we describe the different components that compose this human assisted lifelong learning.

II.1.1 Lifelong learning within the ALLIES project

Multiple definitions of lifelong learning can be found in the literature. In their work, Silver, Yang, and Li [101] define the lifelong machine learning as follows:

> *"Lifelong Machine Learning, or LML, considers systems that can learn many tasks over a lifetime from one or more domains. They efficiently and effectively retain the knowledge they have learned and use that knowledge to more efficiently and effectively learn new tasks."*

[99] gives a more detailed definition:

"Lifelong learning (LL) is a continuous learning process. At any point in time, the learner has performed a sequence of N learning tasks, $T_1, T_2, ..., T_N$. These tasks, which are also called the previous tasks, have their corresponding datasets $D_1, D_2, ..., D_N$. The tasks can be of different types and from different domains. When faced with the $(N+1)th$ task T_{N+1} (which is called the new or current task) with its data D_{N+1} , the learner can leverage the past knowledge in the knowledge base (KB) to help learn T_{N+1}. The task may be given or discovered by the system itself (see below). The objective of LL is usually to optimize the performance of the new task T_{N+1} , but it can optimize any task by treating the rest of the tasks as the previous tasks. KB maintains the knowledge learned and accumulated from learning the previous tasks. After the completion of learning T_{N+1}, KB is updated with the knowledge (e.g., intermediate as well as the final results) gained from learning T_{N+1}. The updating can involve consistency checking, reasoning, and meta-mining of higher-level knowledge. Ideally, an LL learner should also be able to:

1. *learn and function in the open environment, where it not only can apply the learned model or knowledge to solve problems but also discover new tasks to be learned, and*
2. *learn to improve the model performance in the application or testing of the learned model. This is like that after job training, we still learn on the job to become better at doing the job."*

Both definitions are not formal and no unique definition is yet commonly accepted by the community. An important characteristic of the LL is that it is a continuous process during which the system changes to improve its performance. Also, both authors in their definitions specifies the possibility of the system to learn new tasks. Within the ALLIES project, the possibility of LL system to learn new tasks is not considered. The ALLIES project focuses on improving the performance of the system on a single task across time and domains. The scope of the ALLIES project includes two tasks: speaker diarization and machine translation. This research is focusing on the speaker diarization task only but a special care has been taken in order to develop generic approaches that could be adapted to other tasks. This is especially true for the creation of metrics in chapter III. In this work, we consider that learning new tasks for a system does not refer to 'Lifelong learning' but to the general 'Artificial intelligence' concept which is out of the scope for this research.

The definition of LL given by [99] considers a knowledge base that may store information such as the original data used in each previous task, intermediate results from each previous task, and the final model or patterns learned from each previous task. We consider that the knowledge base as a part of the LL system is possible but not necessary.

Regarding the real world application, storing all the data processed by the users may not be a good idea as it can affect scalability and generate legal issues. Additionally, the memory capacity of modern deep learning models makes can replace an explicit knowledge base, rising however the question of explainability.

Considering the tasks addressed in the ALLIES project, we foresee another important characteristic of Lifelong learning systems that is not addressed in the previous definitions: we consider that a real world usage of a lifelong learning system requires human-machine interactions as users are constantly interacting with the system. Current machine learning algorithms allow only one possible interaction when the user provides data to the system. Lifelong learning paradigm can extend the list of possible interactions. The system can take into account the possible feedback of the user or ask some additional questions to guarantee better outputs etc. As a result, the interactions will not only improve the current output of the system but provide additional information for the system to learn from. There are works which implement different interactions between system and user [102]–[104]. The definitions of different types of interactions will be discussed in the next section. It is also important to notice that this work focuses on the design and implementation of the interactions which can be specifically applied to speaker diarization systems.

According to the given statements we now give the definition of lifelong learning which will correspond to the needs of the ALLIES project. We do not pretend to provide final and universal versions of the definition, but propose another point of view focused on our domain of interest, i.e. speaker diarization.

Definition. Given a task, a machine learning expert, an automatic system and a human domain expert, Human Assisted Lifelong Learning (HALL) is the continuous learning process that aims at sustaining or improving the performance of the automatic system across time for the given task, by interacting with the human domain expert.

In this document, we define a "human domain expert" as a person who has knowledge about the final task the system is used for, and intrinsically knowledge about the data to process, but no specific understanding of machine learning.

At time t_0, an initial version of the automatic system, is set up by the Machine Learning expert, by training a version, M_0, of the model on a set, D_0, of initial training data. For the given task, this initial system reaches an error rate of $Err(M_0, D_0)$. Starting from t_0, the system adapts its model across time to reach a version M_{n-1} at time t_{n-1}. At time t_n, the system receives a new batch of data, D_n, to process. Based on its current model, M_{n-1}, and the incoming batch of data D_n, the system is free to interact with the human

domain expert in order to fulfill two goals:

1. process the data D_n to produce the best possible hypothesis;
2. update its model M_{n-1} to produce a new model M_n that is the most likely to produce the best hypotheses on future incoming data. As a minimum, it is expected that the error rate obtained using the new model $M_,n$ is lower than the one using previous models, i.e., $Err(M_n, D_n) < Err(M_{n-1}, D_n)$.

This definition relies on strong assumptions that are arguable. First, we assume that a current version of the model does not have to outperform the previous versions of the model on data from the past. Formally we don't expect:

$$Err(M_n, D_(n-i)) < Err(M_{n-i}, D_{n-i}) \text{ FOR } i \in \{1, n\} \tag{II.1}$$

Second, this definition requires the system to optimize the new model to match the nature of future incoming data that is by definition unknown. This assumption requires the developer of the system to make further assumptions regarding the continuous or stochastic nature of the incoming data by assuming a similarity between data across time to balance knowledge learnt from a distant past from the most recent. This question has been addressed in our work [105] but will not be discussed in this document.

II.1.2 Interactions with human in the lifelong learning process

In this section we discuss and define different types of interactions between the system and the human domain expert. Before going further we emphasize on the fact that the human domain expert is not expected to have any knowledge about machine learning and is thus only able to answer questions or provide information that are related to its domain of expertise: the task that is addressed by the automatic system. In the literature, there is no one single nomenclature for human-machines interaction [102], [106], [107]. In this section we propose our definition for the main types of human system interactions, examples of each type will be discussed in the following chapters.

In order to provide a nomenclature for the interactions, we need to have a classification criterion. We propose to use a criterion about who initiates the interaction. Using this criterion it is possible to separate three types of interactions:

II.1.2.0 - A Active interaction interactions with system initiative. A type of interaction for which the system initiates the interaction with the human domain expert and

learn from its answers. This interaction is based on the query exchange and corresponds to a 'teacher-student' relation. The teacher (human domain expert) gives a task to the student (system). The student analyses the data and initiates the interaction by asking a question (i.e. it generates a query). After the teacher (human domain expert) answers the question, the system eventually updates its models and generates the final hypothesis.

II.1.2.0 - B Passive interaction interactions with human initiative. In this case, the human expert acts as a teacher and gives a task to be performed by the system (student). The system then generates an initial hypothesis that the human expert analyses before suggesting a correction, i.e. generating the query. The system can take this information into account to correct and improve the rest of the hypothesis. In the end, the system provides the final hypothesis.

II.1.2.0 - C Cooperative interaction interaction with no initiator, it corresponds to systems that do not rely on query exchange and which do not encompass 'teacher-student' relations. Such systems are often based on the gamification of the learning process where the human and the system act in the same environment [104]. This aspect falls out of the scope of this work and will not be addressed further in this document.

II.2 Interactions with the human

The human assisted process can also be characterized according to the temporality it occurs in the life-cycle of the automatic system. In some scenarios, human assisted learning is only used during the initial training of the model [107]. Such systems are generally referred to as **active learning** systems. In other scenarios, the system can benefit from the human interaction during its entire life-cycle. Such systems are generally referred to as **online active learning** systems. Although the ALLIES project focuses on *online active learning*, we review in this section both active and online active learning. The literature offers more references on active learning than on online active learning. In this review, we pay attention to the wider domain of speech processing and we will discuss works related to speaker diarization in the next chapter.

II.2.1 Active learning

We previously defined 'active learning' (AL) as the concept of leveraging interactions via human interactions during the initial training of the system. In [107], the authors propose to apply active learning to minimize the amount of labeled data required to train the system. In this context, active learning can be described as follows. The machine learning expert trains a first version of the model using the available labeled data that might be limited and in relatively small quantity. Unlabeled data is processed by this model to obtain hypotheses. Using those automatically generated hypotheses, an acquisition module selects samples of unlabeled data which are expected to be the most reliable for future system training. The selected segments are then labeled according to the automatically generated hypotheses and added to the labeled data to train a new version of the model. It is expected that the performance of the obtained system will not be worse than the performance of the system trained with all unlabeled data. This approach allows to reduce the computational cost of the system by reducing the amount of data to train from. This approach generally involves 'active interactions' where the system starts by generating a query to the human domain expert. It was applied for many different tasks such as clustering [106], natural language processing [108], [109], dialogue systems [110], image processing [111] or health applications [112].

In speech processing, many active learning methods have been proposed for the task of automatic speech recognition [113], [114] and most of them use active learning strategies to reduce the labeling [103], [115], [116], but some work are more oriented on performance improvement [117]. Besides automatic speech recognition, active learning strategies have also been developed to reduce the cost of model training for speech activity detection [118], speaker recognition [119] and emotion recognition [120].

It is important to differentiate between active learning and semi-supervised learning. Semi-supervised approaches (see e.g. [121]) propose to use one or many systems in order to automatically label data which is in turn used to train another system. This has the advantage of involving only computers and machine learning experts and does not require human domain experts to label data (which can be very costly). Also there are attempts to combine both 'active' and 'semi-supervised' learning strategies [122].

II.2.2 Online active learning

We defined 'online active learning'(OAL) as systems that benefit from interactions with the human domain expert, not only during the initial training, but especially also during production time. Such systems are more oriented on performance improvement than cost reduction. After the initial training, an OAL systems can interact with the human domain expert to obtain additional information and use this information to improve itself (e.g. for domain adaptation).

The research domain in which online active learning is the most studied is the domain of dialog systems in which interaction with humans is part of the task. In their works [102] and [123] propose OAL strategies to improve the performance of a dialog system. The use of the term OAL in their works corresponds to the definition of 'human assisted lifelong leaning' proposed in the previous chapter. The strategy proposed in these works aims at using the human evaluations of some hypotheses as training signal within a reinforcement learning framework. Following the terminology introduced in Chapter II.1, these works correspond to 'passive interactions' as humans initiate the interactions. On the other hand in a more recent work, [124] used 'lifelong learning' and 'human assisted learning' term in the same context.

As online active learning is of our interest for the work in the ALLIES project, we pay here attention to the methods used for query selection in the literature. The works presented above use a confidence measure (obtained from the automatic system) to decide whether help from a human domain expert should be requested on a particular sample. A low value of confidence measure means that either the sample is hard and/or it is badly modelled by the system. Both cases require interactions with the human domain expert. An alternative concept of query selection is described in [125] and [126]. The authors propose to use a so called 'second order information' instead of using only the confidence of the system in the hypothesis. The authors also use a measure that describes how often the samples which are similar to the investigated sample have been seen by the system in a recent past. Another approach, described in the works of [127] and [128] proposes to combine outputs from different systems to better estimate the confidence measure in the proposed hypothesis. In [129] and [130], the authors propose to investigate the topology of the input data to select the most informative samples, which will then be annotated by the human domain expert.

The OAL paradigm can be applied to systems that are not neural network based. In their work [131] propose to apply the OAL in composition with 'semi-supervised learning'

to improve the performance of a Bayesian model.

II.2.3 Cooperative learning

This section reviews some examples of 'cooperative interactions'. In [104], the authors solve the traveling salesman problem using the ant colony optimization algorithm. The user can control one of the 'ants' and travel through the graph. The pheromone value of the ant under human control is augmented. Another example of 'cooperative interactions' proposed in [132] is based on gamification of the speaker recognition process. In this approach, the system can ask a speaker to utter words from a close vocabulary in order to improve the representation of the speaker. The Quality of the model is estimated via an automatic speaker verification system.

II.3 Human Assisted Speaker Diarization

This section provides a review of existing active learning methods for speaker diarization. First of all it should be noticed that in the context of interactions, the diarization task has a peculiarity that affects the interaction process. For example, in the case of image recognition, a data sample corresponds to a single image for which a corresponding label should be selected. Thus, the whole sample can be corrected with a single interaction between the system and the human domain expert. For speaker diarization, one sample consists of an audio recording which may have a long duration. The system's hypothesis is thus the mapping of speakers in this file. Such conditions makes labeling of the whole sample much more complex and costly, and labeling the whole sample does not provide much more information than partial labeling due to the multiple examples of speech of the same speakers in one file. For this reason, for speaker diarization, one interactions does not focus on selecting the best samples to be labeled but rather on providing additional information obtained from sub-parts of the samples.

In this chapter we describe the existing approaches for human assisted speaker diarization as well as the importance of cross-show speaker-linking for human assisted lifelong learning speaker diarization.

II.3.1 Human assisted withing-show speaker diarization

Human assisted speaker diarization is a rather new domain and only a few works dealing with this topic can be found in the literature. One of the first works including interactions with a human domain expert proposes a strategy which helps the human domain expert to chronologically correct the automatic diarization [133]. In this work, the diarization system generates an initial diarization hypothesis that the user confirm or correct by labeling all segments in a chronological order. When the human domain expert corrects a label for a segment, the system update its speaker representations and re-estimates the distances between the following segments. Such a system reduces the number of interactions required from the human domain expert to obtain an ideal diarization. An extension of this strategy has been recently proposed by the same authors [100], [134] where they not only work on constraining the clustering process, but also constraining the segmentation. In these works, the author presents a framework that allows to evaluate and improve the effectiveness of the human corrections of a speaker diarization system.

A highly relevant work is proposed by [135]. In this work, the authors propose a system which allows to perform speaker diarization with the help of a human domain expert. The authors implement human-system interactions to improve the clustering step only as they use the reference segmentation. In the proposed interaction mechanism, referred to as 'active interaction', the system proposes the human to listen to two speech segments and tell whether they are from the same speaker or not. The authors propose to interact with the human in two steps. The goal of the first step is to find out the number, N, of unique speakers in the recording. The goal of the second step is performed after clustering all data samples into N classes to verify and correct the clusters (i.e. improving their purity). To find out the number of speakers, the authors propose the following strategy. Segments are represented as vectors and distance are calculated between them. They select one random segment and mark it as the first unique speaker. Then, they iteratively select the farthest segment from all existing clusters and query the user by asking whether the selected segment comes from a speaker in one of the existing cluster. If the human answers that it belongs to one of the clusters, the system add the selected segment to this cluster. If the selected segment does not belong to any of the existing clusters, then a new cluster is created. The authors propose to continue this process until a limit of questions, defined by the human, is reached. After this limit is reached, the number of detected clusters is fixed, and all constraints obtained from queries are used during the following clustering of the rest of the segments. To improve the purity of the clusters, the authors propose to

select segments which should be verified by using expected speaker error (ESE) estimated by modeling a multivariate Gaussian distribution of the i-Vectors. The selected segment is then verified by querying it with the V (selected by user) most confident segments of its cluster (the ESE confidence measure is also used in this case). If the majority of queries returns negative answers, this segment is excluded from its clusters and a new cluster should be selected. To do so, the authors propose to use ESE to select the M (selected by user) most confident clusters and ask V questions to the user about each cluster. The drawback of this work is the number of question which should be answered by the human. The authors propose to ask from $0.1N$ to $0.5N$ questions per recording, where N is the number of segments in the recording. Such number is justified with the statement that to process recording manually human should perform N^2 comparisons. This statement is arguable as it is more reasonable to estimate this value as $Nlog(N)$. In Chapter IV.3.3, we propose a penalization of the DER, based on the estimation of N comparisons.

Part II

Protocols metrics and corpora

Chapter III

Existing corpora and metrics

In previous chapters, we described human-assisted lifelong learning speaker diarization. By nature, this task aims at addressing realistic constraints and to fulfill a need by interacting with human domain experts to improve the performance of incremental cross-show diarization. The complexity of this task and the involvement of a human in the loop makes it especially difficult to evaluate.

Experimenting with lifelong learning speaker diarization requires to collect specific datasets, to define interaction protocols between the human domain expert and the automatic system and to develop new metrics for evaluation of incremental active learning systems. These aspects, related to evaluation are discussed in this chapter. We first provide a critic of existing corpora for speaker diarization, existing metrics for speaker diarization and active learning in order to specify their lacks and to motivate the creation the ALLIES framework for evaluation.

III.1 Existing corpora

As mentioned earlier, lifelong learning speaker diarization is a sequential process during which the information obtained from one file can be used to update the model and process all following shows. For reproducibility purpose the evaluation of automatic systems requires to fix the order of the sequence of shows to process. Indeed, changing the order of the shows will modify the history of information available for the system when processing a given file and eventually lead to different final scores. Many ways to order the files could be considered depending on the topic of the study; it could even be interesting to evaluate the systems while re-ordering the files to provide contrastive results. As a first attempt in this field we decide to organize the show by chronological order. This ordering corresponds to the real use-case of such a system and also guarantees that aging of the speakers is consistent along the sequence of shows. Using the real timestamps, the date and time of the first broadcasting will define the processing order. This protocol is similar

to a real use case of data archiving.

Another important criterion to chose a dataset is the consistency of speaker labels across shows. As mentioned above, cross-show diarization is part of the life-long learning diarization process and thus, speaker labels must be kept consistent across the entire collection to allow cross-show speaker-linking.

Many speech corpora have been introduced by the diarization community with different purposes. Existing corpora offer a number of variability factors to study such as language, number of speakers, speech ratio, overlapping speech, and acoustic conditions. In this section we verify their applicability for lifelong learning speaker diarization, and summarize their properties in Table III.1. The rest of this section provides a comparison of existing corpora according to different criteria.

III.1.0.0 - A Language variability

Most of the corpora only include English speech like AMI [136], CHiME-5 [137], APOLO-11 [138], LibriCSS [139]. but a few other languages have been collected for speaker diarization evaluation, such as Chinese Mandarin in AISHELL-4 [140], Spanish in Albayzin [141] or French in REPERE [142], ESTER [143] and ETAPE [144].

III.1.0.0 - B Speaker and speech quantity

Estimating the number of speakers in a show is a difficult task [145], [146] that strongly conditions the final quality of speaker diarization. Assuming a known number of speakers strongly improves the performance of the diarization systems [137], [139], [147] while discovering this number in an unsupervised manner enables many more application scenarios but is still a very challenging task [148]. Regarding the quantity of speech, most of the existing corpora are collected from telephone conversations, TV, Radio or meetings and exhibit a high speech/non-speech ratio. On the opposite, the speech duration in the recordings of the APOLO-11 mission (as proposed in *Fearless Steps* corpus [138]) counts for only 36% of the total recording duration which requires to modify the prior of the models accordingly.

III.1.0.0 - C Overlapping speech

In the past few years, improvements of the speaker diarization systems has led to focus on the most difficult part of the data, making the detection of overlapping speech a major source of errors for the automatic systems [149], [150]. Overlapping speech ratio strongly

Table III.1 – Existing diarization corpora

Name	Language	Duration	# Speaker	Cross show speaker ID	Recording time for Lifelong
CALLHOME	Multilingual	20 h	2–7 Spk./file	No	No
AMI	English	100 h	3-5 Spk./file	No	No
Voxconvers	Mostly English	74 h	1-21 Spk./file	Yes	No
CHiME-5	English	50 h	4 Spk./file	Yes	No
APOLO-11	English	100 h	34 Spk./hour	Yes, but only 30 files	No
AISHELL-4	Mandarin	118 h	4-8 Spk./file	unknown	unknown
DIHARD3	Multilingual	67 h	1-7 Spk./file	No	No
LibriCSS	English	10 h	8 Spk./file	No	No
Albaizin	Spanish	569 h	Avg. 27 Spk./file	Yes, but only 166 speakers	Yes

varies across existing corpora. A corpus like AISHELL-4 [140] ecxhibits an overlap ratio of 18.2% while the CHiME-5 [137] does not include overlapping speech at all. The AMI corpus [151], that is widely used by the community includes a number of sessions for which participants followed a script in order to guarantee the appearance of overlapping speech. Such "simulated" overlapping speech could enable a finer analysis but is however less realistic.

III.1.0.0 - D Corpora size Deep learning algorithms are extremely data-hungry and require larger and larger corpora to train the models on. Additionally, the strong reduction of error rates observed during the last decades in speaker diarization requires larger or more challenging corpora to guaranty the significance of the results. For speaker diarization, the majority of available corpora do not include more than 100 hours. And recent new diarization corpora do not contain large-scale training data, but rather challenging testing conditions. Corpora like AISHELL-4 with high overlap ratio or Dihard with challenging acoustic conditions challenge the researchers to develop new approaches that are robust to those specific recording conditions.

Lifelong learning and incremental cross-show diarization are specific use-cases of speaker diarization for which large corpora are not publicly available. This was the motivation for the production of the ALLIES corpus.

Table III.1 presents the most frequently used corpora in the literature with their properties, focusing on criteria required for lifelong learning speaker diarization. An analysis of this table shows that lifelong learning diarization can not be performed on corpora such as CALLHOME, AMI, LibriCSS, DIHARD1, DIHARD2 and DIHARD3 due to the absence of the cross-show speaker labeling. The corpora which pass this criteria such as CHiME-5, Voxconvers and APOLO-11 can not be used for lifelong learning application due to the absence of information about recording time (timestamps). Indeed, it is impossible to find

the chronological order of the files.

The only corpus that passes both criteria is the Albaizin corpus which contains a large quantity of data but that exhibits a low ratio of speaker recurrence. A low ratio of recurrent speaker will make the cross-show diarization task close to an averaging of within-show diarization and thus less challenging for systems.

This analysis of existing speech corpora for the diarization task tells us that no corpus really satisfies the two principal requirements of lifelong learning diarization application: presence of the recording timestamps necessary for ordering the files and high recurrence of speakers across files (with same and unique labels). Those are the reasons why we introduce the ALLIES corpus, an extension of the existing French corpora released for the ESTER [143], REPERE [142] and ETAPE [144] benchmarking campaigns.

A detailed description of the ALLIES corpus is given in chapter IV.1 while the following chapter describes existing metrics used to evaluate speaker diarization and active learning.

III.2 Existing metrics

Another important part of the experiment setup is the metrics applied to evaluate the system. This section first describes the metrics used to evaluate the performance of speaker diarization systems before reviewing existing metrics that integrate the cost of active learning in different fields and especially in speech processing.

III.2.1 Standard metrics for diarization

The main evaluation measure to evaluate speaker diarization systems is the Diarization Error Rate (referred to as DER). It was proposed during NIST evaluation campaign [152] and represents the percentage of speech time wrongly labeled (lower is better). The DER computation is performed by comparing the hypothesis generated by the system with the ground truth (reference) diarization, created by human annotators. Considering that the speaker labels in the hypothesis do not represent the actual identity (names, surnames) of the speakers they should be first match with the identifiers from the reference. This matching is done to maximise the correspondence between hypothesis and reference [153]. The computation of the DER consists in summing three different errors: Miss, False alarm and Confusion.

Miss error corresponds to the duration of speech segments, which are present in the

reference, but were not detected as speech by the diarization system. Those segments are identified as silence or other kinds of disfluencies by the system.

False alarm error corresponds to the duration of segments which were labeled as speech by diarization system when in the reference there is no speech.

Confusion error corresponds to the duration of segments which are labeled with the wrong speaker label after optimal matching between hypothesis and reference. The errors provided by undetected speech overlapping, when two speakers speak at the same time but the system detects only one of them, is also counted as part of the confusion error.

The final DER measure is calculated as the sum of the three previously described errors divided by the total duration of speech in the reference, represented in Equation III.1.

$$DER = \frac{T_{miss} + T_{false} + T_{confusion}}{T_{total}} \tag{III.1}$$

Also it is a common practice to grant a tolerance of a few milliseconds to the segments borders in the reference due to the possible inaccuracy of the human annotator. The tolerance depends on the frequency of speaker changes in the recordings. For radio and TV shows recordings which we use in our research, it is common to use the tolerance of a 250 ms collar.

DER is not the only existing measure for speaker diarization evaluation. Several metrics can be found in the literature, namely precision, recall, F-score [154] and newly proposed Jaccard error rate [61]. Precision, recall and F-score were applied to speaker change detection task, and calculated using true positive, true negative, false positive and false negative detections with respect to the tolerance. Jaccard error rate is based on the Jaccard similarity index, a metric commonly used to evaluate the output of image segmentation systems, which is defined as the ratio between the sizes of the intersections and unions of two sets of segments. However DER is still the most used and standard metric for speaker diarization.

III.2.2 Metrics for active learning

The metrics presented in the previous section allow to evaluate the quality of a standard diarization system. In our work, we are considering an active learning process that involves human experts in the processing loop. It is thus then necessary to take into account the cost of human work during the evaluation. There are few propositions in the literature to do so for both offline and online active learning.

For offline active learning, the main method consists in representing the performance of the system as a score that is a function of the cost of interactions. [155]–[160]. The cost of interactions can be estimated in several ways and the optimal calculation strongly depends on the task under study. The most popular ways to estimate the cost of interactions is to quantify the amount of mouse clicks and keyboard strokes, as proposed in [161], or to calculate the average duration of one interaction and multiply it by the actual number of interactions as proposed in [134].

For online active learning, it is important to notice that the estimation of the human expert work is more important due to its direct influence on the final hypothesis. In the literature, the evaluation methods for online active learning speaker diarization systems are the same as for offline active learning. In their work [134] propose to separately evaluate DER and the estimated time spent by the human annotator. Another work calculates the number of interactions to represent the amount of human work. In [135], the authors count the number of questions asked by the system to the human domain expert and fix different limits to the number of questions permitted. Then, the authors compares the DER obtained with different maximum number of questions. In such case, it is hard to select the best system due to the necessity to compare two factors with different natures. Another approach have been proposed by [162]. They propose to continue interacting with the human annotator until the hypothesis is completely correct and they measure then the quantity of interaction needed to reach the perfect hypothesis. In their work, the authors named their metric as Minimum Supervision Rate (MSR). On one hand, this approach allows to encapsulate the metric in one scalar value, enabling a direct comparison between systems. The other metrics, which use separately a quality measure (such as DER for diarization) and an estimation of the amount of human work, make it more difficult to compare different systems. On the other hand, it is difficult to rank a system that generates a better hypothesis with more interactions and another system which is less greedy for interactions but showing lower performance overall because depending on the use case and the strategy, one would give more importance to the budget or to the accuracy of the diarization. Developing a system that requires too many human interactions to reach the expected quality. The system should know when to stop asking questions to guarantee that those questions will significantly improve the score. We have to distinguish between systems that aims a perfect hypothesis (whatever the cost is, see MSR) and systems that search for an optimal compromise (whatever it means) between interaction cost (i.e. number of interactions with human expert) and the resulting performance level.

Another drawback inherent to approaches which estimate the cost of human interactions by considering the time spent on the interactions or the number of mouse clicks is their strong dependency to the design of the user interface an to the users themselves. In our work, we don't address the issue of ergonomic interfaces and decide to develop a measure that considers both the final performance of the system and the cost of human interaction into one single scalar value to enable straightforward comparisons between systems. The following chapter, and more precisely Section IV.3, describes this approach.

Chapter IV

Protocols and metrics for ALLIES

As it was shown in the previous chapter, there is no publicly available dataset for that satisfies the requirements for the evaluation of lifelong learning speaker diarization. Also, the evaluation of lifelong learning systems is still an open question. This chapter introduces the ALLIES dataset, developed especially for lifelong speaker diarization. This corpus comes with protocols which we propose and the metrics to evaluate the active learning part of the lifelong learning strategy.

IV.1 The ALLIES corpus for lifelong learning diarization

The ALLIES dataset was specifically designed to satisfy the two main requirements of lifelong learning diarization, as described in previous chapters: time stamps for each audio sample so that the corpus can be considered as a temporal sequence of shows, and a high ratio of recurrent speakers in different recordings that could be leveraged by the cross-show speaker diarization system.

The ALLIES corpus is build on top of existing French corpora. It includes the corpora used during the ESTER [143], REPERE [142] and ETAPE [144] evaluation campaigns, with additional audio samples recorded over the years. Note that the new collected data has been precisely annotated for overlapping speech, providing the names of all speakers speaking at the same time as in other modern datasets. The ALLIES data includes 1,008 [1] French TV and radio shows collected from 7 Radio stations and 4 TV channels for a total duration of 612 hours and 46 minutes. 53% of this data is already annotated, which gives a usable amount of 328 hours and 21 minutes of recordings. The different parts of the ALLIES corpus are presented in Table IV.1 to compare the statistics of previously existing corpora with the resulting ALLIES dataset.

1. extended to 1,079 since this work has been done

Table IV.1 – Comparing ALLIES corpus with previously existing corpora.

Corpus	**ESTER**	**ETAPE**	**REPERE**	**ALLIES**
Number of shows	157	73	291	1,008
Average number of speakers per show	28.4	10.5	9.6	11.6
Number of unique speakers	3,059	688	1,518	5,901
Annotated time (h:m:s)	110:40:48	34:09:26	52:37:26	328:21:17
Speech ratio	0.97	0.96	0.94	0.96
Overlap ratio (%)	<1	3	4	3
Start date	1998-12-17	2010-02-03	2011-07-06	1998-12-07
End date	2008-12-02	2011-05-26	2013-04-24	2014-12-01

The ALLIES corpus provides 487 additional annotated shows, i.e., an additional duration of 130 hours and 53 minutes of annotated speech. The additional shows have been selected to maximise the temporal coverage on the time span and to consolidate series of shows to enable longitudinal studies of those series.

IV.1.1 Speaker statistics

As mentioned above, an important requirement for lifelong learning dataset is the high number of recurrent speakers across recordings. The corpus should however propose a high speaker variability to guarantee the generalisation of system's knowledge and their applicability on real world tasks. The ALLIES corpus satisfies both conditions: it includes 5,901 unique speakers recorded over 16 years, which guarantees the variability of speakers due to the high number of unique speakers as well as the intra-speaker variability due to the aging of the recurrent speakers. In order to respect the age of the speakers within the series of shows, TV and Radio shows are labeled according to the date of their first broadcast. It allows to determine the aging of the speakers appearing in recordings released at different dates. We do not deny the possibility of broadcasting archived recordings but it is a minor case due to the high percentage of news recordings which should be released straightaway after recording. The recurrent speakers (i.e. that appear at least in two different shows) may appear in different shows through the years, the longest 'lifetime' of a speaker in the ALLIES dataset being more than 15 years. This speaker appeared in 12 shows from 1998 to 2014. Another speaker appears in more than a hundred shows over a smaller period of time. Table IV.2 and Figure IV.1 provide a more detailed picture of the top recurrent speakers appearance across time.

While ensuring a high speaker variability, the ALLIES corpus also exhibits a high ratio

Table IV.2 – ALLIES speakers across time

#Speakers	Min #occurrences	Avg. recording period
1	146	1,107 days
10	27	965 days
50	5	1,502 days
1018	2	785 days

of recurrent speakers, appearing in different recordings. On average, 49% of the speakers present in a show have already been seen in a previous show, i.e. in a show with older broadcasting date. Taking into account the nature of the data, it is obvious that the

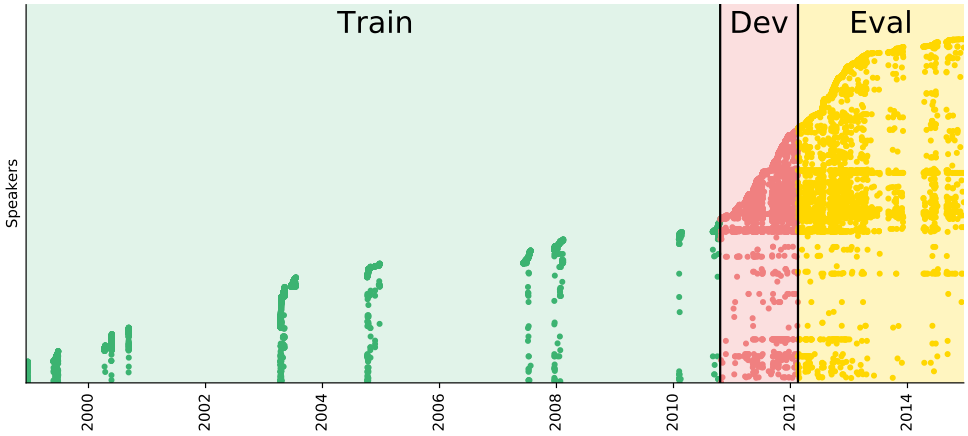

Figure IV.1 – Appearance of all recurrent speakers in the ALLIES corpus according to the recording time. Each horizontal line corresponds to a unique speaker and each dot represents one occurrence of this speaker in the recorded time (x-axis)

majority of recurrent speakers are presenters or journalists who work on a single series of shows with similar acoustic condition and speaking style. But in average 7% of the speakers are presented in more than one series of show and 2% of speakers are seen in shows from more than one channel.

Also, it is important to evaluate the systems on shows with different speaker turn duration, due to the huge impact on diarization performance that can cause different speaker turn durations. On average, speaker turn duration in broadcast news is longer than in other types of recordings, such as telephone conversations or meetings, as shown in [55]. For the ALLIES corpus, the average speaker turn duration is equal to 14.1 seconds

Table IV.3 – Global partitioning of the ALLIES corpus recorded from 4 channels and 19 shows series with their corresponding duration (and number of shows). All timestamps are given in hh:mm:ss format.

Type	Channel (dur.)	Show title	Total	train	dev	eval
TV (204:14:03)	BFM (29:25:53)	BFM Story	26:40:33 (49)	2:28:29 (3)	12:45:46 (25)	11:26:18 (21)
		Planète Showbiz	2:24:14 (73)	-	2:24:14 (73)	-
		Ruth Elkrief	0:21:06 (4)	-	-	0:21:06 (4)
	LCP (171:01:07)	Ça Vous Regarde	24:22:29 (45)	1:32:18 (2)	14:58:13 (27)	7:51:58 (16)
		Culture Et Vous	2:45:12 (87)	-	0:16:49 (8)	2:28:23 (79)
		Entre Les Lignes	25:32:35 (62)	0:52:47 (2)	10:36:20 (29)	14:03:28 (31)
		LCP Actu	21:34:51 (80)	-	-	21:34:51 (80)
		LCP Info	46:40:14 (156)	-	28:54:48 (97)	17:45:26 (59)
		Pile Et Face	25:57:08 (76)	2:13:07 (5)	14:40:09 (46)	9:03:52 (25)
		Top Questions	24:08:38 (104)	-	9:59:32 (46)	14:09:06 (58)
	TVME	-	2:09:23 (8)	2:09:23 (8)	-	-
	TV8	-	1:37:40 (4)	-	1:37:40 (4)	-
Radio (124:07:14)	Africa1	-	3:47:36 (18)	3:47:36 (18)	-	-
	Classique	-	1:00:04 (1)	1:00:04 (1)	-	-
	Culture	-	1:01:21 (1)	1:01:21 (1)	-	-
	France Info	-	12:00:43 (13)	12:00:43 (13)	-	-
	France Inter	-	54:56:52 (86)	52:59:09 (79)	1:57:43 (7)	-
	RFI	-	28:49:18 (38)	28:49:18 (38)	-	-
	RTM	-	22:31:20 (103)	22:31:20 (103)	-	-
Total			328:21:17 (1008)	131:25:35 (273)	98:11:14 (362)	98:44:28 (373)

with a large standard deviation of 27.46 seconds. This high value of standard deviation shows the wide diversity of genres of shows that is covered by the ALLIES corpus.

IV.1.2 Partitioning of the ALLIES corpus

To enable fair comparisons of systems on the ALLIES dataset we propose an evaluation protocol, that will be used for the ALLIES challenge(`https://www.chistera.eu/projects/allies`). This subsection describes the partitioning of the dataset, when the other aspects of the protocol are described in the following section.

The dataset is split into three disjoint parts for a **train** set, a **dev** set and an **eval** set. The splitting is done chronologically to simulate the real use case for a lifelong learning system and to provide the correct order to process the **dev** and **eval** files. The annotated data is separated on parts of approximately 40%/30%/30% for **train**, **dev** and **eval** sets respectively. The percentage was calculated in terms of annotated speech duration, and the partitions include complete shows. Table IV.3 lists channels, shows and duration for each partition of the ALLIES corpus.

To better represent the chronology of the ALLIES corpus, Figure IV.2 displays the cumulative duration of annotated data across time together with the time limits of the **train**, **dev** and **eval** sets.

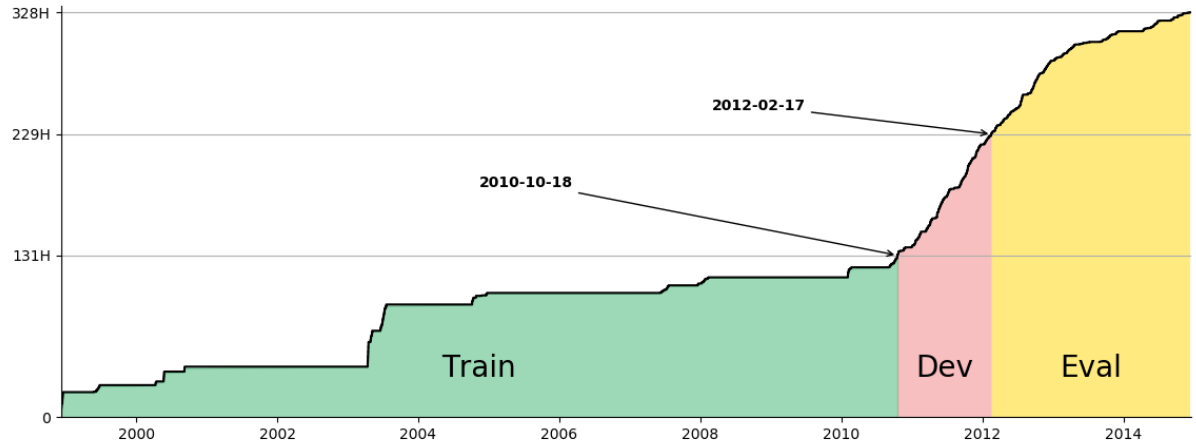

Figure IV.2 – Cumulative duration of annotated signal across time. The shows recorded before the 18^{th} of October 2010 are used as **train** data, shows between the 18^{th} of October 2010 and the 17^{th} of February 2012 are used as **dev** data and the remaining shows (recorded after the 17^{th} of February 2012 are part of the **eval** data

Due to historical reasons in the collection process, the duration of annotated data is highly variable across shows. The earliest data is less annotated than the recent data. Also the sampling of TV and Radio shows is not uniform across time. It explains why the **train** set runs over 12 years while **dev** and **eval** sets spread over 16 and 34 months respectively. Also it is important to notice that the amount of recurrent speakers in the **train** partition is lower than in **dev** and **eval** partitions, as depicted in Figure IV.1. This property will not affect the evaluation purpose of the corpus, but it can complicate the task for systems trained only on **train** partition of the corpus.

The number of speakers in the three partitions is also very different. In Figure IV.3 a Venn diagram displays the number of speakers for the three parts of the corpus with details of speakers overlapping in the different partitions. The ALLIES corpus contains 66 speakers who appear in the three parts of the corpus and 261 speakers who appear both in **dev** and **eval** parts.

We can conclude that the proposed dataset satisfies the main requirements of life-long learning speaker diarization and respect the tendency of modern datasets to provide challenging experimental conditions. The chronological split of the **train**, **dev** and **eval** partitions allows to provide experimental conditions close to real life scenario when new data appears continuously. Also, in the case of such splitting, the **dev** partition may contain show titles that are absent from the **train** partition and similarly, the **eval** partition may contain show titles that are not in both **train** and **dev** partitions. This allows to

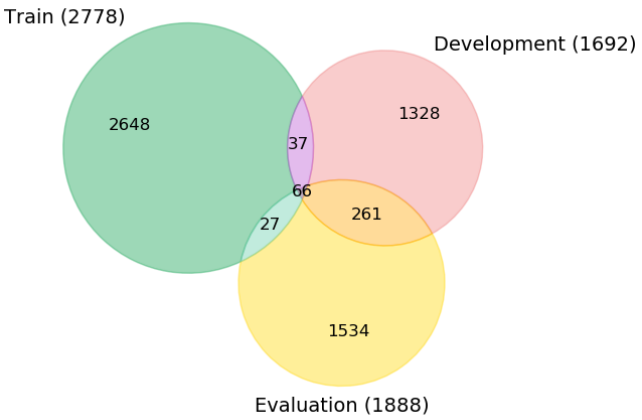

Figure IV.3 – Number of speakers in different partitions of ALLIES corpus and the number of common speakers.

verify the ability of the system to generalise and adapt to new conditions, which is one of the main goals of life long learning.

IV.2 The ALLIES protocols

The following sections present the protocols for baseline within-show and cross-show speaker diarization as well as for active correction process and life-long learning process. Active correction process (in Section IV.2.2) is a part of life-long learning during which a system can ask questions to the user to improve the final hypothesis and collect additional information for further system adaptation. The active correction protocol describes the possible interactions between an automatic system and a human domain expert in Section IV.2.2. This protocol aims at creating an environment for fair comparison of different interaction strategies. The lifelong learning protocol (in Section IV.2.3) describes the environment created to compare systems during a lifelong learning process similar to the one described in chapter II.

IV.2.1 Baseline protocols

Designed for lifelong-learning human assisted diarization, the ALLIES corpus can also be used for a classic speaker diarization task. The corpus is split into **train**, **dev** and **eval** sets that should be used in the standard following way. The initial system is trained on

the **train** partition and/or another publicly available dataset such as VoxCeleb. Then, the **dev** partition of the ALLIES dataset is used to determine the meta parameters of the system, such as the threshold used for within-show clustering or the threshold for cross-show speaker identification I.5. When the system is ready, it is evaluated on the **eval** partition of the ALLIES dataset.

IV.2.2 Active correction diarization protocol

This section describes the protocol designed to evaluate the active correction mechanism. This protocol is widely applied in the following chapters. The active correction diarization protocol proposes the same initial training as in previous case, the initial system should be trained on the **train** partition of the ALLIES dataset and/or an another publicly available dataset. After initial training, training data must not be used anymore, to only evaluate the active correction systems, without any modification of the core speaker diarization system.

During the following step the meta-parameters are fine-tuned on the **dev** partition of the ALLIES dataset. During this step the initial system processes the recordings one by one in the chronological order. After generating an initial hypothesis for each single show, the active correction system has access to this hypothesis and to the recording to prepare requests to the user. In this protocol we propose two types of requests.

The first type of request aims at correcting the clustering errors. The system suggests two speech segments with a duration of 3 seconds each for the user to listen to and asks the following question: *"Are the speakers in those segments the same?"*. We suppose that such type of request will provide maximum information with minimal time spent by the human. We also considered other variants of questions such as *"What is the name of the speaker speaking in this segment?"* or *"Indicate the borders of speech segment"*, but those may be too complex for the human annotators and would force them to spend a lot more time to answer.

For reproducibility of the evaluation, the answers for the questions are generated by a human expert simulator, which has access to the reference diarizations and thus provide the ground truth answers. This type of request is deeply studied in this document, the proposed active correction mechanisms are described in following chapters.

The second type of request aims at correcting the segmentation errors. The system proposes to the human annotator to listen to a fragment of recording and asks the question: "What are the borders of the segment around time t?". This type of request was not

studied in this work due to the larger cost of human expert work and the lower impact of segmentation on the performance of the system in terms of DER [134].

The answers to the questions are used by the system to (eventually) improve over the initial hypothesis and provide the final hypothesis. The active correction system can also be used to improve the cross-show speaker identification. In this case, the system has access to all the recordings and hypotheses that have been produced before and can provide to the user two fragments of speech to listen to. One fragment of speech should belong to the recording which is currently being processed while the other one belong to an already processed show. The human expert should answer the question: *"Are the speakers speaking in both segments the same?"*. The obtained answer should thus be used to modify the label of the speaker in the recording being processed to match the label of the corresponding speaker from the previously processed recording.

Data from the **dev** set can be used for development, to determine the hyper-parameters of the system but not to retrain or adapt the automatic system itself. The **eval** set is then used to fairly evaluate the system. While processing the **eval** set, adaptation of the automatic system and tuning of the hyper-parameters is forbidden to compare performance of the only active correction system. During the evaluation step, each request sent to the user is logged and used to measure the performance of the system. For evaluation it is proposed to use the DER, its cross-show variant if cross-show diarization is enabled and a penalized DER, a metric that takes into account the amount of interactions with the human domain expert that is further described in section IV.3.3.

IV.2.3 Lifelong-Learning protocols

In this section, we describe the protocol designed to evaluate the lifelong learning strategies with active correction mechanisms. In this scenario, the **train** set can be used the exact same way as described previously to train an initial system. After this initial training phase, the **train** data remains available for the system to perform any kind of adaptation, fine-tuning, etc...

For **dev** and **eval**, the extension of the previous protocol for lifelong-learning human assisted speaker diarization requires to strictly process the files in chronological order. Each show is processed as described in the previous protocol with possible interactions with the human domain expert. After the system produces its final hypothesis for one show, the system can use newly obtained knowledge to adapt itself. When processing one show, the system can make use of any information gathered on previously seen shows, in-

cluding models of previously seen speakers that are used for cross-show clustering (speaker linking across shows). The **dev** data can be used to optimize the hyper-parameters of the system that are then fixed when processing the **eval** set. For **eval**, two *lifelong* protocols are proposed depending on the state of the human assisted diarization system when starting processing the **eval** set. In a first scenario named *ALLIES-reset-lifelong*, it is possible to use the initial system trained on the **train** set with hyper-parameters tuned on the **dev** set. In a second scenario named *ALLIES-lifelong*, one can use a version of the human assisted diarization system that has already gathered knowledge by processing the **dev** set. In this former scenario, the system might have learnt about the speakers encountered in the **dev** set.

IV.3 The ALLIES metrics

This section describes the evaluation metrics existing for speaker diarization as well as our proposal for online active learning evaluation. First, we present the baseline metrics, then we present a general concept to measure the system performance which takes into account the amount of information provided by the human instead of only considering the time spent or its equivalent in terms of budget. In the end we describe the exact application of this concept to the speaker diarization task.

IV.3.1 Metrics for baseline diarization systems

The system can be evaluated in two different ways depending on the purpose of the evaluation. If the system does not include the cross-show speaker identification module or the experiment is meant to only evaluate the within-show performance of the system, the standard DER metric can be used.

IV.3.1.0 - A Evaluation of the cross-show diarization If the experiment is meant to evaluate the cross-show performance of the system, then both DER and DER cross-show metrics are applied to evaluate the system. The DER cross-show is an extension of the DER metric which instead of linking the hypothesis and reference of the different recordings separately, provides the linking between the concatenation of the hypothesis and references of the all corpus. In this case, if the same speaker in two different recordings is labeled differently by the system, then only the segments from one recording will be linked with the reference. This change allows to evaluate the efficiency of the within-show

clustering and cross-show speaker linking in one metric. In case of cross-show diarization, the recordings from **dev** and **eval** sets should be processed in the chronological order to preserve the order of appearance of the same speakers in different files. This protocol can be used separately or, like in this work, it can be used for comparison to estimate the impact of the active correction and active learning systems.

IV.3.2 Generic penalization framework for active learning

An ideal performance measure must provide a fair comparison of multiple systems using different types of user interfaces and different types of interactions. For this reason, we propose to take into account the cost of online active learning as a scalar metric given in the same unit as the performance score (error rate).

This penalisation can be equally applied to active learning with active or passive interaction (see II.1) and allows comparing systems that have different types of human/system interfaces. Moreover, measuring the cost of interaction in the same unit as the performance of the system allows summing the score and the cost of interactions in order to provide a unique measure that combines the evaluation of the final performance of the system together with the cost of the human assisted learning.

A first idea is to compute the cost of interaction according to the quantity of information given by the user. However, some evaluation functions (for example BLEU [163] for machine translation) are not linear. In this case, it is not possible to estimate the cost of interaction as a function of the quantity of data provided by the user. We propose to compute a penalisation term as the quantity of score that corresponds to the data corrected by the human expert during the process. For this purpose, it is proposed to compute two intermediate values: the corrected (S_{cor}) and the impaired (S_{imp}) scores. Computation of those scores is described in Figure IV.4

Let's assume that the system produces a first hypothesis and obtains the score S_{base} before applying any online active learning (see Figure IV.4-A). Then the human domain expert corrects (or is asked to correct) part of the current hypothesis. This corrected part of the hypothesis is shown in Figure IV.4-B and the resulting hypothesis obtains a score S_{cor}. Depending on the task, the part of the hypothesis that is corrected by the human domain expert might not be entirely wrong. For instance, in a speech transcription task, the human expert might correct a whole sentence while the current transcription of this sentence might include both correct and wrong words. The difference between S_{base} and S_{cor} corresponds to the score reduction (improvement on error rate) resulting from the

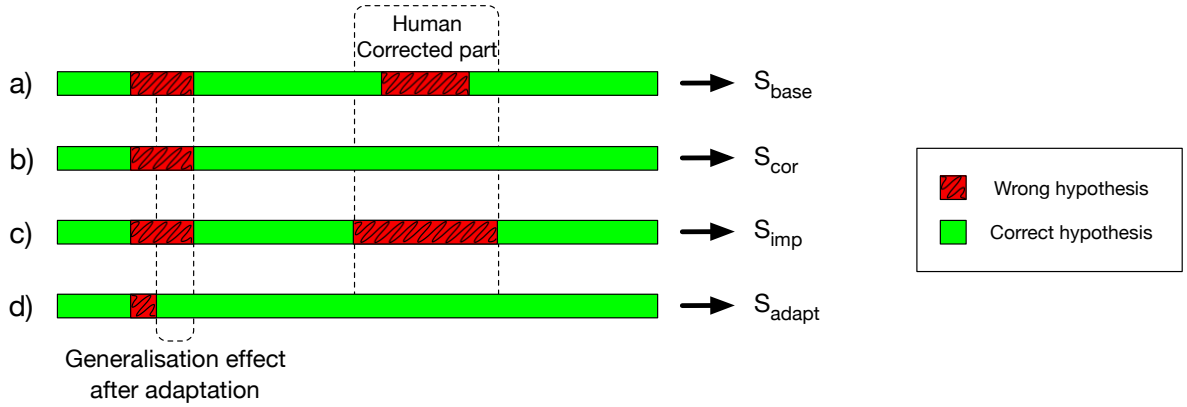

Figure IV.4 – An hypothesis (a) produced by the system contains correct parts (green) ans errors (red) and obtains a score S_{base}. During human assisted learning, the human applies (or is asked to apply) corrections on a part of the hypothesis that might be partially correct. To penalise the system, we introduce a first score, S_{cor}, computed on the corrected hypothesis (b) and a second score, S_{imp}, computed when the human corrected part of the hypothesis is replaced by a wrong hypothesis (c). After the system receives the human correction, it is allowed to generate a final hypothesis (d) by taking into account the correction. Hopefully, the system will generalise the knowledge learnt from the correction to other parts of the data and improve to obtain a score S_{adapt}.

corrections provided by the human domain expert only.

This difference, $S_{base} - S_{cor}$, does not reflect the cost of interaction as it is only related to the part of the corrected data for which the hypothesis was wrong. This is why we compute another score, S_{imp}, that is obtained on another version of the current hypothesis shown in Figure IV.4-C and where the hypothesis corresponding to the corrected part of the data has been modified with strictly incorrect values. The difference between the impaired score S_{imp} and the score obtained with the user correction (S_{cor}) gives the quantity of score that corresponds to the whole corrected part of the data and that could be considered somehow correlated to the cost of interaction. Eventually, the corrected hypothesis is fed into the system that reprocesses the data with regard to this correction and generates a new hypothesis as depicted in Figure IV.4-D where the system takes into account the correction and might leverage this new knowledge to generalise on other parts of the data to obtain a score S_{adapt}. The penalised score is then computed according to Equation IV.1

$$S_{pen} = S_{adapt} + (S_{imp} - S_{cor}) \tag{IV.1}$$

Note that a system which does not take the correction into account or ask for already

known information may be penalised twice: once in a sub-optimal S_{adapt} and once by the second term of Equation IV.1.

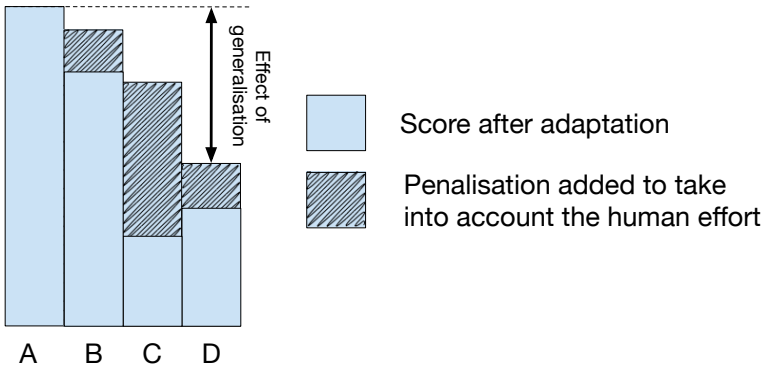

Figure IV.5 – Illustration of the effect of the penalisation method for different cases. Score (A) is the score obtained on the system hypothesis returned before human assisted learning, S_{base}, while the three other columns represent scores S_{pen} obtained on final hypotheses generated after performing human assisted learning with three different methods and using the proposed penalisation policy (note that lower scores are consider better). Plain parts of the columns B, C and D correspond to the score obtained after adaptation, S_{adapt} while the hatched area is the part added to penalise according to the cost of human interaction. Score (B) illustrates the case of a system that takes a limited benefit from a limited amount of user interaction, for (C) a great score reduction is obtained with a strong human interaction and (D) illustrates the case where a limited interaction strongly benefit the system adaptation. Note that the difference between S_{base} and S_{pen} (illustrated for score D on the figure) corresponds to the gain obtained by the system when generalising on the human corrections.

The effect of penalisation is illustrated in Figure IV.5. The score, S_{base}, shown in column A is obtained before online active learning. Columns B, C and D illustrate the penalised scores, S_{pen}, obtained for different cases of adaptation of the initial model (A). The plain part of columns B, C and D is the score S_{adapt}, obtained after using the information obtained from the human expert. By nature, it is supposed to be lower than score A. The hatched part of those columns represent the penalisation, computed as in Equation IV.1. The difference between score S_{base} and S_{pen} is the gain obtained when the system is able to generalise the corrections provided by the human expert to correct other parts of the data.

Penalisation helps to figure out the optimal model based on two parameters, final score and cost of interactions. In general the optimal system is a system with the largest generalisation power, but it can depend on the end-user needs. Some end-users may need

system witch provide the hypothesis without any errors, in this case MSR metric III.2 may be a better criterion.

IV.3.3 Penalized DER for speaker diarization

To take into account the amount of human expert work and represent both performance of the system and interaction cost in one scalar metric for the task of speaker diarization, we used the concept of penalized score as described in the previous section.

According to this concept, the penalized score can be expressed as in Equation IV.1, repeated here for convenience.

$$S_{pen} = S_{adapt} + (S_{imp} - S_{cor})$$

where S_{cor} is the corrected score computed on the corrected hypothesis and S_{imp} is the impaired score computed when the human corrected part of the hypothesis is replaced by a wrong hypothesis. To adapt the scoring method for the diarization task we propose to implement the following modifications. All terms of Equation IV.1 (S_{adapt}, S_{cor} and S_{imp}) are calculated using the DER III.2 formula as follows:

$$DER = \frac{T_{miss} + T_{false} + T_{conf}}{T_{total}} \tag{IV.2}$$

where T_{miss}, T_{false} and T_{conf} are respectively the duration of missed speech, non-speech considered as speech and wrongly classified speech. T_{total} is the total speech duration in the document. In our case T_{miss}, T_{false} and T_{total} are the same for the S_{adapt}, S_{cor} and S_{imp}, because we do not question the segmentation and thus the boundaries of the segments do not change. Based on that observation, we can rewrite Equation IV.1 and define the penalized DER as follows:

$$DER_{pen} = \frac{T_{miss} + T_{false} + T_{conf,adapt} + (T_{conf,imp} - T_{conf,cor})}{T_{total}} \tag{IV.3}$$

To estimate $T_{conf,imp} - T_{conf,cor}$ for the case of a single correction, we compute the duration of parts of segments the user is offered to listen to. To compare the speakers in two segments it is not necessary to listen the whole segments (they may be longer than a minute) but only a few seconds of each segment. According to this, we can compute the

penalised DER as:

$$DER_{pen} = \frac{T_{miss} + T_{false} + T_{conf,adapt} + N \cdot t_{pen}}{T_{total}} \tag{IV.4}$$

where N is the number of corrections applied to the document and t_{pen} is the estimated duration of parts of segments which were sent to the human expert during the active learning process.

In this research, we assume that the human expert needs to listen to 3 seconds of each segment to take a decision about whether they are from the same speaker or not.

IV.4 Conclusion

In the beginning of this part we've analyzed the existing datasets protocols and metrics for speaker diarization task, and highligted the aspects that required modifications or addition to work with lifelong-learning human assisted speaker diarization. Then we present our proposal which fill the existing gaps. The main ideas of our proposal can be described in the following points.

- Creation of a timestamped dataset which can be processed in the chronological order.
- Creation of a dataset with a high number of annotated recurrent speakers.
- Creation of protocols for active correction and lifelong-learning process.
- Creation of penalized metric which takes into account the amount of information provided by the human expert. The generic concepts of this metric can be adapted to many tasks and has been specially derived in our work for speaker diarization and machine translation to show that its possible to generalize.

The following parts will introduce our proposed methods to apply active correction for lifelong learning speaker diarization using the described dataset, protocols and metrics.

Part III

Active correction for lifelong learning diarization

Chapter V

Active correction for within-show diarization

Modern diarization systems achieve decent performance depending on the type of data they process [38] but those performances are often not good enough to deploy such systems without any human supervision. In some cases the error rate of the automatic diarization system is too high to meet the quality requirements of some business applications [61], [134]. One of the difficulties of deploying automatic diarization systems is associated with the nature itself of the task that is composed, at least, of a segmentation step and a clustering steps [1]. Errors at each step are accumulating and influence the final hypothesis. Another difficulty lies in the high variability of the data being processed. Automatic diarization system should be able to process audio files recorded in different acoustic conditions and having a large range of speaker number. However, [134] shows that producing a diarization hypothesis with an automatic system and having a human domain expert correcting it afterward is more efficient than having the human domain expert performing manual diarization from scratch.

According to this conclusion, we are interested in finding ways to optimize the correction process and enable possible improvements of the diarization system by making use of those manual correction in a feed-back loop manner. This process is referred to as the Human Assisted Learning (HAL) paradigm. Human assisted learning offers a way to achieve better performance by engaging an interaction between the automatic system and a human domain expert in order to correct or guide the automatic diarization process [164].

Amongst the different modes of human assisted learning, our work focuses on active learning where the automatic system, while processing an incoming stream of audio, is allowed to ask simple questions to the human domain expert [165]. The corresponding sys-

1. Note that the latest end-to-end architecture perform both in the same framework but the structure of the system is still divided in two parts

tem architecture is depicted in Figure V.1. Given an audio file, the human assisted speaker diarization system (HASDS) 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 hypothesis and possibly adapt the diarization system. This process iterates until reaching a stopping criteria. In our experiments, the human expert, depicted as part of the experimental protocol, will be simulated by automatically finding the answer to the question in the ground-truth reference.

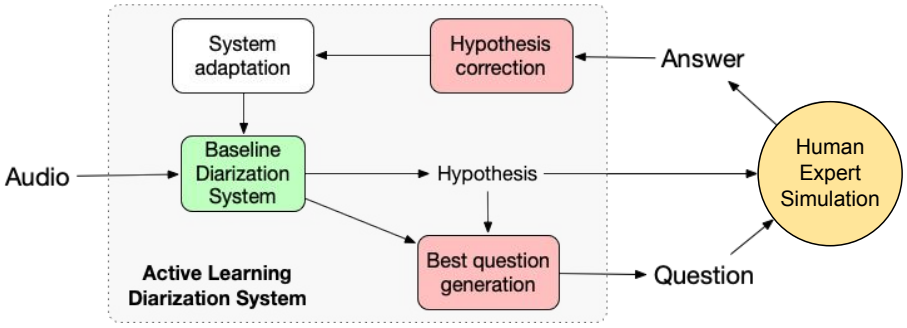

Figure V.1 – Life-cycle of a human assisted speaker diarization system.

To apply the described architecture to the diarization system, several questions need to be answered.

First, what questions can the system ask to the user? Many different kinds of questions can be asked, but not all of them are easy to answer. Moreover, since the goal is to use the answers from the user to update and adapt the system, one should make sure that the collected information can be used to do so.

Second, in which order should the system ask the possible questions? Since we are dealing with a synchronous process, the same set of questions in a different order could lead to sub-optimal HAL. Identifying the criteria that allow to ask the most relevant questions first seems important for the efficiency of the correction process.

Is it always worth asking more and more questions and how to decide when to stop asking questions? Indeed, obtaining information from the user is not free and may not necessarily lead to a better model (if redundant or non-informative questions are asked).

How to modify the hypothesis in order to take into account human expert answers? Once the answers to the questions are obtained, how to update the model accordingly to ensure a positive impact when processing the future shows. The protocol to adapt the model must be investigated because the results depend on the quantity of

available data, the fine-tuning of the hyper-parameters and the information contained in the answers.

We've already defined the type of questions that can be asked and the way of taking into account the answers from the human expert during the description of the protocols in section III.2. In this chapter we will address the remaining questions. The adaptation of the baseline system is out of the scope of this research.

In this part we start with the description of the baseline diarization systems, then we describe the question generation and hypothesis correction modules, and we finish with the presentation and comparison of the results of the proposed solutions.

V.1 Baseline diarization systems

For this work, we considered two different systems. Both systems share the same architecture but differ by the representation they use for the acoustic segments. The first system, as described below, is based on the original x-vectors standard and is strongly outperformed by a system from the next generation based on ResNet architecture (see below). All results reported in this document are provided for the two generations of systems to enable a comparison of the active correction benefit with respect to the baseline system performance. Indeed, it is important to make sure that the human assisted correction process is beneficial regardless of the performance of the automatic system. In the following chapters, for the sake of clarity, the systems will be simply referred to as *SincNet* and *Resnet* systems (respectively first and second baseline systems).

In this section we first describe our two baseline diarization systems (cf. Figure V.1). We also report the performance of those two systems on the ALLIES development set. Those results are used as reference in the remaining of this work.

V.1.1 Systems description

The baseline diarization systems consist of 5 main steps depicted in Figure V.2. The green color illustrates the steps which are common to both *SincNet* and *Resnet* systems, and the yellow color illustrates the steps in which there are differences. For both systems, a first step of segmentation splits the show into several segments and a first clustering is operated. Embeddings are then extracted for each cluster by using a *SincNet* or *ResNet* architecture depending on the system. Those embeddings are then compared using a

PLDA scoring in the *SincNet* system and a cosine similarity for the *ResNet* system. A final clustering is finally applied based on the scores from the previous step.

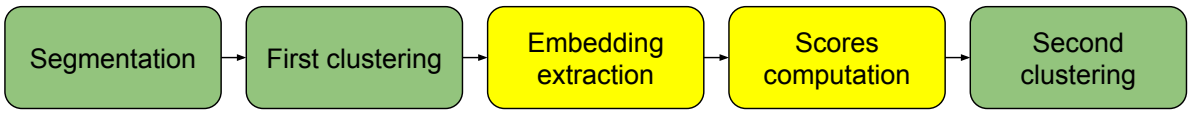

Figure V.2 – Pipeline of the baseline diarization system.

In the following section, we describe the segmentation step. Then, we describe the embedding extraction process and the corresponding score computations for both systems. The two clustering steps, which rely on the same algorithm, are detailed at the end of this section.

V.1.1.1 Segmentation

The segmentation step is essential for all speaker diarization systems. Due to its difficulty, it is nowadays very often performed first and refined after a step of clustering and acoustic modeling [82], [149] In our work, we focus on the correction of the clustering step, which led us to take two decisions to obtain results that are independent from the segmentation step: 1) all experiments that are performed on top of an automatic segmentation are also performed using the reference segmentation, 2) no re-segmentation is applied after the correction by the human. The benefit of the correction for the re-segmentation process has thus not been evaluated in this work and remains for following works.

V.1.1.1 - A Reference segmentation In this condition the segmentation step is considered perfect, i.e., borders of the speech segments are taken from the human annotations (reference). No other processing is done on top of it before running the clustering step. Results of the experiments with this condition will be referred(labeled) as 'REF'.

V.1.1.1 - B Automatic segmentation A simple voice activity detection, based on the energy of the signal energy, is performed to remove non-speech parts from the signal before clustering. Fixed length segments of speech are then compared using two sliding windows that cycle through the audio document to decide whether they were spoken by the same speaker or not. The Bayesian Information Criterion (BIC) is used as the measure (see subsubsection I.2.2.2). It was applied to the vectors of 13 MFCC features

Table V.1 – 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		

(see subsubsection I.1.1.1). Results of the experiments with this condition will be referred (labeled) as 'VAD'.

V.1.1.2 Embedding extraction

The *SincNet* Diarization System uses the *SincNet* extractor described in Table V.1. The dimension of the produced x-vectors is 100. The input of the *SincNet* model is 80 dimensional MFCC extracted on a sliding window of 25ms with a shift of 10ms. The *ResNet* Diarization System uses a Half-ResNet34 extractor (see Table V.2) to produce embeddings of size 256. As input, the *SincNet* 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

Table V.2 – 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	-	7,205

(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 [166] while the remaining of the system makes use of S4D [56].

V.1.1.3 Clustering

The clustering for both systems is performed in two steps. During the first step, hierarchical agglomerative clustering (HAC) is applied using the vectors composed of 13 MFCC and the BIC criteria (BIC-HAC clustering). The threshold of the first clustering is set to 3. This step allows to group short segments obtained with the segmentation step to form longer samples of speech that are further processed by the neural embedding extractor.

Embeddings are extracted from each segment and averaged to provide one single representation per BIC-HAC cluster. Embeddings are then compared using PLDA scoring

or cosine similarity depending for *SincNet* and *ResNet* respectively. In both cases, the obtained similarity measures are converted into a pseudo-distance that can be used by the HAC algorithm. The threshold for the final clustering is fine-tuned using the development part of the ALLIES dataset for both systems and both segmentations (REF or Automatic).

V.1.2 Baseline results

In this section we present the results of the baseline diarization systems, *SincNet* and *ResNet*, with two different segmentation conditions REF and VAD. Table V.3 contains the results on both development and evaluation splits of the ALLIES dataset. The threshold of the second HAC clustering, fine-tuned on the development partition, is given in the third column of Table V.3. The performance of all systems are reported in terms of DER.

Table V.3 – Performance of the baseline diarization systems in terms of DER. The threshold of the second HAC clustering, fine-tuned on the development partition, is given in column 3 as ~~it fine-tuned on the DEV set and used on the EVAL set.~~

System	Segmentation	Threshold	DER Dev	DER Eval
SincNet	REF	47	17.77	13.38
SincNet	VAD	45	19.07	20.20
Resnet	REF	0.23	14.12	<u>10.63</u>
Resnet	VAD	0.21	14.97	16.74

We can see in Table V.3 that overall, the *ResNet* system obtains the best performance. The best performance of the ResNet system are due to the quality of its embeddings that enable a better discrimination between speakers. Using the REF segmentation results in lower DER by a large margin. This is expected as the VAD segmentation tends to over-split the signal, generating many more segments and making it harder for the diarization system to cluster afterward.

V.2 Generating question and integrating answers

The proposed Human Assisted Speaker Diarization System (HASDS) is depicted in Figure V.1 and includes five modules. A fully automatic baseline diarization system, a question generation module, a human expert simulation, a correction module and an adaptation module.

The **baseline diarization module** is described in section V.1, it generates the first hypothesis which is necessary for future steps.

The **question generation module** analyses the hypothesis generated by the baseline diarization module, generates a request and sends it to the human domain expert who is here, for the sake of reproducibility replaced by a simulation module.

The **human domain expert simulation module** answers the question by searching the reference diarization and sends the result to the hypothesis correction module.

The **hypothesis correction module** uses the newly obtained information to improve the hypothesis generated by the baseline diarization module and sends the new hypothesis to the adaptation module.

The **adaptation module** may modify the acoustic model of the baseline diarization module.

In this chapter, we will describe our contributions to the question generation and hypothesis correction modules. We will especially address the issue about the type, the order and the number of questions that should be asked to the human expert.

The errors of the baseline diarization module can be grouped into two types, related to the two main steps of the diarization process. Segmentation errors occur when the errors can be directly associated with a wrong segmentation of the signal leading to wrong segment borders. Clustering errors occur when segments are not correctly grouped.

As it has been shown in [134] that clustering errors are the most harmful in terms of performance, we decided to focus only on clustering errors, as mentioned in the protocol description (see section IV.2). According to this, we decided to perform initial experiments with the automatic diarization system available at LIUM at the time, namely *SincNet* (see section V.1), using the borders of the speech segments from the reference (REF) segmentation, i.e. the segmentation step is considered perfect.

It is important to keep in mind that both our automatic diarization systems include two clustering steps. Having two consecutive clusterings 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 but to only apply active correction to the second clustering step while considering the BIC-HAC clusters as frozen. This choice has the advantage to reduce the correction to a simpler HAC-tree correction process. Another drawback is that errors from the BIC-HAC clustering will not be corrected (see Figure V.3) and the purity of those clusters is thus very important.

On the final step of the baseline diarization system we obtain a dendrogram as de-

picted in Figure V.3. Each speaker is represented by a specific color. The horizontal axis

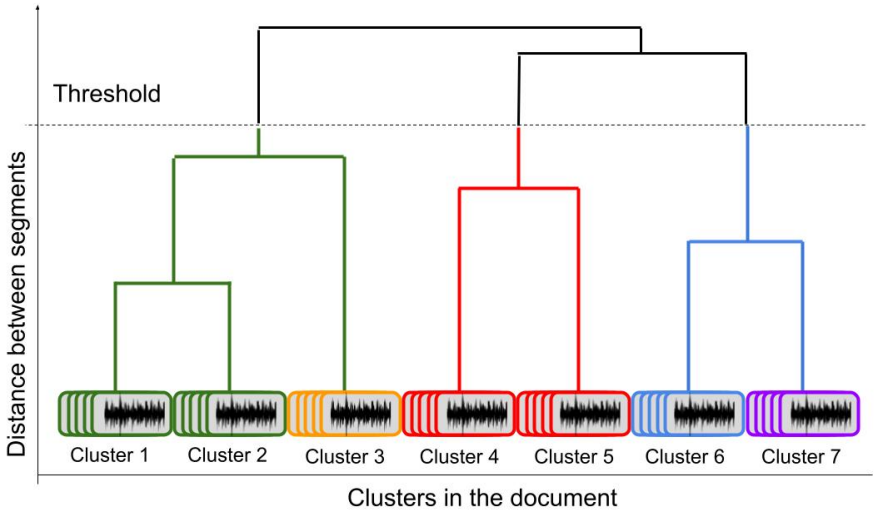

Figure V.3 – Illustration of the dendrogram obtained on the final step of the baseline diarization system.

corresponds to the clusters obtained during the first clustering step. Those clusters are considered pure and are not questioned. The vertical axis represents the distance between the clusters. The distance can be computed in several ways, this is described in subsection V.2.4. The dendrogram thus, hierarchically links the closest clusters together. The threshold is the parameter which plays an important role in deciding how the clusters are grouped or kept separate, it is fine-tuned on the development set.

V.2.1 What question to ask?

Working on the clustering dendrogram offers the advantage of locating the possible questions at each node of the dendrogram as a binary question questioning the correctness of the node. We decided to explore the potential of simple binary questions in order to minimize the time of human expert work, see section III.2 for more details. We propose to ask questions to the human domain expert in order to decide whether a clustering made at a given node is correct or not. As each branch of the considered node might already regroup several speech segments, we propose to select one segment from each cluster that are considered for merging. The human domain expert is thus asked to listen to those two segments (or part of them) and to determine whether they've been produced by the same

speaker or not. This approach allows to improve the clustering with only a small amount of time spent by the human expert. To implement the proposed strategy, it is necessary to take several aspects into consideration: (i) the order in which the nodes should be verified, (ii) the maximum number of nodes it is reasonable to verify (stopping criterion), (iii) the method to select the segments which will be listened to by the human domain expert.

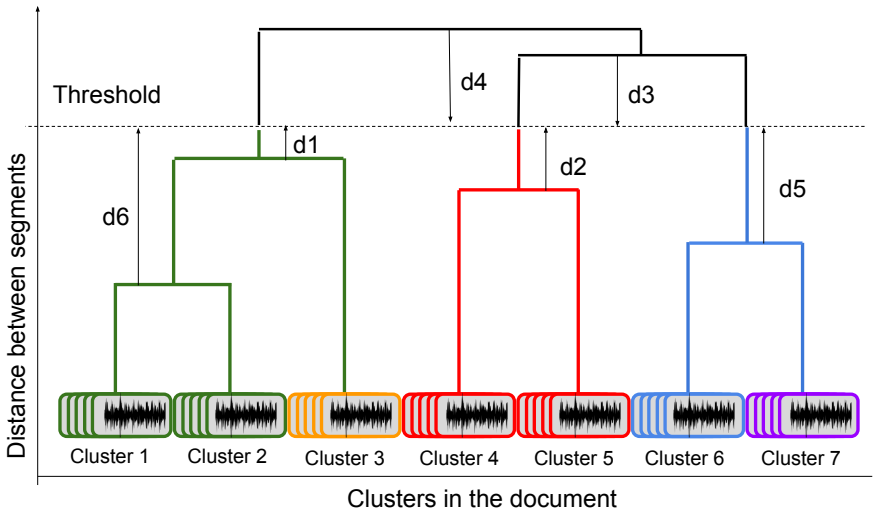

Figure V.4 – Dendrogram obtained on the final step of the baseline diarization system.

V.2.2 Node order

To define the order in which the nodes will be verified, we assume that decisions (merging or keep separate) on the nodes that are closer to threshold are less reliable than the decision on those nodes that are further from the threshold. To illustrate this point on Figure V.3, the system is more confident in merging *cluster 1* and *cluster 2* than when merging *cluster 4* and *cluster 5*. This is assumed from the fact that the node joining *cluster 1* and *cluster 2* is further from the decision threshold (horizontal line on Figure V.3) than the node joining *cluster 4* and *cluster 5*. According to this assumption we propose to verify nodes in an order related to the confidence of the automatic system. To do so, we rely on the distance between the threshold and each node, referred to as *delta* to differentiate with distance between x-vectors. Examples of those *delta* are labeled $d1$ to $d6$ on Figure V.4. Nodes are ranked in increasing order according to their absolute *delta* value. We propose to ask questions about the nodes in this order.

V.2.3 Stopping criteria

Remember that for a given node, the human domain expert is given two speech samples to listen to (one selected from each side of the current node) with the following question: "Are those two speech samples spoken by the same person?" To find out the optimal number of questions to ask, we use the same assumption and try to find out the borders of zone with low confidence nodes. To do so we implemented two different criteria.

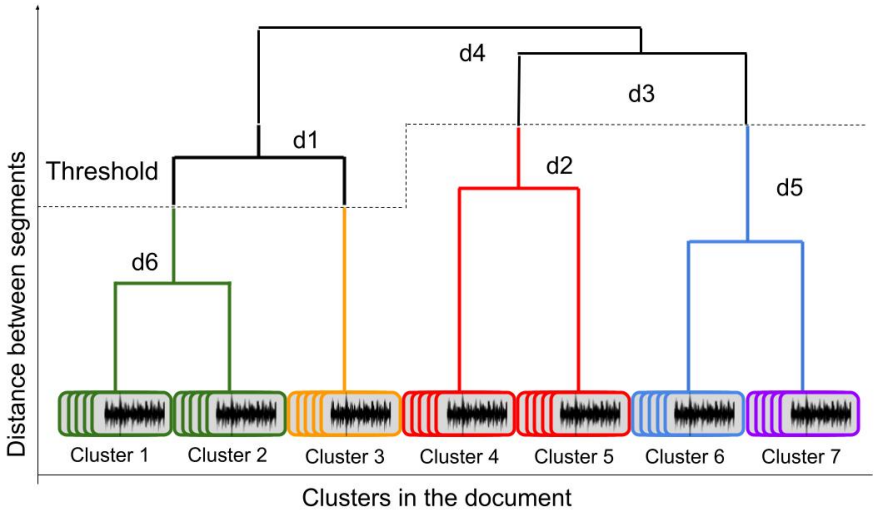

Figure V.5 – Dendrogram obtained on the final step of human assisted correction system using two confirmation criterion.

First, a **Two confirmation criterion (*2c criterion*)** is illustrated in V.5. For this criterion, we assume that if a node located above the threshold is confirmed by the human expert (i.e., when presented the two speech samples and ask the question, the human domain expert answers "no"), then the other nodes above the current one, i.e., with higher values of *deltas*, will not be investigated.

Similarly, if one node located below the threshold is confirmed by the human domain expert (i.e., the human domain expert considers that both speech samples belong to the sale speaker), the the other nodes, lower in the dendrogram, will not be investigated.

In the example depicted on figure V.5, 3 nodes was verified. First, node $d1$ was verified and was found wrong. Then node $d2$ was verified and confirmed, this result stops us from further verifying nodes $d5$ and $d6$. Then node $d3$ was verified and confirmed, this result stop us from further verifying node $d4$. At this moment, the correction process stops and

the corrections are applied on the dendrogram, in this case only $d1$.

In this case we find out the upper and lower bounds of the zone with low confidence, by approving two nodes with different decisions. But this zone can have non-linear borders (have different borders for different branches).

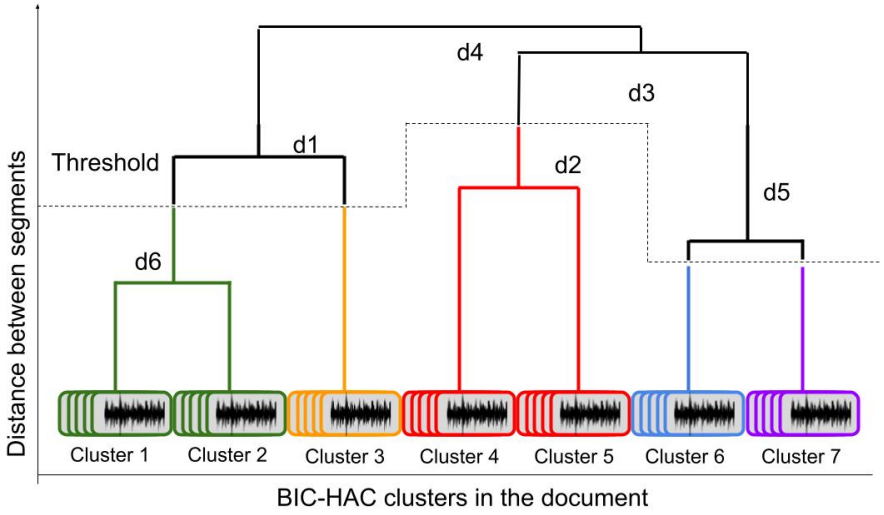

Figure V.6 – Dendrogram obtained on the final step of human assisted correction system using two All branches criterion.

For more flexibility, we propose a second criterion, a criterion exploring the tree per branch and referred as *All*. The result of using this criterion is illustrated on Figure V.6. Nodes are still processed according to their ranked *delta* but the dendrogram is explored in more details. In this case confirmation of the node do not stop us from verifying all other nodes on the same side of the threshold. Indeed, it only stops us from verifying nodes that are directly connected with the confirmed node (i.e., that are in the same branch).

On figure V.6), the first three nodes processed are the same as for *2c criterion*. After confirmation of node $d3$, we skip node $d4$ (because $d3$ was confirmed as a split) and we verify the node $d5$, which we find out wrongly merged. Then we also verify the node $d6$, because there was no confirmed nodes in this branch. The *2c* criterion relies on a high confidence on 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.

V.2.4 Segment selection methods

As mentioned previously, we consider that the HASDS 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*. To do so, we investigate 4 *sample* selection methods and 2 additional methods to find out the lower and upper bounds of the system performance:

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 segment's x-vectors.

Max / Min selects the couple of segments at once, one from each branch, with the lowest (max) or highest (min) similarity in terms score.

Random is a contrastive criterion, a random segment is selected from each cluster (statistics from this method are consolidated by repeating experiments 10 times).

Ideal correction method was considered to establish a lower bound of the system performance. When a node has been chosen to be investigated, the optimal decision (correct or wrong) is found by looking at the ground truth (reference) to maximize the gain in terms of DER.

V.2.5 Hypothesis correction

During the active correction process, the hypothesis correction module saves all answers given by the human expert. When the question generation module reaches its stopping criterion, the hypothesis correction module modifies the diarization hypothesis according to the human domain expert answers. The separated clusters obtain new different labels, and the merged cluster are given the same label.

V.3 Results

As mentioned in previous chapters, we started this PhD by using the *SincNet* system for our experiments with the reference segmentation (REF) and the real automatic segmentation (VAD). We then achieved better performance with the *ResNet* system and apply the same active correction strategies to this second baseline system. In this chapter we present the results in the same order.

V.3.1 SincNet results

In the following tables(Table V.4 and Table V.5) we present the results obtained on the *SincNet* baseline with the reference segmentation. The active correction system was tested on both (Dev and Eval) partitions of the dataset. This table provides the results obtained with both proposed stopping criteria: two confirmation criterion(2c criterion) and exploring tree per branch criterion (ALL). For each stopping criteria, we tested all 6 segment selection methods. The results are presented with the DER metric, the number of questions asked to the human expert per hour of speech and penalized DER. The results of the random segment selection method are represented in the form of the averaged value and standard deviation of 10 experiments with different seeds.

Table V.4 – Performance of the active correction strategies on *SincNet* baseline with reference segmentation on the 'Development' partition

Stopping criteria	Selection criteria	DER	N questions per hour	DER penalised
Baseline	-	17.77	-	-
2C	Ideal	14.93	6.85	16.07
2C	Longest	15.29	7.00	16.45
2C	Cluster center	15.66	7.03	16.83
2C	Min	18.8	7.72	19.8
2C	Max	18.57	6.67	19.68
2C	Random	17,29 ± 0,25	688,9 ± 7,88	18,45 ± 0,26
ALL	Ideal	13.81	25.56	18.07
ALL	Longest	14.29	25.55	18.54
ALL	Cluster center	15.19	25.60	19.45
ALL	Min	18.99	26.32	23.37
ALL	Max	19.22	25.13	23.40
ALL	Random	17,52 ± 0,4	2533±3,6	21,79±0,4

Table V.5 – Performance of the active correction strategies on *SincNet* baseline with reference segmentation on the 'Evaluation' partition

Stopping criteria	Selection criteria	DER	N questions per hour	DER penalised
Baseline	-	13.38	-	-
2C	Ideal	10,58	7.11	11,77
2C	Longest	11,06	7.07	12,24
2C	Cluster center	10,86	7.12	12,05
2C	Min	12,28	7.72	13,57
2C	Max	14,22	7.01	15,39
2C	Random	12,55±0,15	718,5±9,3	13,76±0,15
ALL	Ideal	9,39	27.14	13,91
ALL	Longest	9,96	27.16	14,49
ALL	Cluster center	9,82	27.17	14,35
ALL	Min	11,82	27.63	16,43
ALL	Max	14,64	26.32	19,03
ALL	Random	12,28±0,2	2689,2±3,96	16,81±0,2

In both tables V.4 and V.5, we can observe that 'Longest' and Cluster center' segment selection methods show much better performance than the 'Min' and 'Max' methods. Also 'Longest' and Cluster center' segment selection methods exhibit performance that is slightly worse than the performance of the 'Ideal' segment selection method. We can also notice that the 'Min' and 'Max' segment selection methods under-perform the 'Random' segment selection method and leads to the degradation of the performance even in terms of DER. We explain the observed behavior with the next statement. The clusters, from which the segments are selected, are obtained from the BIC-HAC clustering and they are not completely pure. In case when cluster contain, for example, 90% duration of Speaker1 and 10% duration of Speaker2 we are interested in identification of this cluster as Speaker1. In other words, we would like to select a segment containing speech from Speaker1. In case of 'Longest' segment selection method we select the longest segment, which rises the possibility of selecting speech from Speaker 1, as most of the speech in this cluster belongs to this speaker. In case of 'Cluster center' segment selection method we select the segment, representation which is closer to the center of the cluster and with the highest probability to belong to Speaker1. But for the case of the 'Min' and 'Max' segment selection methods the selected segment will be located far from cluster center, on the border of the cluster. In this case there is a higher probability to select a segment that contains speech from Speaker2 and will lead to a wrong interpretation of the answer from the human domain expert.

Comparing the performance of the different stopping criteria we can observe, that in terms of DER the exploring tree per branch stopping criteria(ALL) out-performs the two confirmation(2C) stopping criterion, but the number of question asked by the 'ALL' stopping criteria is much higher. In this case the DER penalized for the '2C' stopping criteria reflects this behavior .

Here we can conclude that the 'Longest' and 'Cluster center' segment selection methods perform better, for the following experiments will be discussed only their results. The results of the other segment selection methods have been found consistent with this conclusion. Talking about stopping criteria, we can not conclude the domination of one of the criterion because of the respect of different possible user policies. So in the remaining of this document we present both criteria.

In Table V.6 we represent the results obtained on the *SincNet* baseline with the VAD segmentation. The active correction system was tested on both Dev and Eval partitions of the dataset. This table presents the results of the 2 proposed stopping criteria: two confirmation criterion(2c criterion) and exploring tree per branch criterion (ALL). For each stopping criteria we tested 'Longest' and 'Cluster center' segment selection methods. The results are presented in terms of DER, number of questions asked to the human expert per hour of speech and penalized DER.

Table V.6 – Performance of the active correction strategies on *SincNet* baseline with segmentation based on VAD

Data	Stopping criteria	Selection criteria	DER	N questions per hour	DER penalised
Dev	Baseline	-	19.07	-	-
Dev	2C	Longest	17,22	7.95	18,54
Dev	2C	Cluster center	17,56	8.12	18,91
Dev	ALL	Longest	16,83	29.97	21,82
Dev	ALL	Cluster center	17,34	30.18	22,37
Eval	Baseline	-	20.20	-	-
Eval	2C	Longest	18,37	8.53	19,79
Eval	2C	Cluster center	18,56	8.78	20,02
Eval	ALL	Longest	18,04	34.11	23,73
Eval	ALL	Cluster center	18,19	34.23	23,90

In Table V.6, we can observe that the 'Longest' segment selection method outperforms the 'Cluster center' method. Comparing the stopping criteria, we observe the same situation as when using the reference segmentation: the 'ALL' stopping criterion outperforms the '2C' stopping criterion in terms of DER, but in terms of penalized DER, the '2C'

criteria shows better performance.

V.3.2 ResNet results

In (Table V.7 and Table V.8), we represent the results obtained with the *ResNet* baseline with both REF and VAD segmentations. The active correction system was tested on both (Dev and Eval) partitions of the dataset. In the table presented results of 2 proposed stopping criteria: two confirmation criterion(2c criterion) and exploring tree per branch criterion (ALL). With each stopping criteria we tested 'Longest' and 'Cluster center' segment selection methods. The results are presented with the DER metric, number of question asked to the human expert per hour of speech and penalized DER.

Table V.7 – Performance of the active correction strategies with the ResNet baseline and REF segmentation

Data	Stopping criteria	Selection criteria	DER	N questions per hour	DER penalised
Dev	Baseline	-	14.12	-	-
Dev	2C	Longest	12,78	21.65	16,39
Dev	2C	Cluster center	14,56	21.29	18,11
Dev	ALL	Longest	11,93	27.09	16,45
Dev	ALL	Cluster center	12,71	26.90	17,19
Eval	Baseline	-	10.63	-	-
Eval	2C	Longest	9,42	24.02	13,42
Eval	2C	Cluster center	9,42	24.17	13,45
Eval	ALL	Longest	8,15	29.96	13,14
Eval	ALL	Cluster center	8,58	29.98	13,58

In the Tables presented above, we observe that we achieve to improve the DER for both REF and VAD segmentations on both data partitions, even with a good baseline such as the *ResNet* system. In this cases the high number of questions asked to the user is reflected by the penalized DER.

For both baseline systems considered in this work, the proposed approach leads to significant improvement in terms of DER. An additional experiment allows us to establish the benefit of our human assisted correction process compared to a fully manual correction. Based on the automatically generated hypotheses, we computed the duration of speech signal that a human annotator would have to listen to in order to get the same improvement in terms of DER when correcting the files in a chronological order.On the evaluation partition, using the VAD segmentation, a human annotator has to listen to

Table V.8 – Performance of the active correction strategies with the ResNet baseline and VAD segmentation

Data	Stopping criteria	Selection criteria	DER	N questions per hour	DER penalised
Dev	Baseline	-	14.97	-	-
Dev	2C	Longest	16,04	22.84	19,85
Dev	2C	Cluster center	17,32	22.35	21,05
Dev	ALL	Longest	14,73	31.75	20,02
Dev	ALL	Cluster center	15,65	31.56	20,91
Eval	Baseline	-	16.74	-	-
Eval	2C	Longest	18,38	24.47	22,46
Eval	2C	Cluster center	17,94	24.48	22,02
Eval	ALL	Longest	15,78	35.98	21,78
Eval	ALL	Cluster center	16,17	35.80	22,14

26.80 hours of speech before bringing the DER down from 16.74% to 15.78% while our approach only requires the human domain expert to listen to 5.93 hours of speech.

Also we can observe that improvement achieved with the proposed systems depends on the quality of the segmentation. When using the reference segmentation, we achieve much higher relative DER improvement $29,82\%$ compared to $19,3\%$ using SincNet baseline and 20.79% compared to 5.73% using ResNet baseline. This suggests that improving the segmentation will lead to further improvement for the active correction system performance.

Also we noticed that the 'Longest' segment selection method outperforms the 'Cluster center' segment selection method in all cases, that allows us to focus on the 'Longest' segment selection method for the following part.

V.3.3 Conclusion

In this part we described our proposal for the within-show active correction system. For within-show the human-assisted system is based on the the analysis of the dendrogram obtained during final step of hierarchical agglomerative clustering, and then asking questions to human domain experts which allows improve the dendrogram. For this task we tested various strategies of question selection, which segments should be provided to user to compare. Also we tested different stopping criteria to solve the question when it is not reasonable to continue asking questions. For the within-show diarization, we achieved to reduce DER up to 18,83% relatively and penalized DER up to 9,94% relatively. The results on penalised DER can be interpreted as we achieved to correct almost 10% of

errors by generalisation of information obtained from human domain expert.

Chapter VI

Active correction for cross-show diarization

As mentioned in the previous chapters, cross-show speaker linking is essential for lifelong-learning speaker diarization. In this chapter, we first present a cross-show speaker-linking approach that will be regarded as a baseline for our work. We then describe the proposed human-assisted active-correction process that is eventually evaluated and analysed at the end of this chapter.

VI.1 Cross-show speaker linking

In this section, we propose an automatic baseline method for cross-show speaker-linking. After a detailed description we present the results of this system that will serve as reference for human-assisted cross-show speaker-linking.

VI.1.1 Introduction to incremental cross-show diarization

Cross-show speaker diarization can be performed in two ways: global and incremental [95]. Global cross-show diarization can be applied on a finite set of shows that are first processed independently to be segmented and to detect distinct speakers through a within-show speaker diarization process. In a second step, a global clustering is performed to cluster all speakers from all available shows and link recurrent speakers across shows.

When all shows are not available at once, for instance for the case of a series of daily TV shows that grows over time, an incremental cross-show speaker diarization process must be performed. We assume that at time T, a number N of shows has been already processed by the automatic system to produce a cross-show diarization. Along this process, a database of *known*-speakers has been produced. When receiving a new show to process $(N + 1)$, a within-show diarization process is performed before all detected speakers are

compared to the *known-speakers* from the database in order to detect recurrent speakers and link them with their previous occurrences in the database. The database of *known-speakers* is then updated and used to process new incoming shows. Along this process we chose to forbid the linking of two *known-speakers* together and the linking of two speakers from the within show diarization hypothesis of the current show. The reason for this choice is that we chose not to question the within-show diarization at this stage nor to reprocess all previous shows from the past.

The incremental cross-show diarization is more complex than its global counterpart due to the fact that information about all speakers is not available at the beginning of the processing. A global cross-show speaker diarization could be applied in the context of lifelong speaker diarization but it would require to potentially modify the entire archived corpus every time a new show is processed. For this reason, we only focus in this work on incremental cross-show diarization.

VI.1.2 Baseline system description

The organization of the baseline incremental cross-show diarization system is depicted in Figure VI.1. When processing a new show F_n, within-show speaker diarization is applied. For each speaker, S_i, detected in the current show, all segments assigned to S_i in this show are processed to extract a collection of x-vectors that are then averaged to obtain one single x-vectors F_nS_i representing this speaker. At time T, a collection of N shows $\{F_1, F_2, ...F_N\}$ has been processed and a set of M *known*-speakers has been detected. Each of those M speakers might have appeared in one or several shows from the collection. A database of x-vectors is built by including one single x-vector per speaker and per show, resulting in a collection of x-vectors: $\{F_1S_1, F_1S_2, F_1S_3, F_2S_4, F_2S5, ...F_NS_M\}$. Note that a single speaker S_k might have appeared in several shows F_i, F_j and thus be represented in the x-vector database by several x-vectors $\{F_iS_k, F_jS_k\}$.

When processing a new show F_{N+1}, a new speaker S_α is detected and its single x-vectors representation, $F_{N+1}S_\alpha$, is extracted as explained above. As a result, the current file produces a set $\{F_{N+1}S_\alpha, ..., F_{N+1}S_\lambda\}$ of x-vectors. In a second step, all x-vectors from a given speakers appearing in previous shows are averaged to obtain a single x-vector per speaker. It is important to notice that this representation of a *known*-speaker is an average of x-vectors extracted from multiple shows with different acoustic conditions.

A pseudo-distance matrix is computed by comparing speakers from the current show to *known*-speakers using the same PLDA model as for within-show diarization or cosine

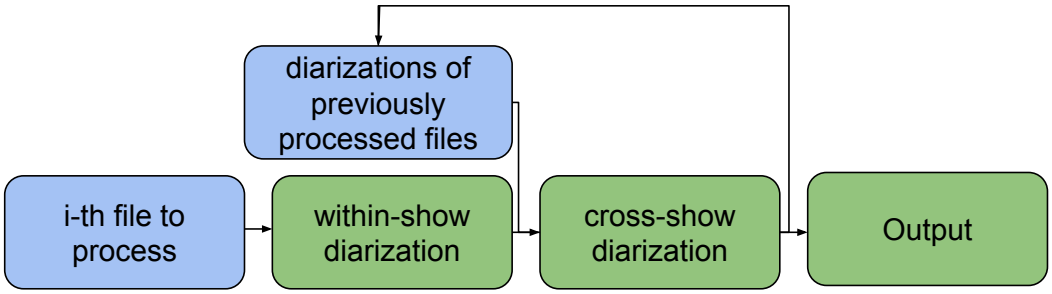

Figure VI.1 – The organization of the baseline incremental cross-show diarization system

similarity measure, depending on the baseline system. To compute this matrix, *known*-speakers are concatenated to new ones in a single list. A verification score is computed for each pair of speakers as described in Figure VI.2. The verification score is then multiplied by -1 and shifted so that the minimum of pseudo-distances is 0. The diagonal values of this matrix is set to inf.

Before using this pseudo-distance matrix to merge speakers, we constrain the clustering by forbidding merging of two new speakers (we do not question the result of the within-show diarization) and merging of two *known*-speakers (we do not question the past cross-show diarization). To apply this constraint, both upper left and lower right block of the pseudo-distance matrix are set to ∞ (see Figure VI.2).

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 all other pseudo-distances involving those speakers are set to inf to prevent merging those two speakers 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 $F_{N+1}S_\beta$ merged with a *known*-speaker F_kS_i is renamed accordingly ($F_{N+1}S_i$) and their x-vector from the current file is added to the database.

VI.1.3 Performance of the baseline approach

Cross-show speaker-linking is applied on top of within-show speaker diarization and its performance is strongly dependent on this first step. In chapter V we've presented and analysed the performance of our proposed human-assisted speaker diarization method

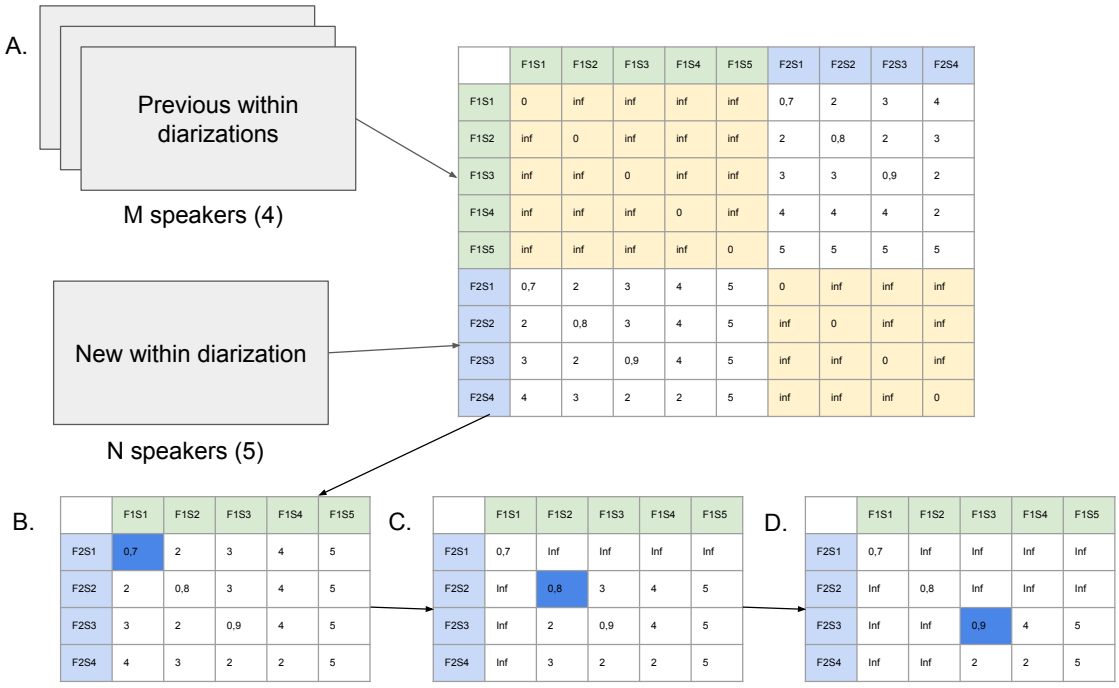

Figure VI.2 – Scheme of one step of the baseline incremental cross show diarization system. A: preparation of the pseudo-distance matrix. B: detection of the nearest already seen speaker and speaker from the new recording. C: modification of matrix to exclude change of withing-show diarization and selection of the next pair of nearest speakers. D: Stopping the process when reaching threshold or processing of all matrix.

when considering two systems: a "Sincnet-diarization system" and a "Resnet-diarization system". The within-show performance of those systems is given in Tables VI.1 as a reminder.

For each system and task reported in Table VI.1, the proposed baseline approach described in Section VI.1.1 is applied and results are given in Table VI.2. From this Table, it is noticeable that the ResNet system strongly over-performs the SincNet system thanks to the quality of its speaker representations (x-vectors). This is due to the large quantity of speakers and sessions the ResNet is trained on and to the more performant architecture of the neural network.

Focusing on the ResNet system, we observe that on the development set, replacing the ground-truth segmentation by an automatic VAD is beneficial; the DER is 32.02% with the reference segmentation and 27.31% with the automatic one. This conclusion is counter intuitive and does not hold on the evaluation set. We have no solid explanation for this phenomenon and the analysis would require experiments that the context of this work

Table VI.1 – 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 weighed according to the duration of the shows.

System	Segmentation	Data	Threshold	DER
SincNet	Ref	Dev	47	17.77
SincNet	Ref	Eval	47	13.38
SincNet	VAD	Dev	45	19.07
SincNet	VAD	Eval	45	20.20
ResNet	Ref	Dev	0.23	14.12
ResNet	Ref	Eval	0.23	10.63
ResNet	VAD	Dev	0.21	14.97
ResNet	VAD	Eval	0.21	16.74

did not allow. However, we make the assumption that our automatic segmentation system over-segmentates the audio stream and leads the within-show speaker diarization system to detect too many speakers per file. When using automatic segmentation, the average duration of speech of one speaker is lower than when using the reference segmentation which makes the cost of an error during the cross-show speaker-linking lower.

Table VI.2 – Performance of the baseline cross-show diarization system with 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 weighed according to the duration of the shows.

System	Segmentation	Data	Threshold for identification	DER
Sincnet	Ref	Dev	-56	53,38
Sincnet	Ref	Eval	-56	51,33
Sincnet	VAD	Dev	-59	53,75
Sincnet	VAD	Eval	-59	53,85
Resnet	Ref	Dev	0.25	32.02
Resnet	Ref	Eval	0.25	30.43
Resnet	VAD	Dev	0.3	27.31
Resnet	VAD	Eval	0.3	31.13

VI.2 The proposed human-assisted cross-show active-correction process

We've previously defined the task of incremental cross-show speaker linking as a clustering task close to open-set speaker identification. Performance of the baseline system presented in Section VI.1.1 shows that there is a large room for improvement. In the following sections, we propose to include a human in the loop during the incremental cross-show speaker linking process in order to improve the quality of the final diarization hypothesis by making use of an active correction process.

Similarly to the human-assisted within-show speaker diarization, our method only focuses on speaker-linking (i.e., clustering) and does not modify the segmentation nor the clustering obtained during the stage of within-show diarization. Subsequently, we decide to use a similar approach that restrains the interaction between human and system to a simple binary question.

During an incremental cross-show speaker-linking process, similar to the one described in VI.1.1, the automatic system selects a couple of speakers: S_i, who appeared in the past and S_α who appears 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 our constraint does not allow to define the question by following a clustering tree.

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 filer 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.

Those two tasks are further discussed in the following sections but one important question that rises for those tasks is the type of speaker representation to use. Three possible approaches are possible. In the first approach, we propose to average all representations of a same speakers from the different files to store only one representation per speaker. In this case there may be a problem due to the influence of the acoustic condition information in the representations. The representations of a same speaker from different files can be significantly different, and averaging of such distant vectors can lead to a noisy representation and generate errors when further detecting this speaker. To avoid this potential problem we consider a second approach that consists of storing one representation of each speaker per recording. In this case, we do not apply averaging of speaker representations across shows. This approach can lead to a high number of questions due to the higher number of representations stored in the database and thus to compare. A third approach consists of storing one representation per speaker for each segment in each file. This solution will significantly increase the number of representations to compare and will lead to a very high number of questions. As this number of vectors will be too high, we will compare only the first and second approaches in the remaining of this chapter.

VI.2.1 Detection of recurrent speakers

This step aims at detecting, amongst the speakers from the current show, a subset of those speakers who have been observed in previous shows. To detect the recurrent speakers we propose to use a pseudo-distance matrix based on the one described in Section VI.1.1. The information conveyed in this matrix is the pseudo-distance between couples of speakers, the lower the pseudo-distance, the more likely both speakers are the same. More precisely, we only focus on the bottom left part of this matrix: the matrix of pseudo-distances between *known*-speakers observed in the past - one column per speaker and per show - and the speakers from the current show - one row per speaker. This matrix is depicted in the left part of Figure VI.3.

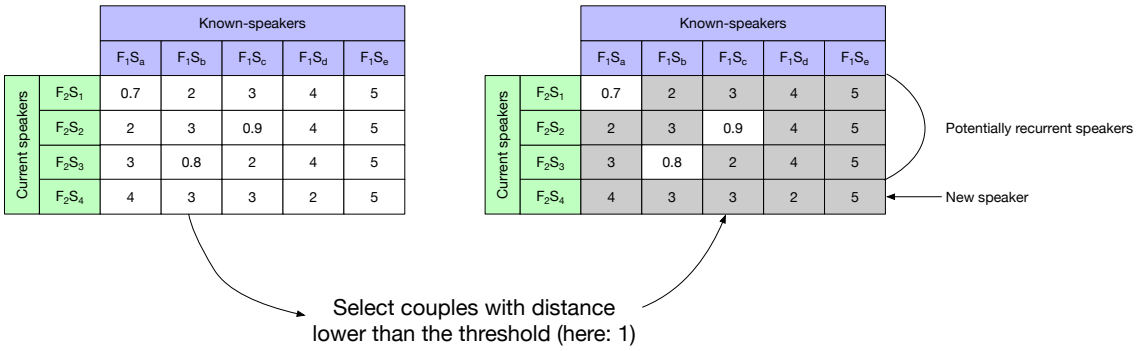

Figure VI.3 – Schema of recurrent speaker detection method based on the analysis of pseudo-distance matrix. If the distance between a new speaker and the nearest known speaker is lower then a threshold, the new speaker is considered potentially recurrent.

A threshold set empirically on a development set is applied on the pseudo-distances. If a current speaker has no distance below the threshold (see Figure VI.3) this speaker is labeled as new (never seen in the past). Other speakers are selected for the second phase: closed-set speaker identification.

The same detection process is depicted on the Figure VI.4. On Figure VI.4.A, speakers from the current show are represented by the red dots while speakers from the already processed files are represented by the green and blue dots. The black circles around the speakers from the current show correspond to threshold. On Figure VI.4.B, each new speaker (red dot) is linked to its nearest speaker from previous shows (by a black line). If the distance between the new speaker and its nearest neighbour from previous shows is lower than the threshold the circle is colored in blue, in case this distance is higher than the threshold, the circle is colored in red, meaning that this new speaker has never been seen before. On Figure VI.4.C, the never seen speaker is excluded from the following processing.

The detection of recurrent speakers can generate two types of errors: a false positive, when a never seen speaker is considered as recurrent and a false negative, when a recurrent speaker is considered as never seen. In our proposed approach, all speakers who are labeled "never seen" will be added to the *known*-speaker database without further consideration while speaker labeled "recurrent" will go through the human-assisted closed-set identification. It is thus important not to miss any recurrent speaker by setting a threshold high enough. On the other side, setting a threshold too high will increase the work load of the human operator.

In the remaining of this section, we aim at fine-tuning the threshold to detect never

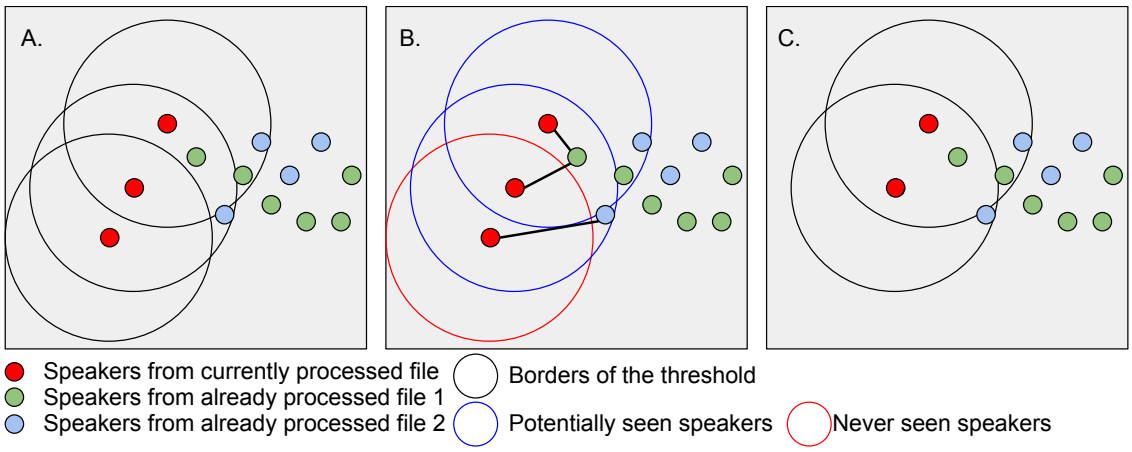

Figure VI.4 – Description of the recurrent speaker detection method based on the analysis of pseudo-distance matrix. A: example of the location of speakers from the currently processed file (red dots) and speakers from the already processed files (green and blue dots). Black circles around speakers from the currently processed file corresponds to the borders of the threshold. B: distances from the currently processed file to the nearest already seen speakers and comparison of this distances with threshold. If the distance is lower than the threshold its borders is colored in blue, in the other case in red. C: the never seen speaker is excluded from the following process.

seen speakers. For each configuration of our within-show baseline system, the algorithm described in section VI.2.1 is applied. Multiple thresholds were tested to select the optimal one. Figures VI.5, VI.6, VI.7 and VI.8 display the percentage of speakers that are misclassified for different thresholds. The color of the bars indicates the part of False Negative in red (known speakers that are classified as never seen) and False positive in blue (never seen speakers that are classified as known).

Once more, we observe that the error rates are much lower for the Resnet system that for the Sincnet systems. While the lowest error rate for Resnet is equal to 17,56% when using the reference segmentation and 21,75% when using a VAD based segmentation, the lowest error rate for Sincnet is equal to 32,84% for reference segmentation and 34,09% for VAD based segmentation.

When targeting the minimum cumulative error rate (the lowest bar on each graph), we can see that the largest part of errors for this threshold is due to the false positives. In other words, the main part of errors will lead the system to ask un-necessary questions.

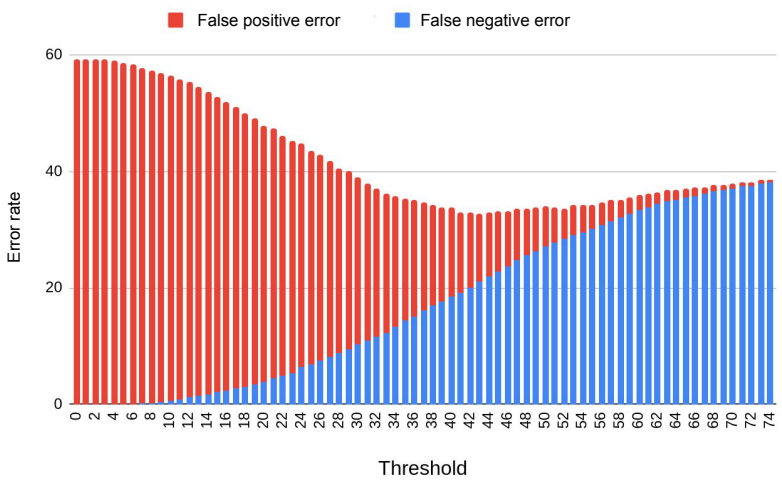

Figure VI.5 – Error rates of recurrent speaker detection on the DEV part of the ALLIES dataset with SINCNET within-show baseline and reference segmentation. For each value of the threshold, the bar plot represents the percentage of false positive and false negative errors in red and blue respectively.

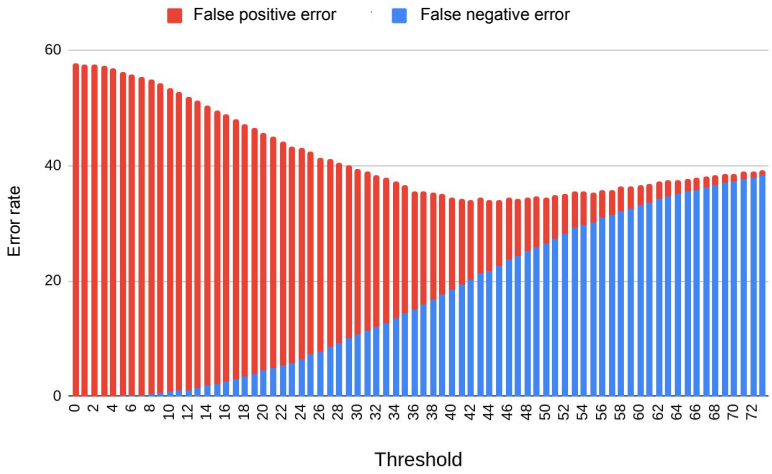

Figure VI.6 – Error rates of recurrent speaker detection on the DEV part of the ALLIES dataset with SINCNET within-show baseline and VAD segmentation. For each value of the threshold, the bar plot represents the percentage of false positive and false negative errors in red and blue respectively.

VI.2.2 Identification of recurrent speakers

In a second step, human-assisted closed-set identification is applied for all speakers labeled as possibly recurrent during the detection of recurrent speakers. For each possibly

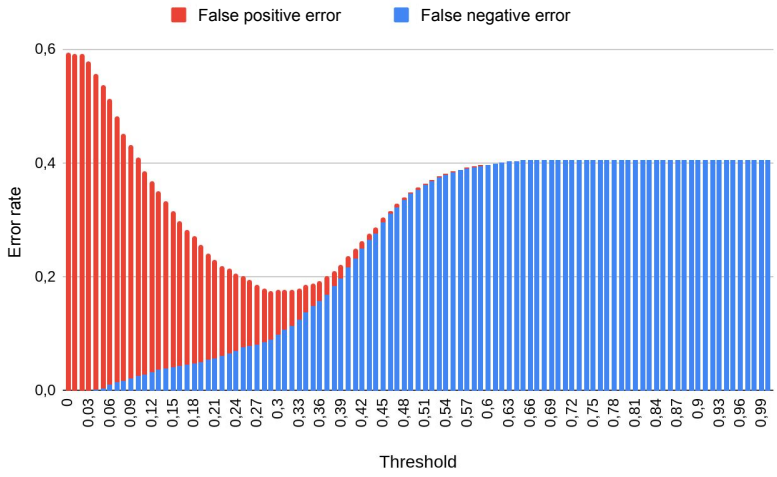

Figure VI.7 – Error rates of recurrent speaker detection on the DEV part of the ALLIES dataset with RESNET within-show baseline and reference segmentation. For each value of the threshold, the bar plot represents the percentage of false positive and false negative errors in red and blue respectively.

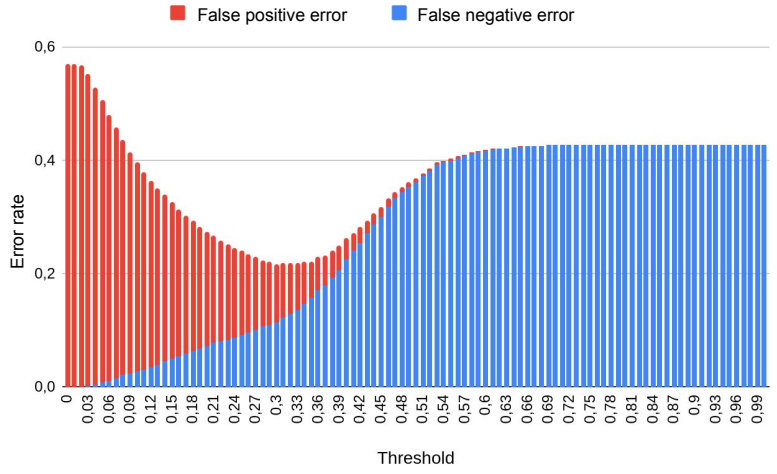

Figure VI.8 – Error rates of recurrent speaker detection on the DEV part if the ALLIES dataset with RESNET within-show baseline and VAD segmentation. For each value of the threshold, the bar plot represents the percentage of false positive and false negative errors in red and blue respectively.

recurrent speaker, x-vectors from all *known*-speakers are sorted by increasing pseudo-distance

Note that the x-vectors of both the possibly recurrent speaker and all *known*-speakers

that are compared to this speaker might have been computed using a cluster of speech segments. Thus asking a human operator to compare a couple of speakers requires, just as for the case of within-show human assisted diarization, to select two speech segments to listen to (one belongs to the current speaker and one belongs to a known-speaker). Thus, following the ranking, binary questions are asked to the human operator to compare two speakers: by listening to their speech segments: "Has this segment (belonging to a known-speaker) been generated by the current speaker?" To answer this question and similarly to the within-show human-assisted process, the human operator is offered two audio segments to listen to: the longest for each speaker. 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.

VI.2.2.1 Speaker representations to handle variability

Based on the idea that *known*-speakers own several x-vectors in each show where they appear and that variability within and across shows can lead to identification errors, we perform the following experiment to evaluate the impact of within- and cross-show variability. Based on the reference segmentation we process all shows in chronological order. For each recurrent speaker we select the closest x-vector from past shows and Figure VI.9 shows the percentage of those vectors that really belong to the recurrent speaker.

Without surprise, the percentage of speakers whose nearest x-vector does indeed belong to the same speaker is higher for the Resnet system than for the Sincnet system. In all cases this metric does not reach one hundred percent which means that the true occurrence of a recurrent speaker is not always the closest x-vector. As a conclusion, for cross-show speaker identification, it is necessary to not only consider the nearest x-vector.

To go further, using automatic within show diarizations, for each speaker detected as recurrent in the current show we represent this speaker by a single x-vector (the average of all x-vectors computed per segment in this show for this speaker). This x-vector is then compared to all speakers from past shows. Past show speakers are represented by one single x-vector (average of all x-vectors extracted for all segment of this speaker in all previous shows. Speakers from the past are then ranked by increasing pseudo-distance.

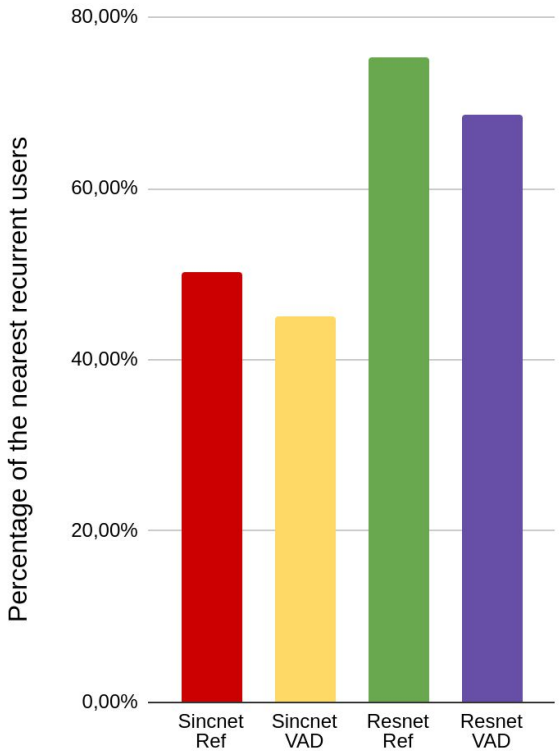

Figure VI.9 – Ratio of the nearest known speakers which is the same as current speaker to all recurrent speakers. Results presented for the systems with Sincnet and Resnet baselines and different segmentation (reference and VAD).

In the ideal case, the correct speaker from the past should be ranked first. We observe on Figure VI.9 that it is not always the case. On Figure VI.10, we show the position the correct speaker obtains in this ranking in percentage. T

Figure VI.10 shows, for each rank, the percentage of speakers that are the same as the current one. Ranks from 2 to 10 are depicted separately and further ranks are grouped as the ratio of speakers is lower. The sum of the all values from this figure and the figure VI.9 represent the distribution of all recurrent speakers for each system. The percentage of recurrent speakers decreases from the earliest positions to the latest positions showing that the majority of recurrent speakers can be found in the earliest positions.

This analysis enables us to determine the number of questions to ask a human operator depending on the expected performance. It was decided to search the optimal limit of question per speaker in the diapason ranging from one to seven. Indeed, the first seven positions contain 89,76% of the recurrent speakers for Resnet system with reference seg-

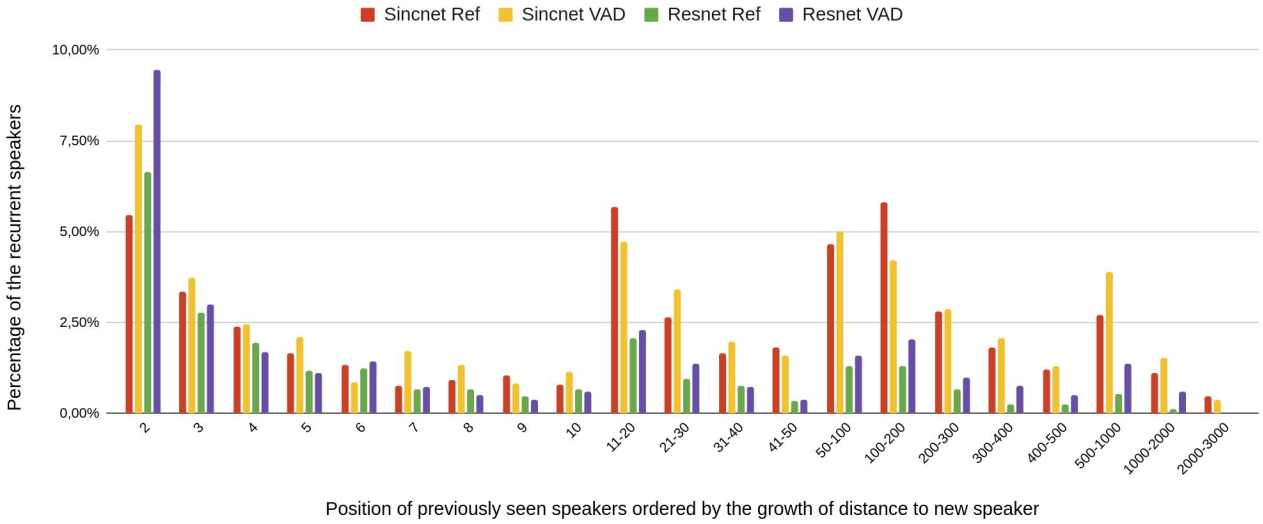

Figure VI.10 – Ratio of the known speakers on different ranks which is the same as current speaker to all recurrent speakers. Results presented for the systems with Sincnet and Resnet baselines and different segmentation (reference and VAD).

mentation, 85,95% of the recurrent speakers for Resnet system with VAD segmentation 65,11% of the recurrent speakers for Sincnet system with reference segmentation and 63,88% of the recurrent speakers for Sincnet system with VAD segmentation. The selection of the optimal limit of questions is provided by the experiments on the DEV partition of the dataset and described in the following subsection.

VI.2.3 Human assisted speaker close-set identification

As mentioned above, the smaller the distance between two x-vectors, the higher the probability those two representations correspond to the same speaker. Consequently, for each potentially recurrent speaker it is reasonable to select the nearest speaker. But such solution can provide errors we want to avoid. At this point we propose to use the possibility to interact with the human expert (in an active learning setup).

As it is not reasonable to ask the human operator to compare a new speaker to all already seen speakers from previous shows, the following subsections describes our proposals in order to minimize the number of questions asked.

VI.2.3.0 - A Questioning the whole list

For each potentially recurrent speaker, we use the pseudo distance matrix described in

VI.1.2. The speakers seen in previous recordings are sorted per increasing pseudo-distance to the considered potentially recurrent speaker. Then, the system asks the human expert to compare the current speaker with the ranked speakers from previous shows, until the human expert detects the recurrent speaker or answers the limit of questions for this current speaker. If the system reaches the limit without finding out an occurrence of the current speaker, this one is considered as never seen. Such solution can lead to miss the recurrent speakers in case its x-vector from a previous occurrence is ranked to far. On the other hand it forbids the system to ask a high number of questions, which would be the case if the current potentially recurrent speaker has been wrongly detected as recurrent during the first step of our approach.

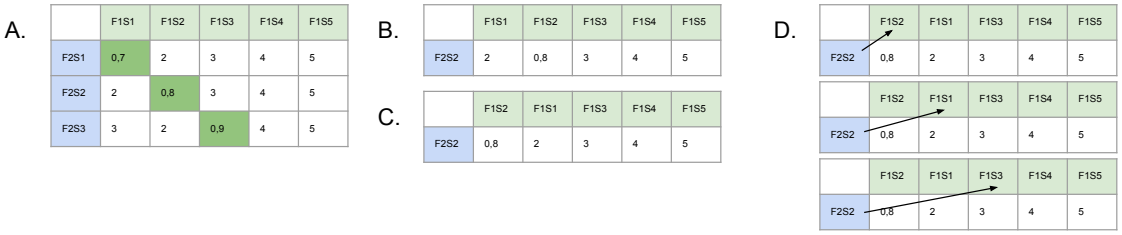

Figure VI.11 – Schema of the All list processing question selection strategy. A: Preparation of the pseudo distances between potentially recurrent current speakers. B: processing of each potentially recurrent speaker one by one. C: for each potentially recurrent speaker sorting the list of known speakers in order of pseudo distance growth. D: asking the questions to the human expert about each speaker in the ordered list until reaching recurrence confirmation or reaching limit of questions.

VI.2.3.0 - B Questioning the closest speaker per file

The second proposed approach is based on the following hypothesis: Speaker representations are extracted from audio recordings under various acoustic conditions. Despite the recent improvement of speaker recognition systems, they still exhibit weaknesses against acoustic or lexical mismatches. Those can lead to high pseudo-distances between two x-vectors extracted for a same unique speaker. On the contrary, different speakers recorded under similar acoustic conditions or pronouncing similar lexical contents might obtain very low pseudo-distances.

In this section, we assume that a speaker might obtain low pseudo-distances when recorded under different acoustic conditions, i.e., during different shows, but that other speakers recorded under the same acoustic condition, i.e., in the same show, will obtain even lower pseudo-distances. In other words, when comparing a current speaker with

speakers from previous shows, for each past show, if the current speaker appeared in it, then it should be the closest from all speakers from this show.

Instead of asking questions about all speakers in the ordered list we propose to ask question only about one speaker per previously processed show. In other words, when we obtain the negative response of the human (compared speakers are different) we delete from the list all speakers who appeared in the same show. This modification aims at reducing the number of questions asked to the human operator. The analysis of the proposed hypothesis and results of the application of this strategy described in the following section.

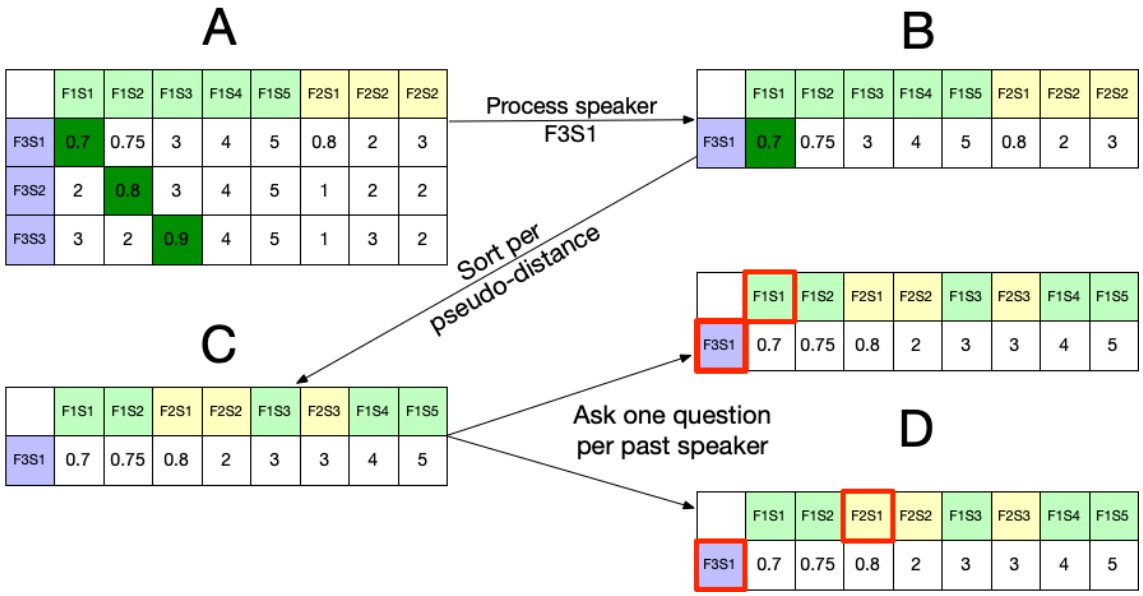

Figure VI.12 – Schema of the Nearest speaker per file processing question selection strategy. A: Preparation of the pseudo distances between potentially recurrent current speakers. B: processing of each potentially recurrent speaker one by one (here only the example of speaker F3S1). C: for each potentially recurrent speaker sorting the list of known speakers in order of pseudo distance growth. D: asking the questions to the human expert about only nearest speaker in each file until reaching recurrence confirmation or reaching limit of questions.

VI.3 Results and analysis

This section presents the performance of our different human assisted cross-show diarization strategies applied with the Sincnet and Resnet systems using both refence and

VAD segmentations. A first series of experiments has been performed on the Dev partition of the ALLIES corpus and in order to select the optimal strategies and meta-parameters of the system while a second series of experiments performed on the Test partition of the ALLIES corpus provides the final performance of the systems.

VI.3.1 Systems performance analysis

Figures VI.13, VI.14, VI.15, VI.16 present the performance of human assisted cross-show diarization strategies for the two systems: SincNet and ResNet using both the reference segmentation and the automatic VAD respectively.

For each system, the DER obtained with our cross show incremental baseline is shown as a red line for reference. For each configuration (system and segmentation) we apply the different strategies for speaker representation (Averaging and No averaging) and human assisted speaker close-set identification questioning (described in Section VI.2.3).

The results of the different strategies are illustrated with different colors. For each strategy, we vary the maximum number of questions asked to the human operator from 1 to 7 and present the DER (bright part of the bar) as well as the penalization introduced in SectionIV.3.3 (dark part of the bar). The sum of the two parts corresponds to the DER penalized.

As a first conclusion, we confirm that in all cases, increasing the maximum number of questions per show reduces the final DER while increasing the amount of penalization. This results shows that our system is able to ask useful questions to the user and that increasing the maximum number of question really leads to asking more questions to the user. This results is a confirmation of our analysis provided in Section VI.2.1 that recurrent speaker are not ranked first when performing cross-show speaker-linking.

For all configurations, we also notice that the optimal maximum number of questions per show, in terms of penalized DER, is lower than 7 as we observe a minimum within this range before the penalized DER increases. Based on this observation, for each configuration (system and strategy), we are able to determine the optimal maximum number of questions per show to use on the Test partition of the ALLIES corpus later.

Comparing the representations of speaker we observe that the "No averaging" strategy outperforms the "Averaging" strategy for all systems and strategies. This comes against the standard approach used for speaker verification that consists of averaging x-vectors obtained on short segments in order to increase the robustness of the representation and is probably due to the fact that some pairs of short segments might exhibit a high

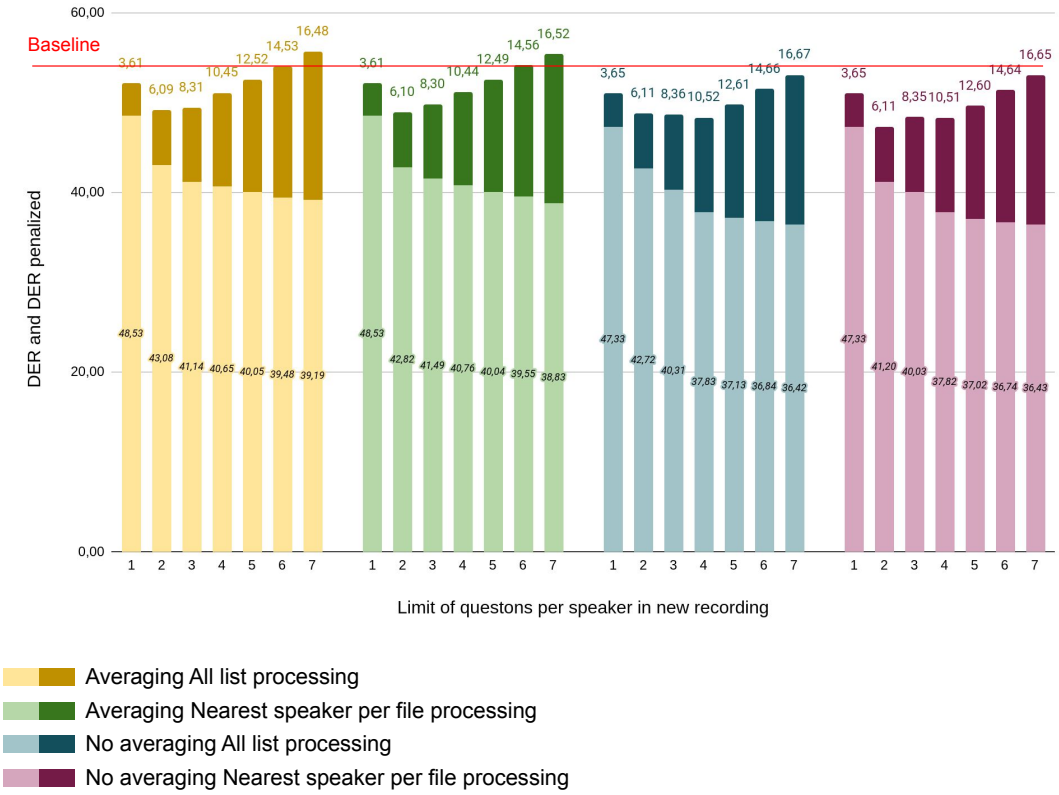

Figure VI.13 – Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Sincnet using the reference segmentation. Each group of columns represents one strategy combining a type of speaker representation (with averaging and without) and a method of question selection (All list and Nearest speaker per file). For each group, the maximum number of questions asked to the user varies from 1 to 7 per show. The light part of the bar corresponds to the final DER while the dark part of the bar corresponds to the penalization. The sum of those two (complete bar) corresponds to the penalized DER.

similarity due to similar lexical contents or acoustic conditions. A deeper analysis of the pairs provided to the human operator might help improving the human assisted process further.

The difference between results obtained for the All list processing and Nearest speaker per file processing strategies is not significant enough to conclude on the superiority of one of those strategies. Nearest speaker per file processing strategy shows best results for the SincNet baseline using an automatic Vad segmentation while in other cases all list processing strategies outperforms it but with minor differences in any case.

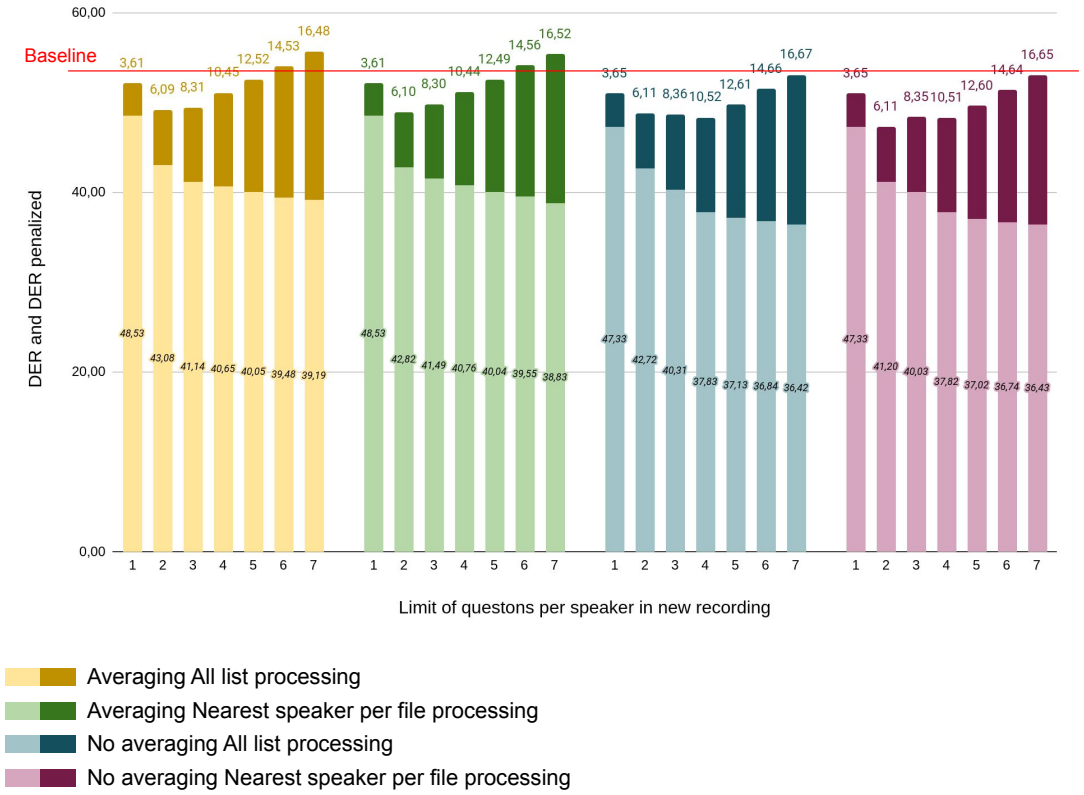

Figure VI.14 – Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Sincnet using an automatic VAD segmentation. Each group of columns represents one strategy combining a type of speaker representation (with averaging and without) and a method of question selection (All list and Nearest speaker per file). For each group, the maximum number of questions asked to the user varies from 1 to 7 per show. The light part of the bar corresponds to the final DER while the dark part of the bar corresponds to the penalization. The sum of those two (complete bar) corresponds to the penalized DER.

Based on those results, we set the optimal strategies and parameters for each case and perform the same experiment on the Eval partition of the ALLIES dataset. Results of this experiments are given in the table VI.3.

We observe in Table VI.3 that in all cases both DER and penalized DER metrics reduce when involving the human in the loop. The largest improvement is achieved for the ResNet system using the reference segmentation. This shows the important role of the segmentation and further work is necessary to enable human assisted correction of the segmentation that is a very tedious and expensibe task.

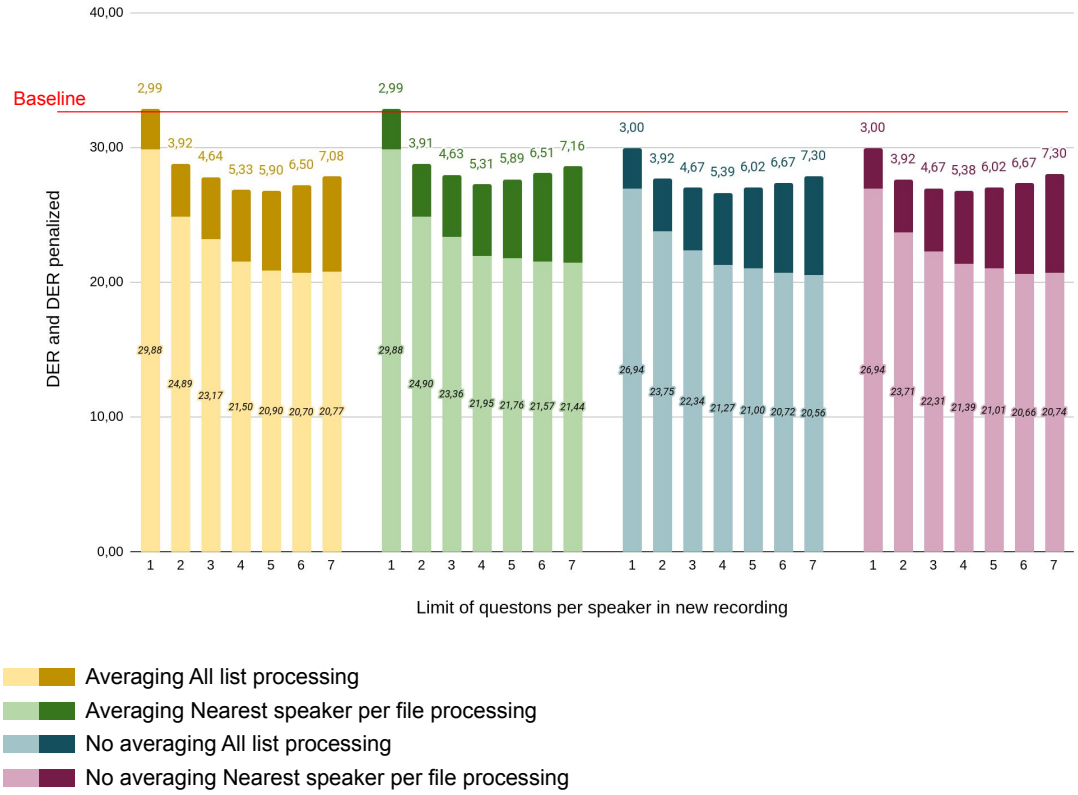

Figure VI.15 – Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Resnet using the reference segmentation. Each group of columns represents one strategy combining a type of speaker representation (with averaging and without) and a method of question selection (All list and Nearest speaker per file). For each group, the maximum number of questions asked to the user varies from 1 to 7 per show. The light part of the bar corresponds to the final DER while the dark part of the bar corresponds to the penalization. The sum of those two (complete bar) corresponds to the penalized DER.

VI.3.2 Conclusions

In this chapter, we've proposed a cross-show active correction system. We have proposed and tested various strategies and identified the most effective ones. The proposed solutions demonstrate significant improvement in terms of DER compared to our baseline system when applied on top of both Sincnet and Resnet systems.

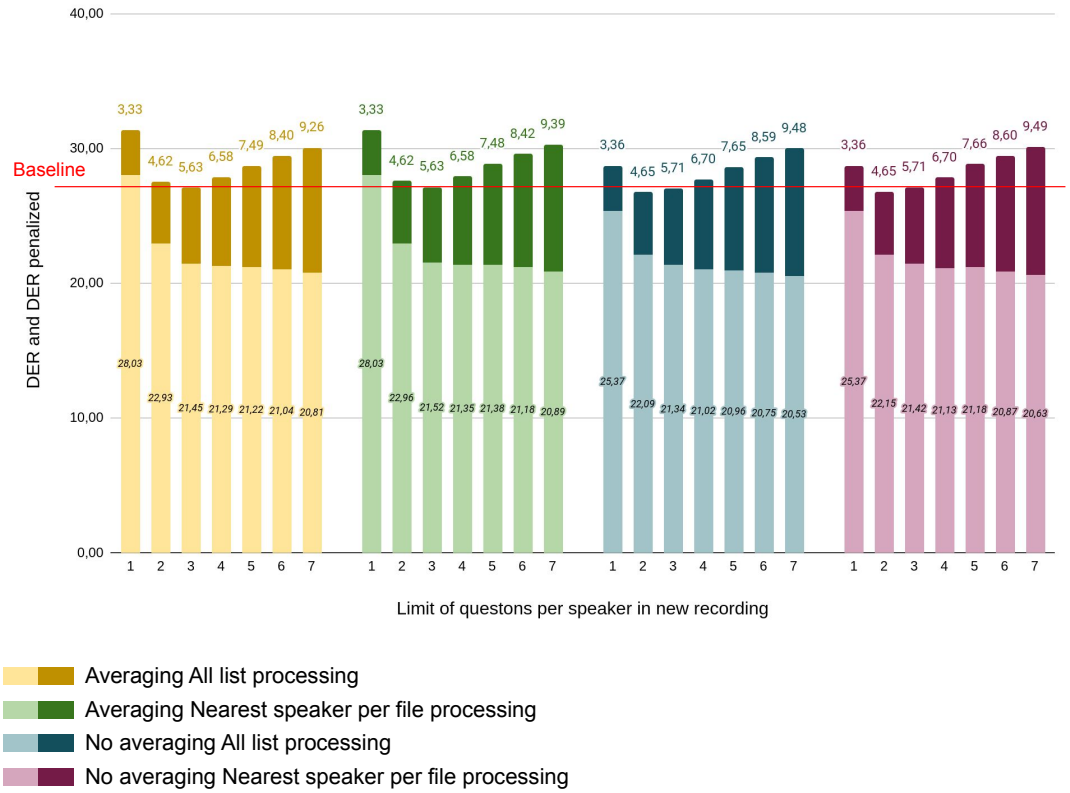

Figure VI.16 – Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Resnet using an automatic VAD segmentation. Each group of columns represents one strategy combining a type of speaker representation (with averaging and without) and a method of question selection (All list and Nearest speaker per file). For each group, the maximum number of questions asked to the user varies from 1 to 7 per show. The light part of the bar corresponds to the final DER while the dark part of the bar corresponds to the penalization. The sum of those two (complete bar) corresponds to the penalized DER.

Table VI.3 – Performance of the human assisted cross-show diarization systems on the Eval partition of the ALLIES corpus.

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	3	2	4	2
Baseline DER	51.33	53,85	30.43	31.13
DER	33.78	37.74	20.30	23,88
Number of questions	6060	5024	3886	3161
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%

Conclusion

This research took part in the wider context of the European Chist-ERA project ALLIES that aims at designing an autonomous speaker diarization system able to adapt and evaluate itself. Our goal in the project was to apply the concept of human assisted lifelong learning to the speaker diarization task, also known as speaker segmentation and clustering. More specifically, our work was to design an efficient way of interaction between the automatic diarization system and a human domain expert in order to improve the quality of diarization generated by an automatic system while limiting the work load for the human domain expert.

Contributions

A definition of human assisted lifelong learning speaker diarization

In this manuscript, we propose our point of view on the definition of the lifelong learning intelligent systems. Our view is focused on optimizing the model to the future incoming data and on minimizing the forgetting effect, when the new versions of the model performs worse than previous versions on previous data. Also, we propose a nomenclature of different types of interactions between the intelligent system and the human expert.

Material to promote the development of the task

We have developed a dataset designed for evaluation of the lifelong learning diarization systems. The proposed dataset has a number of properties such as timestamps and high number of annotated recurrent speakers which allows to process it in the chronological order and learn new information from changes of voices of recurrent speakers. These properties makes the proposed dataset unique. It is the only public dataset which can be used to evaluate lifelong learning diarization task, and it will be used in the ALLIES evaluation campaign.

Another impact of our work is the metric for evaluation of human-assisted systems. A proposed metric was developed for the general case, in other words for estimating the performance on different tasks. It was applied not only for diarization task but also on the machine translation task (BLEU metric) in the context of the ALLIES project. The penalisation term estimates the amount of information provided by the human expert as a portion of the corresponding metric and penalizes the final score to highlight the generalization effect of the human-assisted system. We also presented some protocols

according to which it is possible to perform the evaluation of different human-assisted lifelong learning systems.

Human assisted approaches for within-show and cross-show speaker diarization

The main contribution of our work lies in the development of the human-assisted within-show and cross-show diarization methods. For within-show the human-assisted system is based on the analysis of the dendrogram obtained during the final step of the hierarchical agglomerative clustering, and then asking questions to human domain experts in order to improve the clustering. For this task we proposed various strategies to select the question to ask and to select the segments that should be compared by the domain expert. Also, we tested different stopping criteria to decide when it is not reasonable to continue asking questions. For the within-show diarization, we achieved a DER reduction of up to 18,83% and a penalized DER reduction of up to 9,94% relatively to strong baseline systems. The results on penalised DER can also be interpreted as correcting almost 10% of the errors only by generalising from the information obtained from the human domain expert.

For cross-show diarization, the human-assisted system is based on the the analysis of the pseudo-distance matrix based on speaker representations. Then obtain information helps to ask minimal number of questions to human domain experts which allows to detect the pairs of recurrent speakers. For this task we've tested different strategies of speaker representation and selection of speakers which should be compared to solve the problem of cross-show variability. Also we tested different limits of questions. For the cross-show diarization, we achieved an even higher reduction: up to 34.19% relative for DER and up to 14.31% relative for penalized DER. The results on penalised DER show that we achieved to correct 14.31% of errors by generalisation of information obtained from the human domain expert. For both tasks, tests were applied on different baseline systems to have more details on the performance of the proposed strategies.

Perspectives

These results open a way for further research. One of the perspectives is to combine the within-show and cross-show strategies. It is possible to use them sequentially, but a more interesting case is to use them simultaneously and avoid the possibly not necessary

questions. Such effect can be possibly reached by comparing segments of the current show between themselves and with the speaker representations from the previous shows at the same time. In other words, merge the solving of two tasks: within-show and cross-show diarization in one method.

On import step remaining to solve is the development of a life-long adaptation method for speaker diarization. We've made attempts to create such solution (not reported in this manuscript), but have been blocked by the low performance of the cross-show diarization due to the high cross-show variability. We focused on the solution to this problem using human-assisted cross-show diarization. The results obtained may allow to create the complete lifelong learning pipeline and use the information gathered from the human expert, not only to improve the current results, but also to adapt the system to perform better in general. Also it is interesting to adapt the proposed methods for end-to-end neural approaches, as it may open the way to simpler and more efficient system adaptation process to new data.

Another perspective is to use the video stream for diarization of TV shows by integrating the video system within the correction process. Such approach will provide significantly more information to the system and ease the task for human domain experts.

Another interesting area of research is the study of the interaction process from the point of view of optimizing the work of human experts, the work with ergonomists to develop the ergonomic and efficient user interfaces.

List of Figures

I.1 Standard Mel Frequency Cepstral Coefficient extraction toolchain. 10
I.2 Architecture example of an x-vector extractor neural network 16
I.3 Example of Bi-LSTM neural network architecture used for speaker diarization 17
I.4 Example of dendrogram obtained during a hierarchical agglomerative clustering 21

IV.1 Appearance of all recurrent speakers in the ALLIES corpus according to the recording time 47
IV.2 Cumulative duration of annotated signal across time 49
IV.3 Number of speakers in different partitions of ALLIES corpus and the number of common speakers. 50
IV.4 Visualisation of penalised score calculation 55
IV.5 Illustration of the effect of the penalisation method for different cases . . . 56

V.1 Life-cycle of a human assisted speaker diarization system. 61
V.2 Pipeline of the baseline diarization system. 63
V.3 Illustration of the dendrogram obtained on the final step of the baseline diarization system. 68
V.4 Dendrogram obtained on the final step of the baseline diarization system. . 69
V.5 Dendrogram obtained on the final step of human assisted correction system using two confirmation criterion. 70
V.6 Dendrogram obtained on the final step of human assisted correction system using two All branches criterion. 71

VI.1 The organization of the baseline incremental cross-show diarization system 81
VI.2 Scheme of one step of the baseline incremental cross show diarization system 82
VI.3 Schema of recurrent speaker detection method based on the analysis of pseudo-distance matrix 86
VI.4 Description of the recurrent speaker detection method based on the analysis of pseudo-distance matrix 87

VI.5 Error rates of recurrent speaker detection on the DEV part of the ALLIES dataset with SINCNET within-show baseline and reference segmentation . 88
VI.6 Error rates of recurrent speaker detection on the DEV part of the ALLIES dataset with SINCNET within-show baseline and VAD segmentation . . . 88
VI.7 Error rates of recurrent speaker detection on the DEV part of the ALLIES dataset with RESNET within-show baseline and reference segmentation . . 89
VI.8 Error rates of recurrent speaker detection on the DEV part if the ALLIES dataset with RESNET within-show baseline and VAD segmentation 89
VI.9 Ratio of the nearest known speakers which is the same as current speaker to all recurrent speakers 91
VI.10 Ratio of the known speakers on different ranks which is the same as current speaker to all recurrent speakers 92
VI.11 Schema of the All list processing question selection strategy 93
VI.12 Schema of the Nearest speaker per file processing question selection strategy 94
VI.13 Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Sincnet using the reference segmentation 96
VI.14 Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Sincnet using an automatic VAD segmentation 97
VI.15 Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Resnet using the reference segmentation 98
VI.16 Performance of different human assisted cross-show diarization strategies on the Dev partition of the ALLIES dataset with Resnet using an automatic VAD segmentation 99
List of figures

List of Tables

III.1 Existing diarization corpora 40

IV.1 Comparing ALLIES corpus with previously existing corpora. 46
IV.2 ALLIES speakers across time 47
IV.3 Global partitioning of the ALLIES corpus recorded from 4 channels and 19 shows series with their corresponding duration (and number of shows). All timestamps are given in hh:mm:ss format. 48

V.1 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 64
V.2 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. 65
V.3 Performance of the baseline diarization systems in terms of DER. The threshold of the second HAC clustering, fine-tuned on the development partition, is given in column 3 as it fine-tuned on the DEV set and used on the EVAL set. 66
V.4 Performance of the active correction strategies on *SincNet* baseline with reference segmentation on the 'Development' partition 73
V.5 Performance of the active correction strategies on *SincNet* baseline with reference segmentation on the 'Evaluation' partition 74
V.6 Performance of the active correction strategies on *SincNet* baseline with segmentation based on VAD 75
V.7 Performance of the active correction strategies with the ResNet baseline and REF segmentation 76
V.8 Performance of the active correction strategies with the ResNet baseline and VAD segmentation 77

VI.1 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 weighed according to the duration of the shows. 83
VI.2 Performance of the baseline cross-show diarization system with 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 weighed according to the duration of the shows. 84
VI.3 Performance of the human assisted cross-show diarization systems on the Eval partition of the ALLIES corpus. 100

BIBLIOGRAPHY

[1] M. El Ayadi, M. S. Kamel, and F. Karray, « Survey on speech emotion recognition: Features, classification schemes, and databases », *Pattern Recognition*, vol. 44, *3*, pp. 572–587, 2011.

[2] Y. Gong, « Speech recognition in noisy environments: A survey », *Speech communication*, vol. 16, *3*, pp. 261–291, 1995.

[3] A. Nica, A. Caruntu, G. Toderean, and O. Buza, « Analysis and synthesis of vowels using Matlab », *in 2006 IEEE International Conference on Automation, Quality and Testing, Robotics*, IEEE, vol. 2, 2006, pp. 371–374.

[4] L. Xie and Z.-Q. Liu, « A comparative study of audio features for audio-to-visual conversion in mpeg-4 compliant facial animation », *in 2006 International Conference on Machine Learning and Cybernetics*, IEEE, 2006, pp. 4359–4364.

[5] H. Hermansky, « Perceptual linear predictive (PLP) analysis of speech », *the Journal of the Acoustical Society of America*, vol. 87, *4*, pp. 1738–1752, 1990.

[6] M. Todisco, H. Delgado, and N. W. Evans, « A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients. », *in Odyssey*, vol. 2016, 2016, pp. 283–290.

[7] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacrétaz, and D. A. Reynolds, « A tutorial on text-independent speaker verification », *EURASIP Journal on Advances in Signal Processing*, vol. 2004, *4*, p. 101 962, 2004.

[8] L. Rabiner and R. Schafer, *Theory and applications of digital speech processing*. Prentice Hall Press, 2010.

[9] J. Hernando and C. Nadeu, « CDHMM speaker recognition by means of frequency filtering of filter-bank energies », *in Fifth European Conference on Speech Communication and Technology*, 1997.

[10] H. Hermansky and S. Sharma, « Temporal patterns (TRAPS) in ASR of noisy speech », *in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, IEEE, vol. 1, 1999, pp. 289–292.

[11] A. Lozano-Diez, A. Silnova, P. Matejka, O. Glembek, O. Plchot, J. Pesan, L. Burget, and J. Gonzalez-Rodriguez, « Analysis and Optimization of Bottleneck Features for Speaker Recognition. », *in Odyssey*, vol. 2016, 2016, pp. 352–357.

[12] M. McLaren, L. Ferrer, and A. Lawson, « Exploring the role of phonetic bottleneck features for speaker and language recognition », *in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 5575–5579.

[13] P. Matejka, L. Zhang, T. Ng, O. Glembek, J. Z. Ma, B. Zhang, and S. H. Mallidi, « Neural Network Bottleneck Features for Language Identification. », *in Odyssey*, 2014.

[14] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, « Learning problem-agnostic speech representations from multiple self-supervised tasks », *arXiv preprint arXiv:1904.03416*, 2019.

[15] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, *et al.*, « Wavlm: Large-scale self-supervised pre-training for full stack speech processing », *arXiv preprint arXiv:2110.13900*, 2021.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, « wav2vec 2.0: A framework for self-supervised learning of speech representations », *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[17] J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, Y. Wu, S. Liu, T. Ko, Q. Li, Y. Zhang, *et al.*, « SpeechT5: Unified-Modal Encoder-Decoder Pre-Training for Spoken Language Processing », *in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5723–5738.

[18] H. Muckenhirn, M. M. Doss, and S. Marcell, « Towards directly modeling raw speech signal for speaker verification using CNNs », *in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 4884–4888.

[19] M. Ravanelli and Y. Bengio, « Speaker recognition from raw waveform with sincnet », *in 2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 1021–1028.

[20] H. Gish, M.-H. Siu, and R. Rohlicek, « Segregation of speakers for speech recognition and speaker identification », *in Proc. ICASSP*, vol. 2, 1991, pp. 873–876.

[21] G. Schwarz *et al.*, « Estimating the dimension of a model », *The annals of statistics*, vol. 6, *2*, pp. 461–464, 1978.

[22] P. Delacourt, D. Kryze, and C. J. Wellekens, « Detection of speaker changes in an audio document », *in Sixth European Conference on Speech Communication and Technology*, 1999.

[23] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, « Automatic segmentation, classification and clustering of broadcast news audio », *in Proc. DARPA speech recognition workshop*, vol. 1997, 1997.

[24] X. Zhu, C. Barras, L. Lamel, and J.-L. Gauvain, « Speaker diarization: From broadcast news to lectures », *in International Workshop on Machine Learning for Multimodal Interaction*, Springer, 2006, pp. 396–406.

[25] S. J. Prince and J. H. Elder, « Probabilistic linear discriminant analysis for inferences about identity », *in 2007 IEEE 11th International Conference on Computer Vision*, IEEE, 2007, pp. 1–8.

[26] D. A. Reynolds, « A Gaussian mixture modeling approach to text-independent speaker identification. », 1993.

[27] R. C. Rose and D. A. Reynolds, « Text independent speaker identification using automatic acoustic segmentation », *in International Conference on Acoustics, Speech, and Signal Processing*, IEEE, 1990, pp. 293–296.

[28] S. J. Prince, *Computer vision: models, learning, and inference*. Cambridge University Press, 2012.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, « Maximum likelihood from incomplete data via the EM algorithm », *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, *1*, pp. 1–22, 1977.

[30] D. Matrouf, N. Scheffer, B. G. Fauve, and J.-F. Bonastre, « A straightforward and efficient implementation of the factor analysis model for speaker verification. », *in Interspeech*, 2007, pp. 1242–1245.

[31] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, « Front-end factor analysis for speaker verification », *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, *4*, pp. 788–798, 2010.

[32] S. J. Prince and J. H. Elder, « Probabilistic linear discriminant analysis for inferences about identity », *in 2007 IEEE 11th International Conference on Computer Vision*, IEEE, 2007, pp. 1–8.

[33] P. Kenny, « Bayesian speaker verification with heavy-tailed priors. », *in Odyssey*, vol. 14, 2010.

[34] P. Matějka, O. Glembek, F. Castaldo, M. J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Černocky, « Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification », *in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2011, pp. 4828–4831.

[35] A. Larcher, K. A. Lee, B. Ma, and H. Li, « Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances », *in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 7673–7677.

[36] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, « A novel scheme for speaker recognition using a phonetically-aware deep neural network », *in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 1695–1699.

[37] M. Hussain, M. A. Haque, *et al.*, « Swishnet: a fast convolutional neural network for speech, music and noise classification and segmentation », *arXiv preprint arXiv:1812.00149*, 2018.

[38] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, « Pyannote. audio: neural building blocks for speaker diarization », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7124–7128.

[39] J.-w. Jung, H.-S. Heo, J.-h. Kim, H.-j. Shim, and H.-J. Yu, « RawNet: Advanced End-to-End Deep Neural Network Using Raw Waveforms for Text-Independent Speaker Verification », *Proc. Interspeech 2019*, pp. 1268–1272, 2019.

[40] J.-w. Jung, S.-b. Kim, H.-j. Shim, J.-h. Kim, and H.-J. Yu, « Improved RawNet with Feature Map Scaling for Text-Independent Speaker Verification Using Raw Waveforms », *in Proc. Interspeech 2020*, 2020, pp. 1496–1500. DOI: 10.21437/Interspeech.2020-1011.

[41] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, « Deep neural networks for small footprint text-dependent speaker verification », *in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 4052–4056.

[42] 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)*, IEEE, 2018, pp. 5329–5333.

[43] P. Shen, X. Lu, K. Sugiura, S. Li, and H. Kawai, « Compensation on x-vector for Short Utterance Spoken Language Identification », *in Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 47–52.

[44] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, « x-vectors meet emotions: A study on dependencies between emotion and speaker recognition », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7169–7173.

[45] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, O. Plchot, O. Novotny, H. Zeinali, *et al.*, « BUT System Description for DIHARD Speech Diarization Challenge 2019 », *arXiv preprint arXiv:1910.08847*, 2019.

[46] W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, « Utterance-level aggregation for speaker recognition in the wild », *in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 5791–5795.

[47] T. Zhou, Y. Zhao, and J. Wu, « ResNeXt and Res2Net structures for speaker verification », *in 2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, pp. 301–307.

[48] B. Desplanques, J. Thienpondt, and K. Demuynck, « ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker

Verification », *in Proc. Interspeech 2020*, 2020, pp. 3830–3834. DOI: 10.21437/Interspeech.2020-2650.

[49] Z. Bai and X.-L. Zhang, « Speaker recognition based on deep learning: An overview », *Neural Networks*, vol. 140, pp. 65–99, 2021.

[50] G. Zhu, F. Jiang, and Z. Duan, « Y-Vector: Multiscale Waveform Encoder for Speaker Embedding », *in Proc. Interspeech 2021*, 2021, pp. 96–100. DOI: 10.21437/Interspeech.2021-1707.

[51] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[52] T. Kinnunen and H. Li, « An overview of text-independent speaker recognition: From features to supervectors », *Speech communication*, vol. 52, *1*, pp. 12–40, 2010.

[53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, « The Kaldi speech recognition toolkit », *in IEEE 2011 workshop on automatic speech recognition and understanding*, IEEE Signal Processing Society, 2011.

[54] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1997.

[55] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, « Speaker diarization: A review of recent research », *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, *2*, pp. 356–370, 2012.

[56] P.-A. Broux, F. Desnous, A. Larcher, S. Petitrenaud, J. Carrive, and S. Meignier, « S4D: Speaker Diarization Toolkit in Python », 2018.

[57] E. Rentzeperis, A. Stergiou, C. Boukis, A. Pnevmatikakis, and L. C. Polymenakos, « The 2006 athens information technology speech activity detection and speaker diarization systems », *in International Workshop on Machine Learning for Multimodal Interaction*, Springer, 2006, pp. 385–395.

[58] A. Temko, D. Macho, and C. Nadeu, « Enhanced SVM training for robust speech activity detection », *in 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP07*, IEEE, vol. 4, 2007, pp. IV–1025.

[59] G. Gelly and J.-L. Gauvain, « Minimum word error training of RNN-based voice activity detection », *in Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[60] X. Tan and X.-L. Zhang, « Speech enhancement aided end-to-end multi-task learning for voice activity detection », *in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 6823–6827.

[61] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, « The second DIHARD diarization challenge: Dataset, task, and baselines », *arXiv preprint arXiv:1906.07839*, 2019.

[62] A. K. Sarkar, D. Matrouf, P. M. Bousquet, and J.-F. Bonastre, « Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification », *in Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[63] B. Fauve, N. Evans, N. Pearson, J.-F. Bonastre, and J. Mason, « Influence of task duration in text-independent speaker verification », *in Eighth Annual Conference of the International Speech Communication Association*, 2007.

[64] M. McLaren, R. Vogt, B. Baker, S. Sridharan, and S. Sridharan, « Experiments in SVM-based Speaker Verification Using Short Utterances. », *in Odyssey*, vol. 17, 2010.

[65] A. Kanagasundaram, R. Vogt, D. B. Dean, S. Sridharan, and M. W. Mason, « I-vector based speaker recognition on short utterances », *in Proceedings of the 12th Annual Conference of the International Speech Communication Association*, International Speech Communication Association (ISCA), 2011, pp. 2341–2344.

[66] H. Zeinali, K. Aik Lee, J. Alam, and L. Burget, « Short-duration Speaker Verification (SdSV) Challenge 2020: the Challenge Evaluation Plan », *arXiv*, arXiv–1912, 2019.

[67] V. Gupta, « Speaker change point detection using deep neural nets », *in 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 4420–4424.

[68] R. Yin, H. Bredin, and C. Barras, « Speaker change detection in broadcast TV using bidirectional long short-term memory networks », 2017.

[69] K. Boakye, B. Trueba-Hornero, O. Vinyals, and G. Friedland, « Overlapped speech detection for improved speaker diarization in multiparty meetings », *in 2008 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2008, pp. 4353–4356.

[70] M. Yousefi and J. H. Hansen, « Block-based high performance CNN architectures for frame-level overlapping speech detection », *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 28–40, 2020.

[71] G. Dupuy, M. Rouvier, S. Meignier, and Y. Esteve, « I-vectors and ILP clustering adapted to cross-show speaker diarization », 2012.

[72] S. Chen, P. Gopalakrishnan, *et al.*, « Speaker, environment and channel change detection and clustering via the bayesian information criterion », *in Proc. DARPA broadcast news transcription and understanding workshop*, Virginia, USA, vol. 8, 1998, pp. 127–132.

[73] D. A. Reynolds, E. Singer, B. A. Carlson, G. C. O'Leary, J. J. McLaughlin, and M. A. Zissman, « Blind clustering of speech utterances based on speaker and language characteristics », *in Fifth International Conference on Spoken Language Processing*, 1998.

[74] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, « Exploiting intra-conversation variability for speaker diarization », *in Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[75] S. H. Shum, W. M. Campbell, and D. A. Reynolds, « Large-scale community detection on speaker content graphs », *in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 7716–7720.

[76] G. Dupuy, S. Meignier, P. Deléglise, and Y. Esteve, « Recent improvements on ILP-based clustering for broadcast news speaker diarization », 2014.

[77] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, « Speaker diarization with lstm », *in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 5239–5243.

[78] H. Ning, M. Liu, H. Tang, and T. S. Huang, « A spectral clustering approach to speaker diarization », *in Ninth International Conference on Spoken Language Processing*, 2006.

[79] J. H. Ward Jr, « Hierarchical grouping to optimize an objective function », *Journal of the American statistical association*, vol. 58, *301*, pp. 236–244, 1963.

[80] T. J. Park, K. J. Han, M. Kumar, and S. Narayanan, « Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap », *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[81] Q. Lin, R. Yin, M. Li, H. Bredin, and C. Barras, « LSTM based Similarity Measurement with Spectral Clustering for Speaker Diarization », *in Annual Conference of the International Speech Communication Association*, 2019.

[82] F. Landini, J. Profant, M. Diez, and L. Burget, « Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks », *Computer Speech & Language*, vol. 71, p. 101 254, 2022.

[83] M. Diez, L. Burget, and P. Matejka, « Speaker Diarization based on Bayesian HMM with Eigenvoice Priors. », *in Odyssey*, 2018, pp. 147–154.

[84] N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, « The third DIHARD diarization challenge », *arXiv preprint arXiv:2012.01477*, 2020.

[85] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, « Fully supervised speaker diarization », *in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 6301–6305.

[86] Z. Huang, S. Watanabe, Y. Fujita, P. Garcıa, Y. Shao, D. Povey, and S. Khudanpur, « Speaker diarization with region proposal network », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 6514–6518.

[87] S. Ren, K. He, R. Girshick, and J. Sun, « Faster r-cnn: Towards real-time object detection with region proposal networks », *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[88] T. von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, « All-neural online source separation, counting, and diarization for meeting analysis », *in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 91–95.

[89] K. Kinoshita, M. Delcroix, S. Araki, and T. Nakatani, « Tackling real noisy reverberant meetings with all-neural source separation, counting, and diarization system », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 381–385.

[90] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, « End-to-End Neural Speaker Diarization with Permutation-Free Objectives », *Proc. Interspeech 2019*, pp. 4300–4304, 2019.

[91] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, « End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors », *Proc. Interspeech 2020*, pp. 269–273, 2020.

[92] K. Kinoshita, M. Delcroix, and N. Tawara, « Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds », *in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 7198–7202.

[93] M. Ferras and H. Boudard, « Speaker diarization and linking of large corpora », *in 2012 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2012, pp. 280–285.

[94] M. Ferras, S. Madikeri, and H. Bourlard, « Speaker diarization and linking of meeting data », *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, *11*, pp. 1935–1945, 2016.

[95] G. Le Lan, D. Charlet, A. Larcher, and S. Meignier, « Iterative PLDA adaptation for speaker diarization », *in Interspeech 2016*, vol. 2016, 2016, pp. 2175–2179.

[96] G. Dupuy, S. Meignier, and Y. Esteve, « Is incremental cross-show speaker diarization efficient for processing large volumes of data? », *in Interspeech*, 2014.

[97] S. Meignier, J.-F. Bonastre, and I. Magrin-Chagnolleau, « Speaker utterances tying among speaker segmented audio documents using hierarchical classification: towards speaker indexing of audio databases », *in Seventh International Conference on Spoken Language Processing*, 2002.

[98] H. Ghaemmaghami, D. Dean, S. Sridharan, and D. A. van Leeuwen, « A study of speaker clustering for speaker attribution in large telephone conversation datasets », *Computer Speech & Language*, vol. 40, pp. 23–45, 2016.

[99] Z. Chen and B. Liu, « Lifelong machine learning », *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 12, *3*, pp. 1–207, 2018.

[100] P.-A. Broux, S. Petitrenaud, S. Meignier, J. Carrive, and D. Doukhan, « Evaluating human corrections in a computer-assisted speaker diarization system », *Language Resources and Evaluation*, vol. 55, *1*, pp. 151–172, 2021.

[101] D. L. Silver, Q. Yang, and L. Li, « Lifelong machine learning systems: Beyond learning algorithms », *in 2013 AAAI spring symposium series*, 2013.

[102] P.-H. Su, M. Gasic, N. Mrkšić, L. M. R. Barahona, S. Ultes, D. Vandyke, T.-H. Wen, and S. Young, « On-line Active Reward Learning for Policy Optimisation in Spoken Dialogue Systems », *in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2431–2441.

[103] D. Hakkani-Tür, G. Riccardi, and A. Gorin, « Active learning for automatic speech recognition », *in 2002 IEEE international conference on acoustics, speech, and signal processing*, IEEE, vol. 4, 2002, pp. IV–3904.

[104] A. Holzinger, M. Plass, M. Kickmeier-Rust, K. Holzinger, G. C. Crişan, C.-M. Pintea, and V. Palade, « Interactive machine learning: experimental evidence for the human in the algorithmic loop », *Applied Intelligence*, vol. 49, *7*, pp. 2401–2414, 2019.

[105] Y. Prokopalo, S. Meignier, O. Galibert, L. Barrault, and A. Larcher, « Evaluation of lifelong learning systems », *in International Conference on Language Resources and Evaluation*, 2020.

[106] M. Wang, F. Min, Z.-H. Zhang, and Y.-X. Wu, « Active learning through density clustering », *Expert systems with applications*, vol. 85, pp. 305–317, 2017.

[107] P. Bachman, A. Sordoni, and A. Trischler, « Learning algorithms for active learning », *in international conference on machine learning*, PMLR, 2017, pp. 301–310.

[108] F. Olsson, « A literature survey of active machine learning in the context of natural language processing », 2009.

[109] J. Thomason, A. Padmakumar, J. Sinapov, J. Hart, P. Stone, and R. J. Mooney, « Opportunistic active learning for grounding natural language descriptions », *in Conference on Robot Learning*, PMLR, 2017, pp. 67–76.

[110] K. Xie, C. Chang, L. Ren, L. Chen, and K. Yu, « Cost-sensitive active learning for dialogue state tracking », *in Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, 2018, pp. 209–213.

[111] Y. Wang and B. Wu, « Active machine learning approach for crater detection from planetary imagery and digital elevation models », *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, *8*, pp. 5777–5789, 2019.

[112] A. Holzinger, « Interactive machine learning for health informatics: when do we need the human-in-the-loop? », *Brain Informatics*, vol. 3, *2*, pp. 119–131, 2016.

[113] G. Riccardi and D. Hakkani-Tur, « Active learning: Theory and applications to automatic speech recognition », *IEEE transactions on speech and audio processing*, vol. 13, *4*, pp. 504–511, 2005.

[114] C. Wu, R. W. Ng, O. S. Torralba, and T. Hain, « Analysing acoustic model changes for active learning in automatic speech recognition », *in 2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, IEEE, 2017, pp. 1–5.

[115] J. Bang, H. Kim, Y. Yoo, and J.-W. Ha, « Efficient active learning for automatic speech recognition via augmented consistency regularization », *arXiv e-prints*, arXiv–2006, 2020.

[116] J. Huang, R. Child, V. Rao, H. Liu, S. Satheesh, and A. Coates, « Active learning for speech recognition: the power of gradients », *arXiv preprint arXiv:1612.03226*, 2016.

[117] K. Shinoda, H. Murakami, and S. Furui, « Speaker adaptation based on two-step active learning », *in Tenth Annual Conference of the International Speech Communication Association*, 2009.

[118] D. G. Karakos, S. Novotney, L. Z. 0002, and R. M. Schwartz, « Model Adaptation and Active Learning in the BBN Speech Activity Detection System for the DARPA RATS Program. », *in INTERSPEECH*, 2016, pp. 3678–3682.

[119] S. H. Shum, N. Dehak, and J. R. Glass, « Limited labels for unlimited data: Active learning for speaker recognition », *in Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[120] M. Abdelwahab and C. Busso, « Active learning for speech emotion recognition using deep neural network », *in 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, IEEE, 2019, pp. 1–7.

[121] E. Yilmaz, M. McLaren, H. van den Heuvel, and D. A. van Leeuwen, « Language diarization for semi-supervised bilingual acoustic model training », *in 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2017, pp. 91–96.

[122] W. Han, E. Coutinho, H. Ruan, H. Li, B. Schuller, X. Yu, and X. Zhu, « Semi-supervised active learning for sound classification in hybrid learning environments », *PloS one*, vol. 11, *9*, e0162075, 2016.

[123] N. Asghar, P. Poupart, X. Jiang, and H. Li, « Deep Active Learning for Dialogue Generation », *in Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, 2017, pp. 78–83.

[124] E. Agirre, A. Otegi, C. Pradel, S. Rosset, A. Penas, and M. Cieliebak, « LIHLITH: learning to interact with humans by lifelong interaction with humans », *Procesamiento del Lenguaje Natural*, vol. 63, pp. 147–150, 2019.

[125] S. Hao, J. Lu, P. Zhao, C. Zhang, S. C. Hoi, and C. Miao, « Second-order online active learning and its applications », *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, *7*, pp. 1338–1351, 2017.

[126] Y. Zhang, P. Zhao, J. Cao, W. Ma, J. Huang, Q. Wu, and M. Tan, « Online adaptive asymmetric active learning for budgeted imbalanced data », *in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2768–2777.

[127] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong, « Stream-based joint exploration-exploitation active learning », *in 2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 1560–1567.

[128] E. Weigl, W. Heidl, E. Lughofer, T. Radauer, and C. Eitzinger, « On improving performance of surface inspection systems by online active learning and flexible classifier updates », *Machine Vision and Applications*, vol. 27, *1*, pp. 103–127, 2016.

[129] T. Kim, I. Hwang, H. Lee, H. Kim, W.-S. Choi, J. J. Lim, and B.-T. Zhang, « Message Passing Adaptive Resonance Theory for Online Active Semi-supervised Learning », *in International Conference on Machine Learning*, PMLR, 2021, pp. 5519–5529.

[130] F. Shen, H. Yu, K. Sakurai, and O. Hasegawa, « An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network », *Neural Computing and Applications*, vol. 20, *7*, pp. 1061–1074, 2011.

[131] A. Goldberg, X. Zhu, A. Furger, and J.-M. Xu, « Oasis: Online active semi-supervised learning », *in Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, 2011.

[132] M. Seurin, F. Strub, P. Preux, and O. Pietquin, « A Machine of Few Words Interactive Speaker Recognition with Reinforcement Learning », *in Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.

[133] P.-A. Broux, D. Doukhan, S. Petitrenaud, S. Meignier, and J. Carrive, « An active learning method for speaker identity annotation in audio recordings », *in 1st International Workshop on Multimodal Media Data Analytics (MMDA 2016)*, 2016.

[134] ——, « Computer-assisted speaker diarization: How to evaluate human corrections », *in LREC 2018, Eleventh International Conference on Language Resources and Evaluation*, 2018.

[135] C. Yu and J. H. Hansen, « Active learning based constrained clustering for speaker diarization », *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, *11*, pp. 2188–2198, 2017.

[136] S. Renals, T. Hain, and H. Bourlard, « Recognition and understanding of meetings the AMI and AMIDA projects », *in 2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, IEEE, 2007, pp. 238–247.

[137] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, « The fifth CHiME speech separation and recognition challenge: dataset, task and baselines », *in Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1561–1565.

[138] J. H. Hansen, A. Sangwan, A. Joglekar, A. E. Bulut, L. Kaushik, and C. Yu, « Fearless Steps: Apollo-11 Corpus Advancements for Speech Technologies from Earth to the Moon. », *in Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 2758–2762.

[139] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, « Continuous speech separation: dataset and analysis », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7284–7288.

[140] Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu, *et al.*, « AISHELL-4: An Open Source Dataset for Speech Enhancement, Separation, Recognition and Speaker Diarization in Conference Scenario », *arXiv preprint arXiv:2104.03603*, 2021.

[141] E. Lleida, A. Ortega, A. Miguel, V. Bazán-Gil, C. Pérez, M. Gómez, and A. De Prada, « Albayzin 2018 evaluation: the iberspeech-RTVE challenge on speech technologies for spanish broadcast media », *Applied Sciences*, vol. 9, *24*, p. 5412, 2019.

[142] A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard, « The REPERE Corpus: a multimodal corpus for person recognition. », *in International Conference on Language Resources and Evaluation (LREC)*, 2012, pp. 1102–1107.

[143] G. Gravier, J.-F. Bonastre, E. Geoffrois, S. Galliano, K. McTait, and K. Choukri, « The ESTER Evaluation Campaign for the Rich Transcription of French Broadcast News. », *in International Conference on Language Resources and Evaluation (LREC)*, 2004.

[144] G. Gravier, G. Adda, N. Paulson, M. Carré, A. Giraudel, and O. Galibert, « The ETAPE corpus for the evaluation of speech-based TV content processing in the French language », *in International Conference on Language Resources, Evaluation and Corpora*, 2012.

[145] E. Timofeeva, E. Evseeva, V. Zaluskaia, V. Kapranova, S. Astapov, and V. Kabarov, « Improvement of Speaker Number Estimation by Applying an Overlapped Speech Detector », *in International Conference on Speech and Computer*, Springer, 2021, pp. 692–703.

[146] C. Peng, X. Wu, and T. Qu, « Competing Speaker Count Estimation on the Fusion of the Spectral and Spatial Embedding Space », *Proc. Interspeech 2020*, pp. 3077–3081, 2020.

[147] A. Canavan, D. Graff, and G. Zipperlen, « CALLHOME American English Speech LDC97S42 », *Linguistic Data Consortium*, 1997. DOI: doi.org/10.35111/exq3-x930.

[148] J. S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, « Spot the conversation: speaker diarisation in the wild », *in Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 299–303.

[149] L. Bullock, H. Bredin, and L. P. Garcia-Perera, « Overlap-aware diarization: Resegmentation using neural end-to-end overlapped speech detection », *in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 7114–7118.

[150] M. Kunešová, M. Hrúz, Z. Zajıc, and V. Radová, « Detection of overlapping speech for the purposes of speaker diarization », *in International Conference on Speech and Computer*, Springer, 2019, pp. 247–257.

[151] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, *et al.*, « The AMI meeting corpus », *in Proceedings of the 5th international conference on methods and techniques in behavioral research*, Citeseer, vol. 88, 2005, p. 100.

[152] J. S. Garofolo, J. G. Fiscus, A. F. Martin, D. S. Pallett, and M. A. Przybocki, « NIST Rich Transcription 2002 Evaluation: A Preview. », *in LREC*, 2002.

[153] O. Galibert, « Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. », *in INTERSPEECH*, 2013, pp. 1131–1134.

[154] M. Cettolo, M. Vescovi, and R. Rizzi, « Evaluation of BIC-based algorithms for audio segmentation », *Computer Speech & Language*, vol. 19, *2*, pp. 147–170, 2005.

[155] A. Krogh and J. Vedelsby, « Neural network ensembles, cross validation, and active learning », *in Advances in neural information processing systems*, 1995, pp. 231–238.

[156] A. Siddhant and Z. C. Lipton, « Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study », *in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2904–2909.

[157] T. Drugman, J. Pylkkönen, and R. Kneser, « Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models », *Interspeech 2016*, pp. 2318–2322, 2016.

[158] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, « The power of ensembles for active learning in image classification », *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9368–9377.

[159] R. Pérez-Dattari, C. Celemin, J. Ruiz-del-Solar, and J. Kober, « Interactive learning with corrective feedback for policies based on deep neural networks », *in International Symposium on Experimental Robotics*, Springer, 2018, pp. 353–363.

[160] C. Celemin and J. Ruiz-del-Solar, « An interactive framework for learning continuous actions policies based on corrective feedback », *Journal of Intelligent & Robotic Systems*, vol. 95, *1*, pp. 77–97, 2019.

[161] C. Barras, E. Geoffrois, Z. Wu, and M. Liberman, « Transcriber: development and use of a tool for assisting speech corpora production », *Speech Communication*, vol. 33, *1-2*, pp. 5–22, 2001.

[162] E. Geoffrois, « Evaluating interactive system adaptation », *in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016, pp. 256–260.

[163] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, « BLEU: A Method for Automatic Evaluation of Machine Translation », *in Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, ACL, 2002.

[164] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, « Power to the people: The role of humans in interactive machine learning », *Ai Magazine*, vol. 35, *4*, pp. 105–120, 2014.

[165] M. Wang and X.-S. Hua, « Active learning in multimedia annotation and retrieval: A survey », *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, *2*, pp. 1–21, 2011.

[166] A. Larcher, K. A. Lee, and S. Meignier, « An extensible speaker identification sidekit in python », *in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 5095–5099.

[167] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer Science & Business Media, 2012, vol. 247.

[168] J. Gonzalez-Rodriguez, « Evaluating automatic speaker recognition systems: An overview of the nist speaker recognition evaluations (1996-2014) », *Loquens*, 2014.

[169] Z.-H. Tan, N. Dehak, *et al.*, « rVAD: An unsupervised segment-based robust voice activity detection method », *Computer Speech & Language*, vol. 59, pp. 1–21, 2020.

[170] R. Hibare and A. Vibhute, « Feature extraction techniques in speech processing: a survey », *International Journal of Computer Applications*, vol. 107, *5*, 2014.

[171] M. Diez, L. Burget, F. Landini, and J. Černocky, « Analysis of speaker diarization based on bayesian hmm with eigenvoice priors », *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 355–368, 2019.

[172] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, « Joint factor analysis versus eigenchannels in speaker recognition », *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, *4*, pp. 1435–1447, 2007.

[173] F. Bimbot, I. Magrin-Chagnolleau, and L. Mathan, « Second-order statistical measures for text-independent speaker identification », *Speech communication*, vol. 17, *1-2*, pp. 177–192, 1995.

[174] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, « Speaker verification using adapted Gaussian mixture models », *Digital signal processing*, vol. 10, *1-3*, pp. 19–41, 2000.

[175] S. Berg, D. Kutra, T. Kroeger, C. N. Straehle, B. X. Kausler, C. Haubold, M. Schiegg, J. Ales, T. Beier, M. Rudy, *et al.*, « Ilastik: interactive machine learning for (bio) image analysis », *Nature Methods*, vol. 16, *12*, pp. 1226–1232, 2019.

[176] S. Teso and K. Kersting, « Explanatory interactive machine learning », *in Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 239–245.

[177] M. Gillies, « Understanding the role of interactive machine learning in movement interaction design », *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 26, *1*, pp. 1–34, 2019.

[178] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, « The First DIHARD Speech Diarization Challenge », *in Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018.

[179] A. Nagrani, J. S. Chung, and A. Zisserman, « VoxCeleb: A Large-Scale Speaker Identification Dataset », *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[180] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, « A review of speaker diarization: Recent advances with deep learning », *Computer Speech & Language*, vol. 72, p. 101 317, 2022.

[181] S. Horiguchi, N. Yalta, P. Garcia, Y. Takashima, Y. Xue, D. Raj, Z. Huang, Y. Fujita, S. Watanabe, and S. Khudanpur, « The Hitachi-JHU DIHARD III system: Competitive end-to-end neural diarization and x-vector clustering systems combined by DOVER-Lap », *arXiv preprint arXiv:2102.01363*, 2021.

[182] S. Horiguchi, P. Garcia, Y. Fujita, S. Watanabe, and K. Nagamatsu, « End-to-end speaker diarization as post-processing », *in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 7188–7192.

[183] S. Horiguchi, Y. Fujita, and K. Nagamatsu, « Utterance-Wise Meeting Transcription System Using Asynchronous Distributed Microphones », *in Proc. Interspeech 2020*, 2020, pp. 344–348. DOI: 10.21437/Interspeech.2020-1050.

[184] F. Valente, P. Motlicek, and D. Vijayasenan, « Variational Bayesian speaker diarization of meeting recordings », *in 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2010, pp. 4954–4957.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

LIUM
Laboratoire Informatique
Université du Mans

Titre : Corrections assistées par l'humain pour la diarisation incrementale de collection

Mot clés : Apprentisage tout au long de la vie, Apprentisage actif, Diarisation, Diarisation incrementale de collection

Résumé : La tâche de diarisation des locuteurs, également appelée segmentation et regroupement en locuteurs, consiste à déterminer le nombre de locuteurs et le moment où ils parlent dans un document audio ou un ensemble de documents audio. Cette tâche intéresse de nombreuses entreprises souhaitant indexer leurs contenus audiovisuels, améliorer l'accessibilité et fournir des annotations pour leur contenu audio. De plus, la diarisation du locuteur est utilisée comme étape de prétraitement pour de nombreuses autres tâches de traitement de la parole telles que la reconnaissance de parole, la reconnaissance du locuteur et des émotions, etc.

Pour être valable, la qualité des annotations des documents audio doit atteindre un niveau suffisant qui n'est, la plupart du temps, pas encore atteint par les systèmes de diarisation automatiques du locuteur à l'état de l'art. Pour atteindre les performances souhaitées, de nombreuses entreprises emploient des annotateurs humains pour produire des annotations manuelles à partir de zéro ou, afin de réduire le coût du processus d'annotation, demandent à l'expert du domaine humain de corriger la sortie d'un système de diarisation automatique. Néanmoins, l'intervention humaine est généralement chronophage et très coûteuse en raison de la difficulté de la tâche et de l'énorme quantité de données à traiter.

Même lors de la correction d'une annotation automatique existante, le processus manuel est extrêmement long, coûteux et fastidieux pour plusieurs raisons. Tout d'abord, l'expert du domaine humain ne sait pas quelle partie de l'annotation corriger et doit souvent écouter l'intégralité du document audio pour vérifier l'exactitude des annotations. Ce processus est très sous optimal. La deuxième raison est qu'un système automatique est susceptible d'effectuer de nombreuses erreurs du même type que l'expert du domaine humain devra corriger une par une au fil du temps. Cela rend la tâche répétitive et peut-être très frustrante pour l'annotateur.

Cette recherche a été réalisée dans le cadre du projet européen ChistERA ALLIES, qui vise à jeter les bases du développement de systèmes intelligents autonomes maintenant leurs performances dans le temps. Un tel système non supervisé devrait être capable de se mettre à jour automatiquement et d'effectuer une auto-évaluation pour être au courant de l'évolution de sa propre acquisition de connaissances. Il doit s'adapter à un environnement changeant en suivant un scénario d'apprentissage donné qui équilibre l'importance de la performance sur les données passées et présentes pour éviter une régression indésirable. De tels systèmes ne pourraient être développés sans des métriques et des protocoles adaptés permettant leur évaluation objective et reproductible. Cette évaluation doit évaluer en continu la performance sur la tâche donnée et quantifier l'effort requis pour l'atteindre en matière de données non supervisées collectées par le système et d'interaction avec les humains dans le cas de l'apprentissage actif. Le projet ALLIES vise à développer, évaluer et diffuser ces métriques et protocoles. Notre objectif dans le projet était d'appliquer le concept d'apprentissage tout au long de la vie assistée par l'humaine à la tâche de diarisation du locuteur. Plus précisément, notre travail vise à fournir un moyen efficace d'interagir entre le système de diarisation et un expert du domaine humain afin d'améliorer la qualité de la diarisation tout en limitant la quantité d'effort humain nécessaire.

Pour mener à bien la tâche de diarisation des locuteurs d'apprentissage tout au long de la vie, nous avons dû trouver des solutions à plusieurs problèmes.

Le premier problème auquel nous avons été confrontés est l'absence d'une définition standard de

l'apprentissage tout au long de la vie assistée par l'humain. Dans la littérature, il existe diverses définitions, principalement développées pour le domaine des systèmes de dialogue. Il fallait en proposer une alternative, qui corresponde mieux au périmètre du projet ALLIES. Une autre question était la diversité des différents types d'interactions entre les systèmes automatiques et les humains, qui n'avaient pas de nomenclature commune dans la littérature.

Après avoir fourni ces définitions, nous avons été confrontés à l'absence du matériel nécessaire pour développer et évaluer des systèmes de diarisation de locuteurs d'apprentissage tout au long de la vie assistée par l'humain. Il n'y avait pas de corpus, de protocoles ni de mesures pour prendre en compte la spécificité du processus d'apprentissage tout au long de la vie. Une attention particulière a été accordée à la métrique d'évaluation car les métriques existantes ne prenaient pas en compte l'interaction avec l'expert humain ou le processus d'apprentissage tout au long de la vie.

Finalement, l'une des principales questions était le développement du système de diarisation assistée par l'humain elle-même. Un tel système nécessite des méthodes et des stratégies spécifiques pour interagir avec l'expert du domaine humain qui ne sont pas bien développées, en particulier dans le domaine de la diarisation du locuteur.

Dans ce manuscrit, nous proposons notre point de vue sur la définition des systèmes intelligents d'apprentissage tout au long de la vie. Notre point de vue se concentre sur l'optimisation du modèle pour les futures données entrantes et sur la minimisation de l'effet d'oubli, lorsque les nouvelles versions du modèle fonctionnent moins bien que les versions précédentes sur les données précédentes. Nous avons également proposé une nomenclature des différents types d'interactions entre le système intelligent et l'expert humain.

Nous avons développé un corpus conçu pour l'évaluation des systèmes de diarisation de l'apprentissage tout au long de la vie. Le corpus proposé a un certain nombre de propriétés telles que les horodatages et le nombre élevé de locuteurs récurrents annotés, ce qui permet de le traiter dans l'ordre chronologique et d'apprendre de nouvelles informations à partir des changements de voix des locuteurs récurrents. Ces propriétés rendent le corpus proposé unique. Il s'agit du seul corpus public pouvant être utilisé pour évaluer la tâche de diarisation avec apprentissage tout au long de la vie.

Un autre apport de notre travail est la métrique d'évaluation des systèmes assistés par l'humaine. Une proposition de métrique a été développée pour le cas général, c'est-à-dire pour estimer la performance sur différentes tâches. Elle a été appliquée non seulement pour la tâche de diarisation mais aussi sur la tâche de traduction automatique (métrique BLEU) dans le cadre du projet ALLIES. Le terme de pénalisation estime la quantité d'informations fournies par l'expert humain en dans la même unité que la métrique correspondante et pénalise le score final pour mettre en évidence l'effet de généralisation du système assisté par l'humain. Nous avons également présenté plusieurs protocoles grâce auxquels il est possible d'effectuer l'évaluation de différents systèmes d'apprentissage tout au long de la vie assistés par l'humain.

La principale contribution de nos travaux réside dans le développement des méthodes de diarisation intra-show et inter-show assistées par l'humain. En intra-show, le système assisté par l'humain est fondé sur l'analyse du dendrogramme obtenu lors de l'étape finale du regroupement agglomératif hiérarchique, puis en posant des questions à des experts humain du domaine afin d'améliorer le regroupement. Pour cette tâche, nous avons proposé diverses stratégies pour sélectionner la question à poser et pour sélectionner les segments qui devraient être comparés par l'expert du domaine. Aussi, nous avons testé différents critères d'arrêt pour décider quand il n'est pas raisonnable de continuer à poser des questions. Pour la diarisation intra-show, nous avons obtenu une réduction du DER allant jusqu'à 18,83% et une réduction du DER pénalisé jusqu'à 9,94% par rapport aux systèmes de base. Les résultats sur le DER pénalisé peuvent également être interprétés comme corrigeant près de 10% des erreurs uniquement en généralisant à partir des informations obtenues auprès de l'expert humain.

Pour la diarisation inter-show, le système assisté par l'humain est basé sur l'analyse de la matrice de pseudo-distance basée sur les représentations du locuteur. Pour cette tâche, nous avons testé différentes stratégies de représentation et de sélection des locuteurs qui doivent être comparés pour

résoudre le problème de la variabilité entre les émissions. Nous avons également testé différents critères d'arrêt pour poser des questions. Pour la diarisation croisée, nous avons obtenu une réduction encore plus importante : jusqu'à 34,19% relatifs pour les DER et jusqu'à 14,31% relatifs pour les DER pénalisés. Les résultats sur le DER pénalisé montrent que nous sommes parvenus à corriger 14,31% d'erreurs par généralisation des informations obtenues auprès de l'expert humain. Pour les deux tâches, des tests ont été appliqués sur différents systèmes de base pour avoir plus de détails sur la performance des stratégies proposées.

Ces résultats ouvrent la voie à de nouvelles recherches. Une des perspectives est de combiner les stratégies intra-show et inter-show. Il est possible de les utiliser séquentiellement, mais il serait plus intéressant de les utiliser simultanément et d'éviter les questions éventuellement inutiles. Un tel résultat pourrait être atteint en comparant des segments de l'émission en cours entre eux et avec les représentations des locuteurs des émissions précédentes. En d'autres termes, fusionnez la résolution de deux tâches : la diarisation intra-show et inter-show en une seule étape.

L'étape importante restante à résoudre est le développement d'une méthode d'adaptation tout au long de la vie pour la diarisation du locuteur. Nous avons tenté de créer une telle solution (non rapportée dans ce manuscrit), mais nous avons été bloqués par les faibles performances de la diarisation inter-show en raison de la forte variabilité inter-show. Nous nous sommes concentrés sur la solution à ce problème en utilisant la diarisation inter-show assistée par l'humain. Les résultats obtenus peuvent permettre de créer le pipeline complet d'apprentissage tout au long de la vie et d'utiliser les informations recueillies auprès de l'expert humain, non seulement pour améliorer les résultats actuels, mais aussi pour adapter le système afin qu'il soit plus performant en général. Il serait également intéressant d'adapter les méthodes proposées pour des approches neuronales de bout en bout, car cela peut ouvrir la voie à un processus d'adaptation du système plus simple et plus efficac.

Title: Human assisted correction for speaker diarization of an incremental collection of documents

Keywords: Lifelong learning, Active learning, Speaker diarization, Diarization of an incremental collection of documents

Abstract: This research aims at designing an autonomous speaker diarization system able to adapt and evaluate itself. The goal of this research was to apply the concept of human-assisted lifelong learning to the speaker diarization task, also known as speaker segmentation and clustering. More specifically, this work aims at designing an efficient way of interaction between the automatic diarization system and a human domain expert to improve the quality of diarization generated by an automatic system while limiting the workload for the human domain expert. This manuscript proposes an alternative point of view on the definition of the lifelong learning intelligent systems, a dataset designed for evaluation of the lifelong learning diarization systems and the metric for evaluation of human-assisted systems. The main contribution of this work lies in the development of the human-assisted within-show and cross-show diarization methods.