THE ALLIES EVALUATION PLAN FOR AUTONOMOUS SPEAKER DIARIZATION SYSTEMS

Anthony Larcher
LIUM - Le Mans Université
Avenue Olivier Messiaen
F-72085 – LE MANS CEDEX 9
anthony.larcher@univ-lemans.fr

Olivier Galibert
LNE
Olivier.Galibert@lne.fr

March 4, 2020

ABSTRACT

The ALLIES project aims at catalysing the development of autonomous lifelong intelligent systems by providing the community with scenarios, evaluation plans and metrics to evaluate those systems. ALLIES focuses on two tasks: speaker segmentation, and machine translation. The speaker segmentation evaluation relies on a new corpus of audio-visual documents (news, debates, talk show...) from the French channel LCP. The evaluation is coordinated by the LNE (Laboratoire national de métrologie et d'essais, France), LIUM (Le Mans Université, France), IDIAP (Switzerland) and UPC (Spain). This evaluation plan defines the tasks to be benchmarked and includes all necessary information to participants.

1 Introduction

Speech segmentation with speaker clustering, referred to as speaker diarization, is a key pre-processing step for several speech technologies including enriched automatic speech recognition (ASR) or spoken document retrieval (SDR) in very large multimedia repositories. The base accuracy of such systems is of essential to allow applications to perform adequately in real-world environments.

Speaker diarization systems rely on a data driven knowledge and their development requires competences in machine learning as well as a specific domain expertise. Performance of such systems usually degrades across time as the distribution of incoming data moves away from the initial training data (changes in accents, in recording conditions, etc). Thus sustaining system performance across time requires frequent interventions of machine learning experts which makes the maintenance of such system very costly.

The ALLIES evaluation focuses on freeing speaker diarization systems from the need of machine learning expert interventions upon two axes:

Diarization across time automatic systems use the stream of incoming data to update their knowledge and adapt to new data in order to sustain performance across time;

Lifelong learning diarization defined as a task of diarization across time that allows interaction of the system with a human domain expert who can interact with the automatic system in two modes:

- **Interactive learning with user initiative** given the current knowledge of an automatic system and a set of documents to process, a human domain expert provides corrections on the automatic systems outputs until the system produces a good enough output;
- **Active learning with system initiative** the system itself ask the human domain expert corrections of its diarization hypothesis.

The two tasks of the ALLIES evaluation are described in section 2 together with the scenarios and evaluation metrics.

2 Tasks

The two tasks of the ALLIES [1] evaluation are focusing on the speaker diarization task where the question to be answered by the automatic system is: "*Who speaks when?*"

In the past, many evaluations have been organized to benchmark existing systems and the ALLIES evaluation aims at moving towards evaluation of systems in the real world. For this purpose two flavours of the classical speaker diarization task are considered in the ALLIES evaluation. Participants are free to participate to one or two tasks.

The standard task of speaker diarization is described here to better introduce the two tasks considered in the ALLIES evaluation, i.e.:

- Diarization across time
- Lifelong learning diarization

2.1 Introduction to standard speaker diarization

The standard task of speaker diarization is described here as a reference for the two tasks considered in the ALLIES evaluation.

2.1.1 Description

In its most general form, speaker diarization is applied to a collection of audio files containing speech from multiple speakers as well as music and noises. Speakers can appear in multiple documents in the collection and be recorded under different conditions. The label given to a speaker must be unique across the entire collection of documents.

The diarization task consists, for each speaker of the document collection, to detect in a non-supervised way the temporal regions where (s)he speaks. Each temporal region forming a segment is then annotated with an abstract and unique label representing it.

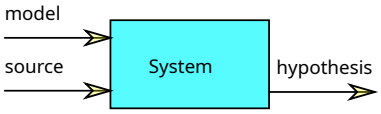

Figure 1: Standard usage of an automatic speaker diarization system. Given a current model and an audio source file, the system returns an hypothesis.

2.2 Training data

Automatic systems make use of knowledge-based models that are trained using two collections of audio documents referred to as the *training set* and *development set*. Automatic systems must use only the provided documents and *references* from the *training set* to learn all necessary models. It is recommended but not required for the developers to use the development set for tuning and inter-system comparisons before the evaluation.

2.3 Standard Evaluation protocol

The protocol is defined by the evaluation scenario and the associated evaluation metrics.

2.3.1 Standard evaluation scenario

The trained system will be given new audio files sequentially. It must process them and return an hypothesis as a list of time segments of speech associated with a speaker label. The label must be associated with the speaker and reused only when the speaker talks in a subsequent show.

2.3.2 Standard evaluation metric

Traditionally, the performance of diarization systems are given in terms of Diarization Error Rate (DER). The DER is computed as the fraction of speaker time that is not correctly attributed to its speaker. This score will be computed over

[1] https://projets-lium.univ-lemans.fr/allies/

the document collection to be processed; including regions where more than one speaker is present (overlap regions). This score will be defined as the ratio of the overall diarization error time to the sum of the durations of the segments that are assigned to a speaker label. Given the data set to evaluate, each document is divided into contiguous segments at all speaker change points found in both the reference and the hypothesis, and the diarization error time for each segment n is defined as:

$$E(n) = T(n)[\max(N_{ref}(n), N_{sys}(n)) - N_{Correct}(n)] \quad (1)$$

where $T(n)$ is the duration of segment n, $N_{ref}(n)$ is the number of speakers that are present in segment n of the reference file, $N_{sys}(n)$ is the number of system speakers that are present in segment n and $N_{Correct}(n)$ is the number of reference speakers in segment n correctly assigned by the diarization system.

$$DER = \frac{\sum_{n\in\Omega} E(n)}{\sum_{n\in\Omega}(T(n)N_{ref}(n))} \quad (2)$$

The diarization error time includes the time that is assigned to the wrong speaker (confusion), missed speech time and false alarm speech time.

Confusion Error Time : The Confusion Error Time is the amount of time that has been assigned to an incorrect speaker.

Missed Speech Time : The Missed Speech Time refers to the amount of time where speech is present but not labeled as belonging to a speaker by the diarizaton system.

False Alarm Time : The False Alarm Time is the amount of time where speech is not present but labeled as belonging to a speaker by the diarization system.

Speakers are not identified (no name is given), thus confusion error time cannot be computed immediately but requires a matching process that assigns each reference speaker to a unique speaker label as generated by the system. The matching used for scoring minimizes the CET, hence the DER.

Consecutive segments of the same speaker with a silence of less that 2 seconds come together and are considered as a single segment. A forgiveness collar of 0.25 s, before and after each reference boundary, will be considered in order to take into account both inconsistent human annotations and the uncertainty about when a speaker begins or ends.

2.4 Diarization across time

In this task, speaker diarization systems are evaluated across time without considering any human in the loop.

2.4.1 Description

The task of Diarization across time simulates the evaluation of an automatic system across time. The system is allowed to update its models using any audio data sent in, creating a better speech model or updating model clusters for instance, and generate a new version of them to handle the next show.

2.4.2 Evaluation scenario

The initial training process is similar to the one described above. The system is free to learn its models using all provided audio files and *references*. Audio files from the *training set* are provided with an additional information: a time stamp corresponding to their broadcasting time (day and time). This offer the possibility to the system to learn the evolution of data across time.

The *development set* consists now of a temporal sequence of audio documents $\{D_t\}_{t=1...T}$ provided with their time stamp t (day and time) in a chronological order. The automatic system must return one *hypothesis* for each document D_t before accessing the following document D_{t+1}. In the evaluation across time, the automatic system is allowed to adapt using all provided data including documents from the *development set* already processed; at all time, the system is free to come back to training data in order to mine for additional information.

The *hypothesis* returned by the system after each document D_t includes a segmentation as well as the speaker class for each segment. Speaker classes can correspond to some speakers previously seen in the *training set* or in the document from the *development set* that have already been processed (i.e., all document D_i with $i < t$) or be a new speaker class. The system must thus be able to detect new speaker classes if required. *Hypotheses* returned by the lifelong learning systems are not independent across documents anymore but depend on document already processed.

This protocol is described by Figure 2.

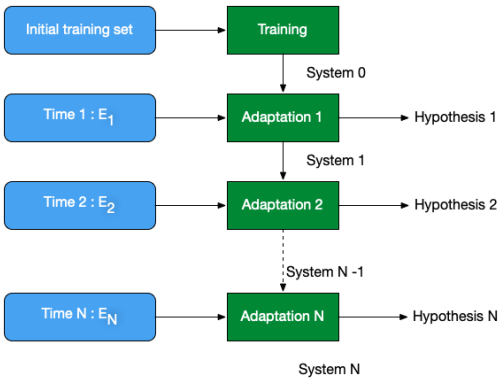

Figure 2: Evaluation across time. Given *training set* an initial version of the diarization system is trained (System 0). The N documents of the *development set* are processed chronologically. After each document E_t from the *development set*, the system returns a corresponding *hypothesis* t as well as a new version of the system (System t) that could benefit from knowledge extracted from E_t.

During the final evaluation, the *development set* is replaced by a third set of documents, disjoint from the *development set*, which is referred to as the *evaluation set*. Note that the *evaluation set* is time stamped and provided to the automatic system in the chronological order. The system is thus required to return a *hypothesis* t after processing each document E_t from the *evaluation set*.

2.4.3 Evaluation metric

Note that the performance of the system is now a sequence of DER for each document. Given a sequence of DER, performance of the systems will be given using a weighed average score. For this score, DER from each file is weighed by the duration of the file.

2.4.4 Data

For the task of diarization across time, participants will be provided with a training set and a development set of data that will be downloadable via a FTP protocol. Each show is given as an audio file in WAV format that comes with a UEM file defining the beginning and end of each audio segments to process. Additionally, MDTM files are given with a diarization reference in a standard format (described in appendix).

Participants are free to make any use of the following standard freely available datasets: VoxCeleb 1[2], VoxCeleb 2[3], MUSAN[4], RIR [5]. Training data must be restricted to the listed corpora.

2.5 Lifelong learning speaker diarization

2.5.1 Description

The protocol is similar to the one in *Across Time Diarization* except that the the system is allowed to performed *active* and *interactive learning* during its adaptation.

This task is **exclusively performed on the BEAT platform**. participants to this task must upload the source code of their system written in Python on the BEAT platform in order to run on the development data. A baseline system is provided as an example (see section 3.2).

The BEAT platform includes a user simulation that is used to evaluate human assisted learning: active or interactive learning as described in the following sections.

[2] `http://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox1.html`
[3] `http://www.robots.ox.ac.uk/~vgg/data/voxceleb/vox2.html`
[4] `http://www.openslr.org/resources/17/`
[5] `https://www.openslr.org/28/`

2.5.2 Active learning

We define active learning as a learning process initiated by the automatic system where this system is allowed to ask questions to a human operator referred to as the *user in the loop*.

In order to allow a fair and reproducible evaluation of automatic systems, a *user in the loop* simulator has been developed and integrated on the evaluation platform. When the system receives an audio file to process, it is allowed to ask the *user in the loop* one of the following questions:

- Is the same speaker talking at times <t1> and <t2>
- What are the boundaries of the speech segment containing time <t>

Each question has a cost, and the user will refuse to answer when the per-show cost is reached. In any case the system should eventually give its final hypothesis.

2.5.3 Interactive speaker diarization

We define interactive learning as a learning process that involves a human in the learning loop, observing the *hypothesis* returned by the system and spontaneously providing inputs meant to improve the *hypothesis*. This human is referred to as the *user in the loop*.

In order to allow a fair and reproducible evaluation of automatic systems, a *user in the loop* simulator has been developed and integrated on the evaluation platform. For each document D_t provided to the automatic system to process speaker diarization, the corresponding *reference*, R_t is provided to the *user in the loop* simulator.

Once the system provides a first hypothesis to the simulated user, the user then spontaneously replies by an information of one of the following types:

- The same speaker is talking at times <t1> and <t2>
- The boundaries of the speech segment containing time <t>

The system should then update the hypothesis and try again. Eventually, when the hypothesis is good or an internal cost is reached the user will stop this iterative process and the latest hypothesis will be retained as the final one.

Note that the information given by the human in the loop is the one that maximizes the potential DER correction. The human in the loop will provide information until possible gain in DER is lower than correction cost or until it has to return the same information twice.

2.5.4 Evaluation metric

Performance of the system is a sequence of DER for each document. The DER is computed on the final version of the hypothesis for each document penalized by the cost of interacting with the *user in the loop*.

Interaction cost The cost of interaction is measured in terms of DER given by the human in the loop. The cost is related to the duration of signal the human in the loop is correcting or validating when providing information to the system. When providing the information "The same speaker is talking at times <t1> and <t2>", we consider that the human in the loop has to listen two segments of 3 seconds centered on <t1> and <t2>. The corrected/validated duration is thus 6 seconds. When providing the boundaries of a speech segment, we consider that the human in the loop has to listen to the duration of the targeted segment with an additional 3 seconds before and after in order to assert the time of exact boundaries.

In order to compute the cost of interaction, it is proposed to compute two intermediate values: the corrected (S_{cor}) and the impaired (S_{imp}) scores. Computation of those scores is described on Figure 3

Let's assume that the system produces a first hypothesis (see Figure 3-A) and obtains the score S_{base} before applying any online AL/IAL. Starting the human assisted learning, the human in the loop corrects (or is ask to correct) part of the current hypothesis. This corrected part of the hypothesis is shown in Figure 3-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 in the loop might not be entirely wrong. The difference between S_{base} and S_{cor} corresponds to the decrease of DER resulting from the corrections provided by the human in the loop only.

This difference, $S_{base} - S_{cor}$, doesn't 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

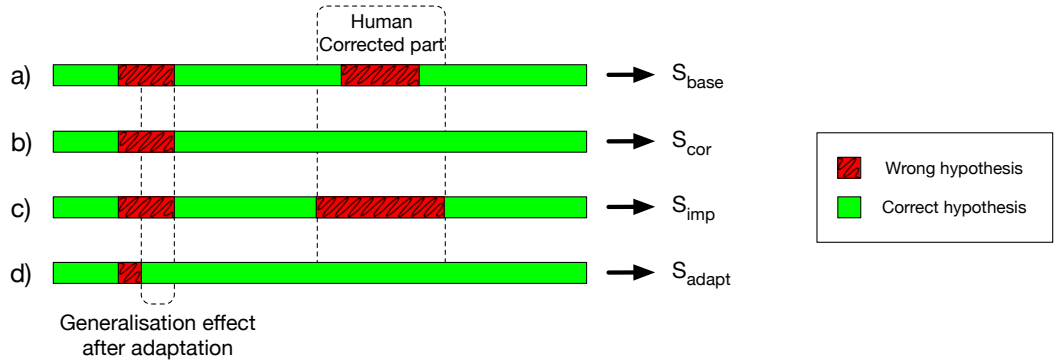

Figure 3: An hypothesis (a) produced by the system contains correct parts (green) ans errors (red) and obtains a score S_{base}. During human assisted learning, the human applies (or is asked to apply) corrections on a part of the hypothesis that might be partially correct. To penalise the system, we 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}.

of the current hypothesis shown on Figure 3-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 simulated on Figure 3-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 3

$$S_{pen} = S_{adapt} + (S_{imp} - S_{cor}) \tag{3}$$

Given a sequence of penalized scores (DER + penalization cost), the performance of the Lifelong learning systems will be given using an weighed average score that is the mean of all scores computed across time over the *evaluation set* weighed by the duration of the file.

2.5.5 Data

For the task of Lifelong learning diarization, participants will be provided with a development package and will have to push their final system on the online BEAT platform.

The development package includes training and development data sets that will be downloadable from via a FTP protocol. Each show is given as an audio file in WAV format that comes with a UEM file defining the beginning and end of each audio segments to process. Additionally, MDTM files are given with a diarization reference in a standard format (described in an appendix). Additionally, each WAV file comes with a flag that indicates what kind of human assisted adaptation mode is allowed for this specific file.

Training and development data are strictly limited to the two sets described above the design of systems must take this parameter into account as the amount of initial training data is limited.

2.5.6 Human adaptation process

When processing a WAV file, the automatic system will receive an "adaptation flag" that can take three values. Depending on the value of this flag, the system has to interact with the human in the loop as follows:

active the system must return an hypothesis and is free to ask a question to the human in the loop. Two types of questions are allowed (see section 2.5.2 or can be empty if the system decides not to ask anything.) The active learning process ends whenever the human in the loop returns an "end-of-interaction" signal. This will appears if the number of allowed questions is reached or if the system decides not to ask questions anymore.

interactive The system must return an hypothesis and will then receive a sequence of information from the human in the loop. After each information, the system has to return an hypothesis. The sequence will end on the human in the loop initiative. In any case, the information given by the human in the loop is the one that maximizes the potential DER correction. Note that the human in the loop will provide information until possible gain in DER is lower than correction cost or until it has to return the same information twice.

None in this mode, no interaction with the human in the loop is possible. The system can still perform unsupervised adaptation by using the incoming data.

3 The evaluation platform

The BEAT Platform (`https://www.idiap.ch/software/beat/platform`) is a European computing e-infrastructure for Open Science proposing a solution for open access, scientific information sharing and re-use including data and source code while protecting privacy and confidentiality. In the context of project ALLIES, BEAT will be used to define and quickly run experiments online, following the ALLIES evaluation protocol. You can browse public resources available on the platform by visiting the platform webpage. To use the BEAT platform for performing experiments you will need to register (for free). After logging in, it is recommend you follow the tutorial on available user documentation (`https://www.idiap.ch/software/beat/documentation`). The web platform user guide is of specific interest to those directly interacting with the web platform (`https://www.idiap.ch/software/beat/docs/beat/docs/stable/beat.web/doc/user/index.html`).

The platform provides its own computing (including GPU-capable) resources that can be freely used by participants. Datasets are stored within the computing system and are **not** accessible from outside. Algorithms uploaded to the platform are executed in a secure environment that provides temporary access to the data.

Samples of the development set will be provided to the participants to have a sense of the data quality and content.

3.1 Available computing resources

This is a summary of the current resources available on the online BEAT platform.

3.1.1 Machines

- 8 single core machines with 12GB RAM
- 1 ten cores machine with 120GB RAM
- 7 dual core machines with 30GB RAM and a NVIDIA Tesla K80 GPU with 12GB RAM

The infrastructure is flexible and can be modified to add and remove machines.

3.1.2 Queues

Name	Memory limit (MB)	Time limit (m)	Cores per slot	Max slots per user
GPU	12000	60	1	1
Long	60561	720	5	1
Default	12043	180	1	4

3.1.3 Execution Environments

The BEAT platform executes the user provided algorithms in what are called "execution environments". These environments are currently Docker based and provide a set of known packages which are precisely versioned so that two runs of the same algorithm will produce the same results. As such, updating an environment cannot happen at will during the development phase of the algorithm.

There are several reasons for that that are related to:

- Security
- Reproducibility
- Hardware

Images must be validated, created and tested. Only after that, they can be deployed which requires a variable amount of time depending on the size of the image.

An image where a dependency is changed becomes a new environment. This mean that the previous version must be kept in order for the experiment that uses it to be reproduced. Therefore changing several times a day for testing purpose is not a viable solution.

That's why local development is for. In case a library needs to be developed and tested on the platform, then the recommended way to do it is to create a BEAT Library object that can more easily be updated.

3.2 System integration

A baseline system is available on the BEAT platform.

Several documents are available on the evaluation webpage[6] including:

- the complete instructions to install a version of the BEAT framework in a local CONDA environment;
- instructions to locally run the baseline system using provided training and development data sets;
- documentation of the baseline system that explains how to modify it to replace with your own code.

4 General Evaluation Conditions

The organizers encourage the participation of all researchers interested in speaker diarization. All teams willing to participate in this evaluation must send an email to:
lifelong-speaker-evaluation@univ-lemans.fr
Indicating the following Information:

- Research group
- Institution
- Contact person
- E-mail

RESEARCH GROUP: before April 30th, 2020.

4.1 Data License Agreement

The ALLIES data is available to the evaluation participants and subject to the terms of a license agreement with LCP. The license agreement can be downloaded from `https://lium.univ-lemans.fr/allies-evaluation/` Participants must follow the process described on this page to register and access the data. ALLIES data will be made freely available on `https://dataset.ina.fr` after the evaluation

4.2 Evaluation Rules

4.2.1 Diarization across time

Access to the evaluation data will be granted on the 1st of June 2020. Participants must upload the output of their systems as a single TAR ball including one MDTM file per WAV file included in the evaluation dataset at last the 30th of June 2020, 23h59 GMT+1.

MDTM files must have the same name as related WAV file but with a different extension.

For each file <filename.wav>, the TAR ball must include a file <filename.mdtm>.

MDTM files must follow the format described in appendix. Each team can submit a primary system and is free to submit up to two contrastive systems. The file submitted for the primary system must be named as follow:

$participant - ID\_allies\_primary.tar$

Submissions for contrastive systems must be named as follosws:

[6] `https://lium.univ-lemans.fr/allies-evaluation/`

$participant - ID_allies_contrastive_1.tar$participant-ID_allies_contrastive$_2.tar$

Where $participant - ID$ is the name of the team chosen by the participant when registering.

4.2.2 Lifelong learning diarization

The online BEAT platform will open on the 1st of March 2020. Each participant can access the platform to run and modify its system on development data until 31st of May, 23h59 GMT +1. After this date, the systems uploaded on the platform will be run on the evaluation data to provide final results.

Each participant team for the Lifelong learning diarization task must upload on the BEAT platform at least a primary system but they can also submit up to two contrastive systems.

Each participant is free to run and modify its systems on the BEAT until the submission deadline (31st of May 2020) on the *development set*. Organizers will then proceed to the final evaluation on the *test set* by using the final versions of each system. The ranking of the evaluation will be done according to results of the primary systems for each core condition but the analysis of the results of the contrastive systems will be also processed and presented during the evaluation workshop. All participant sites must agree to make their submissions (system output, system description, ...) available for experimental use by the organizers.

4.3 System Descriptions

Participants must send a PDF file with the description of each submitted system (primary and contrastive. The format of the submitted documents must fulfill the requirements of the Iberspeech conference. You can use the templates provided for the conference (WORD or LATEX). Please, include in your descriptions all the essential information to allow readers to understand the key aspects of your systems.

4.4 Schedule

- Registrations open: January 1st, 2020
- Development data available and BEAT platform open: March 1st, 2020
- Registration deadline April 30th, 2020
- Deadline for final system submission May 31, 2020
- Results released to the participants July 1st, 2020
- system description due by July 31st, 2020
- workshop: November 2020 at Iberspeech

5 Acknowledgments

This evaluation is funded by the CHIST-ERA project ALLIES (ARN-17-CHR2-0004-01) `https://projets-lium.univ-lemans.fr/allies/`