Computer-assisted Speaker Diarization: How to Evaluate Human Corrections

Broux Pierre-Alexandre[1,2], **Doukhan David**[2], **Petitrenaud Simon**[1], **Meignier Sylvain**[1], **Carrive Jean**[2]
1: Computer Science Laboratory Le Mans University (LIUM - EA 4023), Le Mans, France
2: French National Audiovisual Institute (INA), Paris, France
pabroux@ina.fr, ddoukhan@ina.fr, simon.petit-renaud@univ-lemans.fr, sylvain.meignier@univ-lemans.fr, jcarrive@ina.fr

Abstract

In this paper, we present a framework to evaluate the human corrections of a speaker diarization. We propose four elementary actions to correct the diarization and an automaton to simulate the correction sequence. A metric is described to evaluate the correction cost. The framework is evaluated using French broadcast news drawn from the REPERE corpus.

Keywords: Speaker diarization, annotation, Human-Computer Interaction (HCI), evaluation

1. Introduction

The work presented in this paper has been realized to cope with some needs of the French National Audiovisual Institute (INA). INA is a public institution in charge of the preservation and the promotion of French audiovisual heritage. The promotion task partly relies on the annotation of audiovisual document collections. The annotation consists in enriching the documents with summaries, keywords or participant names in order to satisfy the complex queries elaborated by INA customers or researchers within media databases.

However, due to the increasing number of documents and the limited number of annotators, many documents remain undocumented or only partly documented. The information provided by the annotation greatly varies according to the kind of archives: the broadcast news is usually finely annotated, while the other programs such as games, documentaries, variety shows or reality shows are much less annotated. Thus, enterprises owning large undocumented or partly documented collections such as INA need to exploit their resources even better. One of the solutions to facilitate the annotation and improve the access to its documents is to use automatic speech and speaker recognition technologies as proposed in Charhad et al. (2005; Ordelman et al. (2009; Vallet et al. (2016).

The speaker diarization task is a necessary pre-processing step for speaker identification (Bonastre et al., 2000) or speech transcription (Anguera et al., 2012) in broadcast shows. The speaker diarization and speaker identification tasks allow to determine « who spoke when ». Speaker diarization systems are generally based on unsupervised segmentation and clustering methods, in charge of estimating the number of speakers, and splitting the audio stream into labeled speech segments assigned to anonymous speakers. However, state-of-the-art speaker diarization systems are still not sufficiently accurate to be employed into most of INA's applications, mainly because of the wide variety of INA's collections. The variety relates to the time period (from the end of the nineteenth century to nowadays), the type of broadcast or the recording conditions. For these various reasons, human interventions are most of the time required to obtain robust annotations.

Entirely manual annotation of speech cannot be a reasonable solution as it is a very expensive process. Indeed, nine hours are required to perform the manual annotation corresponding to one hour of spontaneous speech (speech transcription and speaker identity) (Bazillon et al., 2008). Thus, a human annotator should be assisted by an automatic system to be efficient.

In this paper, we propose a framework to experiment human assisted diarization methods. More precisely, the aims are to build an automaton which simulates the annotator corrections and to propose a metric to evaluate these corrections.

In this paper, firstly, we present the state of the art in the field of annotation in speech or speaker recognition systems. Then, we propose an overview of a human assisted diarization system and we propose a new metric to evaluate such systems. In the following part, we describe the human actions used to correct the diarization. Before concluding, we measure the duration of each action to build the proposed metric and we evaluate an oracle system based on the automaton.

2. Related work

The human annotation of an audio document is time consuming. This task is generally manually realized with annotation software like *Transcriber* (Barras et al., 2001) or *ELAN* (Wittenburg et al., 2006). In Bazillon et al. (2008) the authors have shown that the output correction of an automatic speech transcription system decreases the time devoted to the annotation process. An active learning method, proposed in Budnik et al. (2014), used in conjunction with various systems, for example with a speaker diarization system and a face-recognition system, further reduces the number of human-machine interactions. The authors proposed to apply their method to the output of a multilayer perceptron (ML) classifier, based on lip activity and other temporal characteristics. This classifier was used on both speaker and face tracks extracted from videos so as to find associations between them and create multimodal clusters. These clusters were initially labeled thanks to an optical character recognition (OCR). Recently, in Broux et al. (2016), we proposed a system assisting an annotator for the correction of a diarization system reducing the number of human interventions. In that paper, the annotator only corrects speaker clustering errors and the segmentation was assumed to be perfect. More precisely, the input of our system was a segmentation obtained from the ground truth and a hierarchical agglomerative clustering was applied to this segmentation. These last two papers are focused on the correction of clustering errors and forget the segmentation errors. Moreover, the authors unfoundedly assumed that every kind of correction has the same cost. This assumption is not judicious, since each correction requires specific actions from the annotator. These actions require different mental efforts, a different physical effort and provides a different result. For example, it can be assumed that it is easier to change the speaker label (with the use of an exhaustive list of potential speakers) than to create a new speaker label, since it is not provided and the annotator may need much time to find it.

3. Human assisted diarization system

In this section, we propose an overview of a human assisted diarization system as well as a new metric to evaluate such systems.

3.1. Description of the system

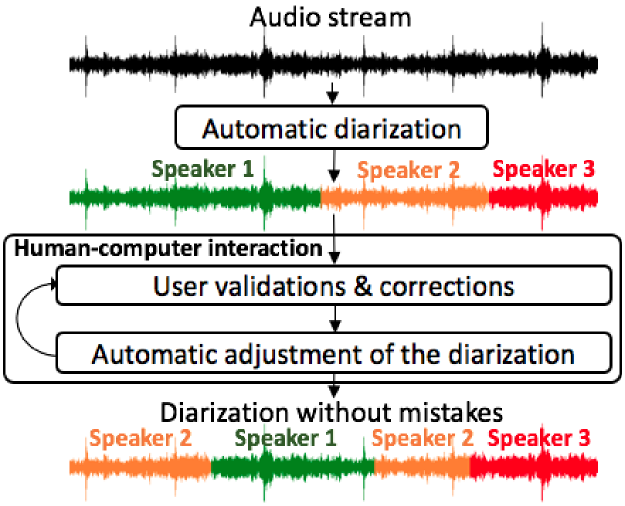

Audio stream
Automatic diarization
Speaker 1
Speaker 2
Speaker 3
Human-computer interaction
User validations & corrections
Automatic adjustment of the diarization
Diarization without mistakes
Speaker 2
Speaker 1
Speaker 2
Speaker 3

Figure 1: Architecture of a diarization system assisted by the human

Figure 1 presents the architecture of a diarization system assisted by a human. It is composed of two main parts. The first one consists in providing an automatic diarization from an audio stream. An initial segmentation of the stream is then obtained. The second one consists of asking a human to correct the output of the first part. Each human correction is in turn taken into account by a system which improves the diarization, generally making easier the remaining actions of the annotator. According to the target, the annotator can achieve corrections for the clustering task and/or the segmentation task. At the end of the process, the diarization error rate (DER (NIST, 2003; Galibert, 2013)) is expected to have decreased thanks to human and system corrections. It is recalled that the DER is the fraction of speaking time which is not attributed to the correct speaker by using the best matching between speaker labels of the references and the hypotheses.

3.2. Experimental framework

From the framework presented in the preceding section, several rules have been defined:

1. the annotator is simulated by an automaton and does not make any error (it is a noiseless automaton);
2. the annotator corrects the show from the beginning to the end in temporal order so as to validate the automatic annotation done a posteriori;
3. only the current speech turn can be corrected by the annotator.

The noiseless automaton rule allows to avoid the complex random modeling of the error according to the annotator. Moreover, this natural simplification allows to have a document without any error at the end of the correction process and thus a DER equals to 0%.Correcting from the beginning to the end is supposed to help the annotator understanding and improving the correction. This rule and the last one are experimental conditions chosen to facilitate our problem. They can be questioned thereafter.

3.3. Proposed metric: HCIQ

Since the DER measures the quality of a diarization (NIST, 2003), it is not relevant to evaluate human interactions. A new metric similar to the Keystroke Saving used in word prediction for people with communication difficulty (Wood and Lewis, 1996) is proposed. We called it Human-Computer Interaction Quantity (HCIQ). This metric estimates the human intervention cost for the diarization correction. The HCIQ can be computed both for assisted systems and systems where a human corrects the diarization alone. Furthermore, as well as the DER, the HCIQ can be computed for each recording or for a set of audio/video recordings. It is defined by the formula:

$$HCIQ = \sum_{i=1}^{K} w_i n_i,$$

where the index i corresponds to a correction action type in the interface, w_i is its associated cost, n_i the number of times the annotator has applied this action type and K is the number of different action types.
The lower the HCIQ measure, the lower the correction duration of the annotation will be. By the way, it allows to compare different assisted diarization systems in a objective way.
The HCIQ measure, in its current form, does not allow to compare different corpora. In order to resolve it, we propose the following formula:

$$HCIQ_n = \frac{HCIQ}{d},$$

where $HCIQ$ is normalized by the corpus duration d on which the $HCIQ$ was assessed. The HCIQ$_n$ measure is a ratio of the number of corrections to do for a unit of time. For a given corpus, when HCIQ$_n$ increases, the amount of corrections increases also.

The HCIQ is close to others metrics that assess the amount of effort needed to correct a given kind of errors. For instance, the word error rate (WER (McCowan et al., 2004)) estimates the number of incorrect words in a transcription and the translation edit rate (TER (Snover et al., 2006)), which is derived from the WER, assesses the number of corrections needed for a human to reach the reference translation.

4. Annotator and assisted diarization tools

In this section, we describe the annotation software and the authorized correction actions which we used to correct a diarization.

4.1. Annotation software: Transcriber

To choose the human actions needed to the correction, we relied on *Transcriber*, a reference software in speech transcription and annotation.

This software allows to cut an audio stream into segments. Each segment corresponds to a speech zone and is labeled with a speaker name. This label may be enriched by information such as the gender or the native language of the speaker.

In *Transcriber*, the segmentation actions are "*Create a boundary*", "*Delete a boundary*" and "*Move a boundary*". The "*Create a boundary*" action adds a boundary by cutting a segment into two pieces, the "*Delete a boundary*" action merges two consecutive segments and the "*Move a boundary*" action moves the boundary of a segment. Concerning the clustering actions, *Transcriber* offers the "*Create a speaker label*" and "*Change the speaker label*" actions. The former allows to create a new speaker label for the current segment whereas the latter allows to change the speaker label.

4.2. Correction actions

In order to facilitate the creation of a simulated annotator, the series of actions will be deterministic. No action can be substituted by a set of actions providing the same correction. One of the *Transcriber* actions does not fulfill this criterion. The "*Move a boundary*" action can be replaced by the two following actions: "*Create a boundary*" and "*Delete a boundary*".

To sum up, we kept two actions for modifying the segment boundaries and two actions for modifying the labels. By combining these actions, we can describe all the corrections in a unique way. Finally, the four selected actions used in the HCIQ metric are :

- "*Create a boundary*",
- "*Delete a boundary*",
- "*Create a speaker label*",
- "*Change the speaker label*".

5. Experiments

In this section, we present the corpus used for our experiments, the measure of the action duration in order to build the proposed metric and the evaluation of an oracle system based on the automaton.

5.1. Corpus

The experiments have been applied on TV recordings from the 2013 evaluation campaign of the challenge ANR-REPERE[1]. The TV shows come from two French channels (BFM and LCP). They are mainly composed of talk shows and new broadcasts.

Show number	7
Recording number	28
Recording time	14h17
Annotation time	2h57
Speaker number	212

Table 1: REPERE test 2013 description

Table 1 describes the corpus used in the experiments. The corpus is balanced : it contains spontaneous and prepared speech. It is made up of street interviews, debates and information shows but only a part of the data is annotated (Kahn et al., 2012).

5.2. Measure of the action duration

According to Arora et al. (2009), three variables can directly affect the assessment of the action duration: the interface, the annotator and the annotated document. So as to obtain an accurate assessment, each of these variables is studied.

5.2.1. The interface

The assessment of action duration varies according to the used interface. The more ergonomic it is, the more a user quickly annotates and the more the annotation time decreases. *Transcriber* offers good ergonomics. In *Transcriber*, a human generally cuts the audio stream by putting boundaries on speaker changes, silences or breaths. In our framework, only speaker changes are useful and are annotated.

5.2.2. The annotator

The annotation time varies according to the experience the annotator can have both in annotation itself and in the annotation software. If a person is used to annotate, he or she will effective and the annotation will take little time. Moreover, people frequently using an annotation software correct more quickly than people discovering it. Therefore, there are two possible strategies to measure the action duration:

1. using the average of time of annotators having the same experience;
2. using the average of time of annotators having several experiences.

[1] http://www.defi-repere.fr/

The former one allows to obtain specific time which can be useful in accordance with the aim. The latter one, that we chose in our experiments, offers global time covering different annotator profiles with variable expertise levels. Unfortunately, this time is constrained by the annotator profiles which we have at our disposal.

5.2.3. The annotated document

As far as speech is concerned, the audio documents can mainly be separated in three groups: telephone, meeting and (radio/TV) show. These three groups essentially varies according to two points: the audio stream quality and the spontaneous degree. The spontaneous degree is correlated with the disfluency number. When the degree is high, the disfluency number is also high and the clauses are more ungrammatical. The spontaneous degree implies various phenomena such as overlapped speech, false starts, etc. (Bazillon et al., 2008; Dufour et al., 2009). When the signal quality is low, it is more difficult to annotate. Furthermore, when the spontaneous degree is high due to the overlapped speech, it is more complicated to determine who speaks when, and then it is more difficult to annotate. The signal quality and the spontaneous degree have consequences on the human annotation time (Bazillon et al., 2008). Therefore, two strategies can also be conceivable:

1. annotating various documents;
2. annotating specific documents.

We choose the latter strategy. Thus the documents to annotate are good audio quality with a low spontaneous degree. This choice allows to facilitate the measure of annotation time.

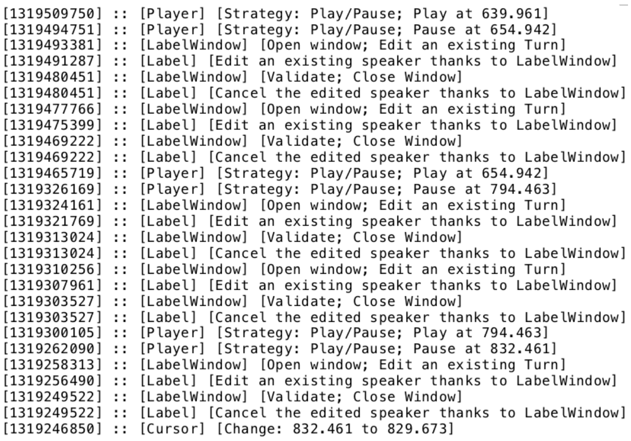
5.2.4. Evaluation

In section 4.2., we selected four correction actions. Now, we propose a method to estimate the average duration of each action, in the framework we defined previously. The history of mouse clicks and keyboard strokes in *Transcriber* permits to indirectly determine the successive actions and to precisely assess the duration of each action. To record this detailed log file, we had to modify the *Transcriber* code. A log file input, i.e. a mouse click or a keyboard stroke, contains three types of information : the click or the stroke time, the active module name and a comment (figure 2). The module identifies an element of the user interface whereas the comment gives accurate details on the event in progress.

The log file itself is not enough to determine the actions in an automatic way. Indeed, the annotator can make some mistakes or take a break during the annotation session. To solve this problem, the recording of the user screen is manually segmented into one or several actions thanks to the log file. Thus the measured duration accurately correspond to the actual actions.

In accordance with 5.2.3., only the regions with few spontaneous speech and without overlapping speech are annotated to assess the annotation time for each action. Table 2 shows the results of the duration of actions.

The most time consuming actions are the ones consisting to "*Create a speaker label*" and "*Create a boundary*", with an average of about 12-13 seconds.The first action requires to enter a speaker label (and possibly other speaker meta data), while the second action requires to look and listen to the signal to detect the speaker boundary. Moreover, it is generally necessary to listen to the signal several times to place a new boundary. The action called "*Change the speaker label*" has an average of 7.6 seconds. Thanks to a contextual window, it consists in selecting the correct speaker label in a drop-down list. Looking for a label in a drop-down list takes a less mental effort compared to the boundary creation. The fastest action is the "*Delete a boundary*" action. It requires to stop listening when a false boundary is detected and to delete it by a simple keyboard key combination. Correcting an error is in reality built upon three phases:



```
[1319509750] :: [Player] [Strategy: Play/Pause; Play at 639.961]
[1319494751] :: [Player] [Strategy: Play/Pause; Pause at 654.942]
[1319493381] :: [LabelWindow] [Open window; Edit an existing Turn]
[1319491287] :: [Label] [Edit an existing speaker thanks to LabelWindow]
[1319480451] :: [LabelWindow] [Validate; Close Window]
[1319480451] :: [Label] [Cancel the edited speaker thanks to LabelWindow]
[1319477766] :: [LabelWindow] [Open window; Edit an existing Turn]
[1319475399] :: [Label] [Edit an existing speaker thanks to LabelWindow]
[1319469222] :: [LabelWindow] [Validate; Close Window]
[1319469222] :: [Label] [Cancel the edited speaker thanks to LabelWindow]
[1319465719] :: [Player] [Strategy: Play/Pause; Play at 654.942]
[1319326169] :: [Player] [Strategy: Play/Pause; Pause at 794.463]
[1319324161] :: [LabelWindow] [Open window; Edit an existing Turn]
[1319321769] :: [Label] [Edit an existing speaker thanks to LabelWindow]
[1319313024] :: [LabelWindow] [Validate; Close Window]
[1319313024] :: [Label] [Cancel the edited speaker thanks to LabelWindow]
[1319310256] :: [LabelWindow] [Open window; Edit an existing Turn]
[1319307961] :: [Label] [Edit an existing speaker thanks to LabelWindow]
[1319303527] :: [LabelWindow] [Validate; Close Window]
[1319303527] :: [Label] [Cancel the edited speaker thanks to LabelWindow]
[1319300105] :: [Player] [Strategy: Play/Pause; Play at 794.463]
[1319262090] :: [Player] [Strategy: Play/Pause; Pause at 832.461]
[1319258313] :: [LabelWindow] [Open window; Edit an existing Turn]
[1319256490] :: [Label] [Edit an existing speaker thanks to LabelWindow]
[1319249522] :: [LabelWindow] [Validate; Close Window]
[1319249522] :: [Label] [Cancel the edited speaker thanks to LabelWindow]
[1319246850] :: [Cursor] [Change: 832.461 to 829.673]
```

Figure 2: Example of a log file

Action	**Nb**	**Avg (sec)**	**Std (sec)**
Create a speaker label	28	12.7	6.0
Change the speaker label	32	7.6	3.8
Create a boundary	38	12	7.6
Delete a boundary	46	5.1	2.3

Table 2: Duration of actions - 20 min of REPERE test 2013 data. Nb: Number of occurrences; Avg: Average duration; Std: Standard deviation.

1. detecting the presence of an error;
2. finding the place of the error;
3. correcting the error.

Each action duration in table 2 represents the sum of the time of these three phases. However, these durations do not take into account the listening time.

5.3. Evaluation of an oracle system

The simulated annotator relies on two types of information to determine whether a correction is required at time t:

- the correspondence between the reference segment (ground truth) and the hypothesis segment;
- the matching between the reference and the hypothesis speaker labels which minimize the DER.

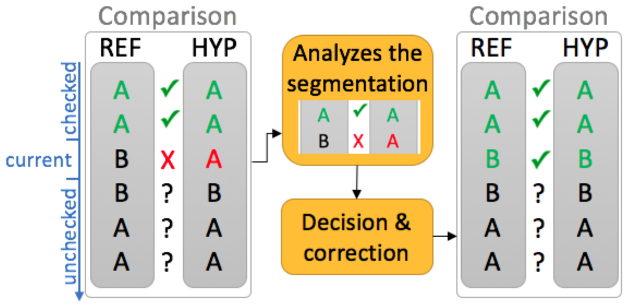


Figure 3: Illustration of the simulated annotator

If there is at least one discordance in the diarization at time t between the reference and the hypothesis, the simulated annotator firstly corrects the segmentation errors and then the speaker clustering errors (figure 3). After each correction, the system can run a diarization system on the unchecked part (segments with start time $> t$) by taking into consideration the already checked segments (segments with end time $\leq t$).

Without segmentation errors, the clustering correction is easy to set up (Broux et al., 2016). The segmentation correction is more difficult, the simulated annotator needs to deal with the accuracy of the reference boundaries. To solve this problem, a tolerance of more or less 250 ms is generally applied to the boundaries of the reference segments for the DER computation. We applied the same tolerance to avoid the numerous and generally useless corrections. So before the assessment of the potential discordance between the reference and the hypothesis, any hypothesis boundary belonging to a tolerance area is moved in order to be aligned with the reference boundary.

The simulated annotator becomes an oracle system when no automatic adjustment is performed as corrections. The oracle system evaluation is reported in table 3. The HCIQ of the test corpus is 331.6 minutes and corresponds to the sum of all duration estimations (table 3). The input diarization of the oracle is provided by the full-automatic diarization system described in Meignier and Merlin (2010). The DER of the input diarization is 13.80%.

Action	**Nb**	**D (min)**
Create a speaker label	295	62.4
Change the speaker label	463	58.7
Create a boundary	986	197.2
Delete a boundary	156	13.3

Table 3: Correction for the oracle system - REPERE test 2013. Nb: Number of occurrences; D: duration estimation (Avg duration × number of occ.)

The occurrence number of the segmentation actions is about one and a half greater than the clustering action number (respectively 1142 and 758). Segmentation corrections represent about 65% of the total correction time (210.5 minutes). The "*Create a boundary*" action is the most costly action, since it corresponds to about 52% of the overall corrections. For an audio recording of 2h57 (177 min), an annotator will take 3h17 (197.2 min) to create boundaries.

If the simulated annotator only corrects the clustering errors, the DER is 5.59% at the end of the correction process. These 5.59% errors are due to the wrong segmentation. This result shows that segmentation errors and clustering errors approximately contribute to 40% and 60% respectively of the DER. Comparatively, the segmentation errors correspond to the main correction cost in terms of HCIQ.

Corpus	**HCIQ (min)**	**AT (min)**	**HCIQ$_n$**
ESTER test 2003	477.2	592	0.81
ESTER test 2009	482.0	430	1.12
ETAPE test 2012	793.7	418	1.90
REPERE test 2013	331.6	177	1.87

Table 4: Comparison of HCIQ$_n$ obtained from corrections of the oracle system on several corpora. AT: annotation time

Table 4 compares the HCIQ$_n$ of the REPERE test 2013 corpus to others. It shows that the REPERE corpus is one of the corpora which requires the most corrections for a unit of time since it needs on average 1.87 minutes of human corrections for 1 minute of audio signal. Moreover, it demonstrates that the ETAPE and REPERE corpora, being mainly more composed of spontaneous speech (false starts, repetition, overlapped speech, interjections, etc (Bazillon et al., 2008)) than the ESTER corpora, are enhanced with high HCIQ$_n$.

6. Conclusions

In this paper, we proposed a framework to assess any interactive system of diarization taking into account the human corrections. The combination of four actions permits to describe the correction steps in a single way. We proposed a metric used to precisely determine the duration of each action in order to assess the cost of the human-computer interactions. The evaluation of an oracle system on REPERE test 2013 shows that segmentation corrections take longer than the clustering corrections. The results of the oracle demonstrates the importance of segmentation errors on the HCIQ and the DER as well. The correction of segmentation errors increases the HCIQ measure whereas it affects the DER in a negligible way. Only the correction of clustering errors directly decreases the DER measure. Future work will be focus on the development of an embedded speaker diarization system to reduce the correction time. Then, a study will be done to determine how to call a human with parsimony (i.e. in some parts of the document) by having a sufficient annotation of the document for a targeted application. The input diarization system will be modified as well, in particular the segmentation step.

7. Bibliographical References

Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. (2012). Speaker diarization: A review of recent research. ieee-tsap, 20(2):356–370, Feb.

Arora, S., Nyberg, E., and Rosé, C. P. (2009). Estimating annotation cost for active learning in a multi-annotator environment. In Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing, pages 18–26. Association for Computational Linguistics.

Barras, C., Geoffrois, E., Wu, Z., and Liberman, M. (2001). Transcriber: development and use of a tool for assisting speech corpora production. Speech Communication, 33(1):5–22.

Bazillon, T., Estève, Y., and Luzzati, D. (2008). Transcription manuelle vs assistée de la parole préparé et spontanée. Revue TAL.

Bonastre, J.-F., Delacourt, P., Fredouille, C., Merlin, T., and Wellekens, C. (2000). A speaker tracking system based on speaker turn detection for nist evaluation. In Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on, volume 2, pages II1177–II1180. IEEE.

Broux, P.-A., Doukhan, D., Petitrenaud, S., Meignier, S., and Carrive, J. (2016). An active learning method for speaker identity annotation in audio recordings. In 1st International Workshop on Multimodal Media Data Analytics (MMDA), In conjuction with the 22nd European Conference on Artificial Intelligence (ECAI).

Budnik, M., Poignant, J., Besacier, L., and Quénot, G. (2014). Automatic propagation of manual annotations for multimodal person identification in tv shows. In Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on, pages 1–4. IEEE.

Charhad, M., Moraru, D., Ayache, S., and Quénot, G. (2005). Speaker identity indexing in audio-visual documents. In Content-Based Multimedia Indexing (CBMI2005).

Dufour, R., Jousse, V., Estève, Y., Béchet, F., and Linarès, G. (2009). Spontaneous speech characterization and detection in large audio database. SPECOM, St. Petersburg.

Galibert, O. (2013). Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech. In INTERSPEECH, pages 1131–1134.

Kahn, J., Galibert, O., Quintard, L., Carré, M., Giraudel, A., and Joly, P. (2012). A presentation of the repere challenge. In Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on, pages 1–6. IEEE.

McCowan, I. A., Moore, D., Dines, J., Gatica-Perez, D., Flynn, M., Wellner, P., and Bourlard, H. (2004). On the use of information retrieval measures for speech recognition evaluation. Technical report, IDIAP.

Meignier, S. and Merlin, T. (2010). Lium spkdiarization: an open source toolkit for diarization. In CMU SPUD Workshop, volume 2010.

NIST. (2003). The rich transcription spring 2003 (RT-03S) evaluation plan. http://www.itl.nist.gov/iad/mig/tests/rt/2003-spring/docs/rt03-spring-eval-plan-v4.pdf, February.

Ordelman, R., De Jong, F., and Larson, M. (2009). Enhanced multimedia content access and exploitation using semantic speech retrieval. In Semantic Computing, 2009. ICSC'09. IEEE International Conference on, pages 521–528. IEEE.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In Proceedings of association for machine translation in the Americas, volume 200.

Vallet, F., Uro, J., Andriamakaoly, J., Nabi, H., Derval, M., and Carrive, J. (2016). Speech trax: A bottom to the top approach for speaker tracking and indexing in an archiving context. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC). European Language Resources Association (ELRA).

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., and Sloetjes, H. (2006). Elan: a professional framework for multimodality research. In Proceedings of LREC, volume 2006, page 5th.

Wood, M. E. and Lewis, E. (1996). Windmill-the use of a parsing algorithm to produce predictions for disabled persons. PROCEEDINGS-INSTITUTE OF ACOUSTICS, 18:315–322.