Is Incremental Cross-Show Speaker Diarization Efficient For Processing Large Volumes of Data?

Grégor Dupuy, Sylvain Meignier, Yannick Estève

Is Incremental Cross-Show Speaker Diarization Efficient For Processing Large Volumes of Data?

Grégor Dupuy, Sylvain Meignier, Yannick Estève

LIUM, University of Le Mans, France
first.lastname@lium.univ-lemans.fr

Abstract

Current cross-show diarization systems are mainly based on an overall clustering process which handles all the shows within a collection simultaneously. This approach has already been studied in various situations and seems to be the best way so far to achieve low error rates. However, this process has limits in realistic applicative contexts where large and dynamically increasing collections have to be processed. In this paper we investigate the use of an incremental clustering cross-show speaker diarization architecture to iteratively process new shows within an existing collection. The new shows to be inserted are processed one after another, according to the chronological order of their broadcasting dates. Experiments were conducted on the data distributed for the ETAPE and the REPERE French evaluation campaigns. It consist of 142 hours of data collected from 310 shows, from a period from Sept. 2010 to Oct. 2012.

Index Terms: speaker diarization, incremental architecture, cross-show, ILP clustering, i-vectors

1. Introduction

Cross-show speaker diarization aims to extend the task, as defined by the NIST, to a broader context. The purpose is to find speaker utterances within a collection of data by partitioning the input audio streams into segments, and by clustering those segments according to the speaker identities. Thus, a cross-show speaker would always be labeled in the same way, in every recording that composes a collection. Contrastingly, speaker diarization that does not focus on the relationship between the speakers involved in several shows is referred to as *single-show* speaker diarization. The cross-show speaker diarization task shares the same objective as the speaker linking task [1][2][3], *i.e.*, to find speaker utterances across several audio recordings. However, speaker linking is a process performed a posteriori, using the single-show speaker diarization segmentations, contrary to the cross-show speaker diarization where the process is performed directly during the last clustering step.

Previous studies have already been made to find effective ways to perform cross-show speaker diarization. [4][5] presented two main architectures designed to process the shows within a collection, using state-of-the-art speaker diarization approaches. The *overall clustering* architecture proposed in previous works considers all the recordings of a collection simultaneously. The alternative method, the *incremental clustering*, has been designed to handle recordings iteratively, avoiding the combinatorial explosion problem which may occur with the overall clustering method. However, problems arise when the collection to be processed is too large. Approaches relying on Hierarchical Agglomerative Clustering (HAC), along with Gaussian Mixture Models (GMMs), are inefficient because of the algorithm complexity. We proposed a global optimization framework [6] where the speaker clusters are modeled by i-vectors, which have become the state-of-the-art in the speaker verification field. The i-vector approach was first adapted to speaker diarization using the *k-means* algorithm, applied to distances between i-vectors, to find utterances of speakers within a corpus where the number of speakers is known *a priori* [7]. In our approach, the clustering problem is expressed as an Integer Linear Programming (ILP) problem, in which all of the clusters are considered simultaneously. This ILP clustering approach is well-suited for processing collections [8]. It achieved the best or second best results in the speaker diarization task at the latest French broadcast news evaluation campaigns [9][10]. Recently, [11] proposed to perform speaker clustering on content graphs built from large audio corpora. We developed a similar approach to improve the ILP clustering [12], where the matrix associating distances between speaker clusters is interpreted as a connected graph. Its decomposition into connected components allows the system to decompose the clustering problem into several sub-problems. Most of these sub-problems are trivial and do not need to be processed with a clustering algorithm.

In this paper we investigate one of the current issues with handling collections: how to deal with large collections which may dynamically increase over the time. We present an incremental cross-show speaker diarization system, based on i-vectors and improved ILP clustering (including the graph clustering approach). The experiments were carried out in a realistic applicative situation, where new shows are inserted one after another into an existing collection, following the broadcasting dates of the shows. This paper is organized as follows: in section 2, we describe our speaker diarization system, and we present the incremental architecture we experimented with in section 3. Then, we present the data, the evaluation metric, and the experimental results in section 4. We finally give a conclusion in section 5.

2. Speaker diarization architecture

The *LIUM_SpkDiarization* toolkit[1] [13] was used to build the diarization system. This system has achieved the best or second best results in the speaker diarization task at French broadcast news evaluation campaigns such as ESTER2, ETAPE (2011) [9] and REPERE (January 2012, 2013 and 2014) [10].

2.1. Single-show diarization

The single-show diarization system is based on an acoustic segmentation and a Hierarchical Agglomerative Clustering using the Bayesian Information Criterion (BIC), both as similar-

[1] http://www-lium.univ-lemans.fr/en/content/liumspkdiarization

ity measure between clusters (speakers), and as stop criterion for the merging process. In this clustering, speakers are modeled with full-covariance matrix Gaussian distributions. Segment boundaries are adjusted through a Viterbi decoding using 8-component GMMs, learned on the data of each speaker via the Expectation-Maximization (EM) algorithm. Another Viterbi decoding, with 8 one-state HMMs represented by 64-component GMMs, trained by EM on ESTER1 train data [14], is carried out to remove non-speech areas. Gender (male / female) and bandwidth (narrow / wide band) detection is performed using 4×128 diagonal component GMMs trained on 1 hour of speech from the ESTER1 training corpus (there is 1 GMM for each of the gender-bandwidth combinations). Segmentation, clustering, and decoding are performed using 12 MFCC parameters, supplemented with energy.

At this point, each cluster is supposed to represent a single speaker; however, several clusters can be related to the same speaker. A final clustering stage, expressed as an Integer Linear Programming problem, is thus performed.

2.1.1. *ILP clustering*

In this clustering approach, the speaker clusters are modeled with i-vectors, and the similarity between i-vectors is estimated with a *Mahalanobis* distance [6]. The i-vector approach has become the state-of-the-art in the Speaker Verification research field [15]. Inspired by the Joint Factor Analysis approach, the i-vectors reduce acoustic data of a speaker into a low-dimension vector that preserves the specific information about that speaker. An i-vector is extracted from the projection of a speech utterance in a low-dimensional subspace, learned from a factor analysis model on a large collection of data. The i-vector approach was first adapted to speaker diarization using the *k-means* algorithm, applied to distances between i-vectors, to find utterances of speakers within a corpus where the number of speakers is known *a priori* [7]. Here, the number of speakers is unknown. According to the BIC segmentation, a 50-dimensional i-vector is extracted from each speaker cluster along with a 256 GMM-UBM. The extracted i-vectors are then length-normalized in an iterative process [16][17]. Acoustic feature extraction is performed using 12 MFCC parameters, supplemented with energy, first and second order derivatives. The features are normalized with mean and variance.

The clustering problem consists in jointly minimizing the number C of cluster centers chosen among the N i-vectors, as well as the dispersion of i-vectors within each cluster. The set $C \in \{1, \ldots, N\}$ is to be automatically determined. The objective solving function of the ILP problem (eq. 1) is minimized subject to constraints:

let $C \in \{1 \ldots N\}$, let $K_{j \in C} = \{k / d(k,j) < \delta\}$

$$\text{Minimize:} \quad \sum_{k \in C} x_{k,k} + \frac{1}{N\delta} \sum_{j \in C} \sum_{k \in K_j} d(k,j) x_{k,j} \qquad (1)$$

$$\text{Subject to:} \quad x_{k,j} \in \{0,1\} \qquad k \in K_j, j \in C \qquad (1.2)$$

$$\sum_{k \in K_j} x_{k,j} = 1 \qquad j \in C \qquad (1.3)$$

$$x_{k,j} - x_{k,k} < 0 \qquad k \in K_j, j \in C \qquad (1.4)$$

where $x_{k,k}$ (eq. 1) is a binary variable equal to 1 when the i-vector k is a center. The number of centers C is implicitly included in equation 1, indeed $C = \sum_{k=1}^{N} x_{k,k}$. The distance $d(k,j)$ is computed using the *Mahalanobis* distance between i-vectors k and j. $1/N\delta$ is a normalization factor. The binary variable $x_{k,j}$ is equal to 1 when the i-vector j is assigned to the center k. Each i-vector j will be associated with a single center k (eq. 1.3). Equation 1.4 ensures that the cluster k is selected if an i-vector is assigned to cluster k. Distances are implicitly taken into account in eq. 1.2, eq. 1.3 and eq. 1.4, by using the set K_j in place of C. Given a value j, the set K_j represents the set of possible values of k (taken between 1 and N) for which distances between clusters k and j are shorter than the threshold δ.

The GMM-UBM was learned on the test corpus provided during the ESTER1 French evaluation campaign [18], and its training corpus was used to train the i-vectors required in the normalization step. The ILP problem is solved by the *glpsol* solver included in the *GNU Linear Programming Toolkit*[2].

2.1.2. *Pre-processing with connected graph sub-components*

The matrix associating distances between speaker clusters can be interpreted as a connected graph, illustrated in Figure 1, where the clusters are represented by the nodes, and distances between clusters are represented by the edges.

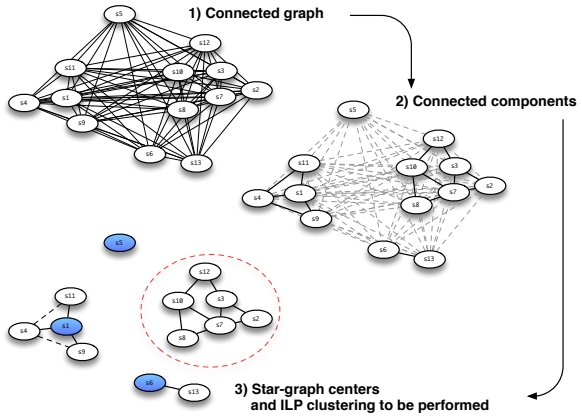

Figure 1: Pre-processing with connected graph sub-components. The dashed circle indicates an ILP clustering to be performed; the colored clusters are identified as star-graph centers.

This graph can be simplified using the connectivity concept of graph theory, by removing all the unnecessary edges corresponding to distances longer than the threshold δ. This simplification transforms the completely connected graph into a set of connected components (the subgraphs). Connected components can be easily found by using the depth-first search algorithm iteratively. Because the resulting subgraphs are no longer connected, they constitute independent sub-problems which can be processed separately (middle scheme of Figure 1).

However, most of the cluster centers are obvious. The search for cluster centers in the connected components can be formulated as the search for star-graphs. A star-graph is a special kind of tree composed of one central node attached to k leaves (a single depth level graph). Therefore, for each of the connected components found using the depth-first search algorithm, we can determine if it displays the star-graph characteristics in order to determine its cluster center. There is no need to use a clustering algorithm on a star-graph because the central

[2] http://www.gnu.org/software/glpk/

node corresponds to the center of the sub-clustering problem, so clusters corresponding to the leaves will be directly associated to the cluster center. If no *star* can be found, then the connected components have to be processed with a clustering algorithm in order to find at least two cluster centers (bottom scheme of Figure 1). We formulated the clustering problem with the ILP, as described in section 2.1.1. The ILP clustering stage is very fast since the number of variables and constraints is reduced [12].

2.2. Cross-show diarization

Cross-show speaker diarization simply consists of an ILP clustering with subgraph pre-processing, performed in exactly the same way as described in the previous section. However, we changed the configuration: the 256 GMM-UBM was replaced by a 1024 GMM-UBM, and the acoustic features extraction is performed using 20 MFCCs (including C_0), with their first and second order derivatives. The features remain normalized with mean and variance. The "cross-show" aspect of this ILP clustering relies on the input segmentation, which corresponds to the concatenation of several shows. To ensure that the clusters from the different single-show segmentations are unique, each cluster label is suffixed with the name of its show. What makes the difference between an overall cross-show ILP clustering and an incremental cross-show clustering is the input segmentation. An overall ILP clustering is performed from the concatenation of the single-show segmentations of all the shows in the collection. In the incremental ILP clustering, the single-show segmentations are concatenated iteratively.

ILP clustering is designed to deal with the input segmentations globally: the system is free to merge two clusters from the same show. Indeed, we noticed that the overall ILP clustering often helps slightly improve the single-show segmentations. That particularity explains why the single-show Diarization Error Rate is likely to evolve over the iterations of the incremental architecture we present in Section 4.

3. Incremental cross-show diarization

Previous studies on the incremental clustering were performed on cross-show collections of small size (23 shows for a total duration of about 4 hours) [4][5]. The system proposed by the authors is based on an Open-set Speaker Identification (OSI) module, which compares the speaker models of the show being processed with those of the already processed shows. Before processing another show, the OSI module is updated with the new speaker models, and the existing speaker models are retrained by taking the additional data from the current show into consideration. Inspired by this incremental speaker diarization architecture, as well as the recent cross-show results we obtained with the overall ILP clustering and the subgraph preprocessing [6][8][12], we built an incremental cross-show diarization system based on i-vector speaker models and overall ILP clustering.

Since the order in which shows are processed affects the results [4], we choose to base our experiments on a realistic applicative context (*cf.* Figure 2), where new shows have to be inserted (processed) into an existing collection, one after another, following a chronological order. Considering the chronological order of broadcasting, the n first shows of our data are selected to constitute an initial collection, on which an overall ILP clustering is performed. The purpose of this initial collection is to start the incremental process considering prior knowledge about the speakers that have already appeared in previous shows (like a bootstrap). Aside from the presence of the initial collection, the main difference between our approach and the one proposed in [4] is that we do not rely on an external module (OSI module) to find the cross-show speakers. Each time a show is to be processed, the incremental system performs a complete overall ILP clustering from the concatenation of the single-show segmentation of that show with the current cross-show segmentation (which results from the shows previously processed).

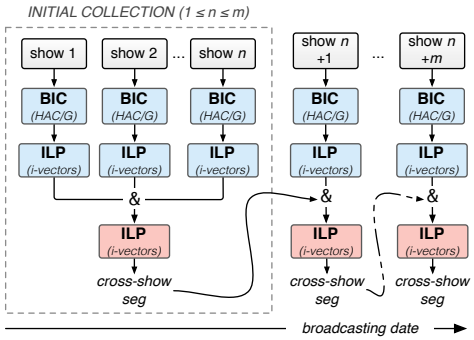

Figure 2: Incremental architecture jointly relying on i-vector speaker models and overall ILP clustering.

Before the incremental process starts, each show of the collection is processed with the single-show diarization system presented in Section 2.1. Then, the initial collection is processed. The single-show segmentations of the n shows that constitute the initial collection are concatenated into a single large file, which serves as input segmentation for an overall ILP clustering, as presented in Section 2.2. When a new show is to be inserted into the collection, its single-show segmentation is concatenated to the current cross-show segmentation (taken from the previous iteration). The i-vectors corresponding to the clusters of the show to be inserted are extracted, and the i-vectors corresponding to the cross-show clusters are retrieved (these are the i-vectors corresponding to the clusters that were identified as connected component centers in the previous iteration). The distance between each pair of i-vectors is then computed, and the overall ILP clustering, including the pre-process with the connected component subgraphs, is performed. The system finally generates a new cross-show segmentation, in which the show to be inserted is taken into account. And so on until the last show.

4. Experiments

4.1. Data

The data for our experiments consists of all audio files from BFMTV and LCP TV channels distributed in the *training*, *development*, and *test* corpora of the ETAPE and REPERE (Jan. 2013) French evaluation campaigns. We also used the training data provided for the Jan. 2014 REPERE evaluation. The data represents a total of 310 files, with 142 hours of audio covering a period from September 2010 to October 2012. Shows were recorded from two French digital terrestrial television channels. They are balanced between prepared and spontaneous speech. The data distribution is presented in Table 1. Only a part of the data are annotated and evaluated: the durations reported in Table 1 (expressed in hours and minutes) were computed according to the Unpartitioned Evaluation Map (UEM) files specifying the regions to score. In order to observe the influence of the bootstrap size during the iterative process, two initial collections of different sizes were formed, referred to as *Boot.1* and

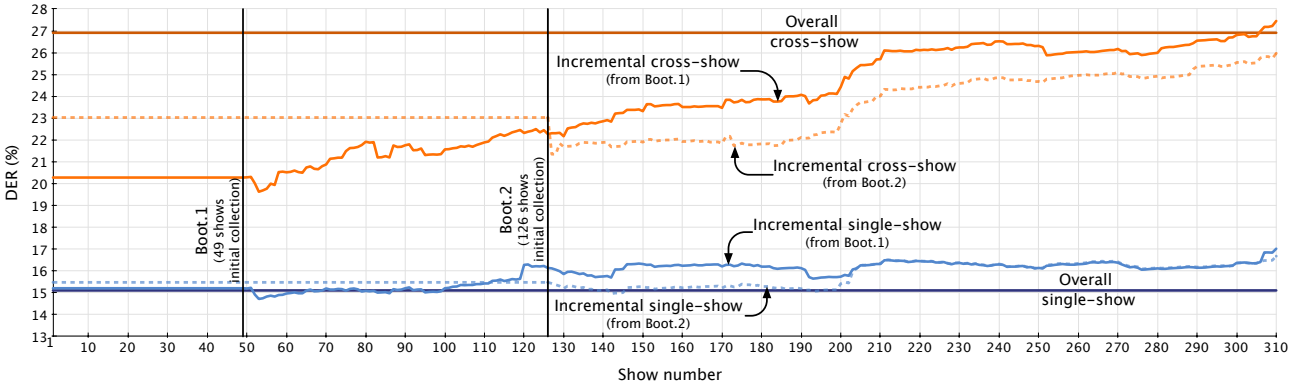

Figure 3: DERs for each iteration of the incremental experiment, with the two bootstrap collections, and for an overall clustering on the whole data set.

Boot.2. The first 49 (resp., 126) files of the data set were selected to constitute the Boot.1 initial collection (resp., Boot.2), and the remaining were iteratively added one after another. The initial collection represents about a third (resp., a half) of the whole data set, in terms of duration, and was used to tune both single- and cross-show system thresholds. Note that we did not use UEM files, nor reference segmentation files, to perform the experiments: the single- and cross-show segmentations files are all automatically produced by our systems.

Show names	Whole data		*Boot.1*		*Boot.2*	
	#S	*Dur.*	*#S*	*Dur.*	*#S*	*Dur.*
BFMStory	37	21:21	7	05:01	19	10:36
CultureEtVous	54	01:42	0	-	0	-
PlaneteShowbiz	73	02:24	1	02:07	35	01:10
CaVousRegarde	20	09:02	9	04:29	12	05:21
EntreLesLignes	24	08:20	10	04:05	16	06:14
LCPInfo	35	08:37	1	00:10	10	01:57
PileEtFace	32	08:15	12	04:07	19	04:52
TopQuestions	35	07:41	9	02:03	15	03:26
Total	310	67:22	49	22:52	126	33:36

Table 1: Number of shows *(#S)* and durations (h:min) for the whole set of data and the two initial collections (*Boot.1* and *Boot.2*).

4.2. Results and discussion

The metric used to measure performance in speaker diarization is the Diarization Error Rate (DER). DER was introduced by the NIST as the fraction of speaking time which is not attributed to the correct speaker, using the best match between references and hypothesis speaker labels. The scoring tool was developed by LNE[3] as part of the ETAPE and the REPERE campaigns. The main difference between this scoring tool and that of the NIST is the speaker hypothesis and reference mapping. The tool from LNE relies on the Hungarian algorithm ($O(n^3)$ polynomial time), whereas the NIST tool uses an algorithm based on heuristics [19].

The single- and cross-show DERs obtained for each iteration of the incremental experiment are presented in Figure 3. The horizontal lines represents the single- and cross-show results obtained with an overall ILP clustering on all 310 recordings that compose the collection. The whole collection is composed of shows recorded in a 2-year period, with some important temporal interruptions between consecutive recordings. The cross-show DER obtained with the overall ILP on all recordings of the collection is high (26.91%), compared to that of the initial collections (20.28% for *Boot.1*, and 23.03% for *Boot.2*). The cross-show DER obtained with the incremental approach, after the last show is processed, is very close to that of the overall clustering (27.45% for *Boot.1*, and 25.97% for *Boot.2*). The single-show evaluation performed on the cross-show segmentations is higher with the incremental approach (17.02% for *Boot.1*, and 16.68% for *Boot.2*), compared to that obtained with the overall ILP on the whole data set (15.11%). One explanation for the drastic increase of DERs between the shows 200 and 210 could be the temporal interruption between the recordings: there were 24 days between the recordings of shows 201 and 202. However, we do not observe similar behavior with the other temporal interruptions, so this could only be a partial explanation of the problem. Removing these shows from the collection gives more linear, and globally better, results. However, these results must be interpreted in a nuanced way. The *Boot.1* initial collection which serves as a starting point for the incremental process represents 34% of the whole data set duration (resp., 50% for *Boot.2*). 55.8% of the speakers present in *Boot.1* are also present in the rest of the data (resp., 58.9% in *Boot.2*), and in addition, 12 speakers are particularly present (most of them are hosts and journalists). The speaking time of these 12 speakers represents about 30.6% of the whole data set speaking time in *Boot.1*, and 20.6% in *Boot.2*.

[3]The French National Laboratory of Metrology and Testing

5. Conclusions

The cross-show speaker diarization task complexity increases according to the size of the data. The incremental approach may be convenient to process large and dynamically increasing collections. The results are close to that obtained with the overall ILP clustering performed on the whole data set, but what makes this architecture interesting is the speed/accuracy trade-off, especially thanks to the pre-processing with connected graph subcomponents. It took 6h17 to process the 184 shows that were iteratively inserted into the collection. The same experiment without using the graph pre-processing lasts for more than 3 days. The same experiment with an overall HAC/GMMs clustering approach was still running after 2 weeks of computation; we had to stop it. It is still reasonable to process the collection with an overall ILP clustering with 142 hours of data, however it will be inappropriate as the collection increases.

6. Acknowledgements

This research was supported by the French ANR, under the *DEFI-REPERE* evaluation project (contract number ANR-2010-CORD-101-01, SODA project), and by the European Commission, as part of the *EUMSSI* project (contract number FP7-ICT-2013-10).

7. References

[1] M. Ferràs and H. Bourlard, "Speaker Diarization and Linking of Large Corpora," in *Proceedings of IEEE Workshop on Spoken Language Technology*, Miami, Florida (USA), December 2012.

[2] H. Bourlard, M. Ferras, N. Pappas, A. Popescu-Belis, S. Renals, F. McInnes, P. Bell, and M. Guillemot, "Processing and Linking Audio Events in Large Multimedia Archives: The EU inEvent Project," in *Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM)*, Marseille, France, August 2013.

[3] H. Ghaemmaghami, D. Dean, and S. Sridha, "Speaker Attribution of Australian Broadcast News Data," in *Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM)*, Marseille, France, August 2013.

[4] V.-A. Tran, V. B. Le, C. Barras, and L. Lamel, "Comparing Multi-Stage Approaches for Cross-Show Speaker Diarization," in *Proceedings of Interspeech*, Florence, Italy, August 2011.

[5] Q. Yang, Q. Jin, and T. Schultz, "Investigation of Cross-show Speaker Diarization," in *Proceedings of Interspeech*, Florence, Italy, August 2011.

[6] M. Rouvier and S. Meignier, "A Global Optimization Framework For Speaker Diarization," in *Proceedings of Odyssey 2012: The Speaker and Language Recognition Workshop*, Singapore, June 2012.

[7] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting Intra-Conversation Variability for Speaker Diarization," in *Proceedings of Interspeech*, Florence, Italy, August 2011.

[8] G. Dupuy, M. Rouvier, S. Meignier, and Y. Estève, "I-vectors and ILP Clustering Adapted to Cross-Show Speaker Diarization," in *Proceedings of Interspeech*, Portland, Oregon, USA, September 2012.

[9] G. Gravier, G. Adda, N. Paulsson, M. Carré, A. Giraudel, and O. Galibert, "The ETAPE Corpus for the Evaluation of Speech-based TV Content Processing in the French Language," in *Proceedings of LREC*, Istanbul, Turkey, May 2012.

[10] O. Galibert and J. Kahn, "The First Official REPERE Evaluation," in *Proceedings of Interspeech satellite workshop on Speech, Language and Audio in Multimedia (SLAM)*, Marseille, France, August 2013.

[11] S. H. Shum, W. M. Campbell, and R. D. A., "Large-Scale Community Detection on Speaker Content Graphs," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013.

[12] G. Dupuy, S. Meignier, P. Deléglise, and Y. Estève, "Recent Improvements on ILP-based Clustering for Broadcast News Speaker Diarization," in *Proceedings of Odyssey 2014: The Speaker and Language Recognition Workshop*, Joensuu, Finland, June 2014.

[13] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An Open-source State-of-the-art Toolbox for Broadcast News Diarization," in *Proceedings of Interspeech*, Lyon, France, August 2013.

[14] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J.-F. Bonastre, and G. Gravier, "The ESTER phase II evaluation campaign for the rich transcription of French broadcast news," in *Proceedings of Eurospeech*, Lisbon, Portugal, September 2005.

[15] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," in *Proceedings of IEEE TASLP*, vol. 19, 2011, pp. 788–798.

[16] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of I-vector Length Normalization in Speaker Recognition Systems," in *Proceedings of Interspeech*, Florence, Italy, August 2011.

[17] P.-M. Bousquet, D. Matrouf, and J.-F. Bonastre, "Intersession Compensation and Scoring Methods in the I-vectors Space for Speaker Recognition," in *Proceedings of Interspeech*, Florence, Italy, August 2011.

[18] S. Galliano, G. Gravier, and L. Chaubard, "The ESTER 2 Evaluation Campaign for the Rich Transcription of French Radio Broadcasts," in *Proceedings of Interspeech*, Brighton, UK, September 2009.

[19] O. Galibert, "Methodologies for the Evaluation of Speaker Diarization and Automatic Speech Recognition in the Presence of Overlapping Speech," in *Proceedings of Interspeech*, Lyon, France, August 2013.