Fast Single- and Cross-Show Speaker Diarization Using Binary Key Speaker Modeling

Héctor Delgado, Xavier Anguera, Corinne Fredouille, and Javier Serrano

***Abstract*—Speaker diarization has become a key process within other speech processing systems which take advantage of single-speaker speech signals. Furthermore, finding recurrent speakers among a set of audio recordings, known as cross-show diarization, is gaining attention in the last years. Current state-of-the-art-systems provide good performance, but usually at the cost of long processing times. This limitation may make current systems not suitable for real-life applications. In this line, the speaker diarization approach based on binary key modeling provides a very fast yet accurate alternative. In this paper, we present the last improvements applied in binary key speaker diarization with the aim of further speeding up the process and improving performance. In addition, we propose a novel method for cross-show speaker diarization based on binary keys. Experimental results show the effectiveness of the proposed improvements for single-show speaker diarization, both in terms of speed and performance, obtaining a real-time factor of 0.0354xRT and a 16.8% relative improvement in performance. Furthermore, our proposed cross-show approach provides very competitive performance, just slightly worse than its single-show diarization counterpart, and exhibits a real time factor of 0.04xRT.**

***Index Terms*—Binary key speaker modeling, cross-show speaker diarization, speaker diarization, within-class sum of squares.**

I. Introduction

SPEAKER diarization is the task of partitioning an input audio stream into speaker-homogeneous segments and grouping them by speaker identity. Speaker diarization is usually performed without any prior information about the number of participant speakers nor their identities. It is, therefore, an unsupervised process which involves two different tasks: speaker segmentation, which aims at finding potential speaker change points within the audio, and speaker clustering to group the originated segments by speaker, assigning them abstract speaker IDs. The importance of speaker diarization is well known as a pre-processing tool for further speech-related technologies which benefit from speech signals of single speakers. It can be used, for example, to improve Automatic Speech Recognition (ASR) accuracy by means of speaker adaptation, to provide speaker-separated segments to speaker identification and verification systems, or in audiovisual media indexing tasks.

Majority of diarization approaches usually follow one of the two main strategies, namely the agglomerative clustering approach [1], [2], and [3]), and the divisive approach [4] and [5]. Alternatively, there are also a number of systems which deviates from those two main clustering trends [6], [7], [8].

Even though much work has been conducted in the field of speaker diarization (obtaining very competitive performance), there are still a number of open challenges. One of them concerns speed when processing large amounts of data. With the increasing volume of audiovisual content, systems should be fast enough to process hundreds of hours in a reasonable time period. Many current systems usually perform a combination of several costly algorithms applied in an iterative scheme, which makes it very difficult to process audio streams faster than real time and, in consequence, those systems may not be suitable for real life applications where time is a key requirement. These slow systems could be sped up though code parallelization, with the subsequent requirement of resources. Other modern approaches provide very fast alternatives but usually at the expense of the requirement of vast amounts of external training and development data used for estimating the resources needed.

A second open challenge is related to recurrent speakers participating across several files within an audiovisual content database. Current single-show speaker diarization assigns abstract labels to the detected speakers in a per-audio-file basis. Thus, a given recurring speaker will receive different IDs in different audio files. In this situation, it would be really valuable to have unique IDs for each recurrent speaker. This task has been referred in the literature to as cross-show speaker diarization, speaker linking [9], or speaker attribution [10], and requires a first application of individual speaker diarization. Performing this task efficiently is directly related to the challenge of speed described above: fast single speaker diarization is required in order to get the initial speaker clusters as fast as possible.

Recently, a novel speaker diarization framework was proposed in [11], based on the "binary key" speaker modeling described in [12]. The paper reports DER scores of around 27% with a real time factor of 0.103 xRT using all the NIST

Manuscript received April 15, 2015; revised July 24, 2015; accepted September 03, 2015. Date of publication September 15, 2015; date of current version September 21, 2015. This work was supported by the Spanish Ministerio de Economay Competitividad as part of the project "Linguistic and sensorial accessibility: technologies for voiceover and audio description," under Grant FFI2012-31024, in part by the Catalan Government Grant Agency Ref.2014SGR027, and in part by the French Research program ANR Project PERCOL (ANR 2010-CORD-102). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tomi Kinnunen.

H. Delgado and J. Serrano are with the Department of Telecommunications and Systems Engineering, Autonomous University of Barcelona, 08913 Cerdanyola del Vallés, Spain (e-mail: hecdelflo@gmail.com; javier.serrano@uab.cat).

X. Anguera is with the Multimedia Research Group, Telefonica Research, 08021 Barcelona, Spain (e-mail: xanguera@tid.es).

C. Fredouille is with the University of Avignon, CERI/LIA, F-84911 Avignon Cedex 9, France (e-mail: corinne.fredouille@univ-avignon.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2479043

RT databases of meeting recordings. This technique provides a fast alternative, at the cost of a slight performance degradation. However, obtained results were still considered preliminary, leaving room for further investigation. In this line, more research in binary key speaker diarization has been conducted. In [13], the binary key based system was tested on broadcast TV data. Later in [14], alternative clustering and stopping criterion methods were proposed against some limitations of the original proposal.

In this paper we summarize these improvements and describe our last efforts on getting a very fast yet accurate speaker diarization system suitable for processing large quantities of data with recurrent speakers. We are aware of the great performance achieved by modern speaker diarization systems based on advanced speaker modeling, such as Gaussian supervectors, speaker factors and i-vectors, but those approaches require the use of vast amounts of training and development data for the estimation of Universal Background Models (UBM), total variability matrices, and so on. We are interested in getting a very fast diarization system with competitive performance, without the requirement of external data. Indeed, our approach does not require any external training nor development data; all the resources involved are trained on the test data itself, which avoids possible effects of training/testing condition mismatch.

To achieve our purposes, we define two main objectives. The first one is to further improve performance of the binary key speaker diarization technique, always having in mind system speed as the strongest feature[1]. In this sense, we propose some modifications to key parts of the system. Second, we propose a new approach to cross-show speaker diarization based on binary keys, where speaker clusters obtained from single diarization are represented by binary keys, which are later clustered to obtain clusters of recurrent speakers. Then all our proposals are assessed and validated experimentally.

This paper is structured as follows: Section II gives a review of recent related work. Section III describes the binary key based speaker diarization system, including baseline performance. Section IV describes the proposed improvements to the baseline system. In Section V, a new cross-show speaker diarization approach based on binary keys is proposed. Section VI shows experimental results. Finally, Section VII concludes and proposes further work.

II. Related Work

This section gives an overview of recent work focused on two different branches. The first one refers to speeding up speaker diarization, while the second refers to cross-show speaker diarization.

A. Fast Single-Show Speaker Diarization

Most state-of-the-art systems rely on the use of Gaussian Mixture Model (GMM) for speaker segmentation and clustering, trained on acoustic features using maximum likelihood or discriminative training approaches [15]. GMMs and metrics like Bayesian Information Criterion (BIC) or Generalized Likelihood Ratio (GLR) are intensively both in the detection of speaker change points, and for speaker cluster modeling, cluster merging and stopping criterion. Moreover, a final re-segmentation stage may be carried out through Viterbi decoding to refine the output clustering. In short, all the mentioned algorithms are applied iteratively, imposing a high computational load which results in processing times which may be too long for some real-life applications (above 1xRT, being xRT the Real Time factor).

Given the need of fast systems able to process large data amounts, some efforts have been done in order to speed up speaker diarization. In [16], fast-match methods are investigated for reducing the hypothesis space of the BIC approach in order to select the most likely clusters to merge, with the consequent computational savings, getting 0.88 xRT. Although faster than real-time, this approach seems not fast enough to process large databases quickly. Later, in [17], a novel framework to speaker diarization based on the information bottleneck principle was proposed. This approach provides similar performance to classic GMM-based systems over meeting recordings, but at a lower computational cost, achieving around 0.3 xRT. Further on, the use of parallel hardware was investigated. [18] proposes a parallel implementation of the GMM training for a GPU, achieving great speed ups between 0.02-0.004 xRT. However, the approach is very dependent on complex, non-standard hardware architectures and low-level programming methodologies.

Lately, with the expansion of the last achievements in speaker recognition (Gaussian supervectors, joint factor analysis, i-vectors), new advanced speaker diarization techniques have emerged. Normally, each speech segment/cluster is represented by a single vector of a certain dimension. Then, speaker clustering is reduced to cluster those single vectors. The vectors can be efficiently compared through some fast similarity measure, such as the dot-score or the cosine distance. Therefore, the recent "supervector" paradigm is presented as a potential fast approach to speaker clustering. In spite of this, speaker segmentation is still commonly faced through BIC-based and other similar approaches. Consequently, a speaker diarization system composed of classic segmentation plus a supervector approach will presumably suffer of computational issues. An example of such an approach is the one described in [19]. This system first performs speaker segmentation based on a maximum likelihood approach, followed by a two-stage clustering scheme, the first one based on BIC, and the second one based on i-vectors. The acoustic-based segmentation stage shows a 0.14 xRT, although this figure was further reduced to 0.02 by taking advantage of the output of a ASR system. With regard to the clustering stage, a real-time factor of 0.05 xRT was reported. Therefore, total 0.19 xRT for the only-acoustic-based system, whilst 0.07 xRT for the ASR-aided system were obtained. In [20], a cluster-voting approach avoids the use of a dedicated Viterbi-based speaker segmentation stage by means of a clustering-only scheme where JFA is performed to equal-sized segments from the input data, which are then clustered by complete-linkage Agglomerative Hierarchical Clustering (AHC). Even if a great speed up is achieved, these approaches require enough external training data to estimate

[1] The Matlab code of the single-session binary key speaker diarization system can be downloaded at http://hectordelgado.me/software

Universal Background Models (UBM), total variability matrices for i-vector estimation, and other required resources.

B. Cross-Show Speaker Diarization

Cross-show speaker diarization aims at expanding the speaker diarization task to a broader context, where speakers participating in different recordings along a collection of audio files must receive the same speaker identifier. Three main schemes for cross-show speaker diarization have been proposed in the literature [21], [22]. The global approach by concatenation (1) is the most straightforward and naive method, in which all the shows in the dataset are concatenated and diarization is performed on the resulted pooled audio file. However, memory and computation requirements of this approach grow exponentially with an increasing number of shows. The hybrid approach (2) partially solves the computational limitations of the global approach by concatenation by first performing individual speaker diarization on each show. Second, speaker clusters returned by the individual processes are globally clustered. This approach is more efficient because the global clustering is done over a limited number of speaker clusters. The incremental approach (3) is intended for dealing with audio databases that are increased over time by the addition of new shows. In this approach, the information of the clusters of the already processed shows is used for incorporating the single-show speaker clusters of the new show into the global cross-show clustering. This task can be addressed, for example, by means of an open-set speaker identification system [22] or by performing a new global clustering on the cross-speaker models of the past shows and the single-show speaker models of the new show jointly [23].

With regard to the global speaker clustering, BIC [21] and Cross Likelihood Ratio (CLR) [22] have been used as merging criteria within a global AHC stage. Other approaches use advanced methods like JFA and i-vectors to represent speaker clusters. For example, the system described in [24] extracts an i-vector for each speaker cluster from the single diarizations, which are next clustered by a global clustering approach formulated as an optimization process of Integer Linear Programming (ILP).

III. Baseline Binary Key Speaker Diarization System

This section describes the binary key based speaker diarization system taken as a baseline in this work, including performance figures. The system is based on the system described in [11] and [13] (with some bug fixes and minor changes that are specified further on in this section). The system is divided into two well-differentiated modules. First, the acoustic module aims at transforming the input acoustic data into a binary representation which preserves speaker related information. Second, the binary processing module performs a bottom-up AHC, in which data re-assignment and cluster merges are performed iteratively, but on the basis of the binary representation obtained in the previous stage. Next, we describe each of the two modules in more detail.

A. Acoustic Processing Module

The acoustic module performs a transformation of the input acoustic feature vectors into binary vectors called Binary Keys (BK). The key element for this transformation is a UBM-like acoustic model, called binary Key Background Model (KBM), which is trained using the test input data itself, but in a particular way. Once the KBM is trained, any sequence of acoustic feature vectors can be compacted into a single BK which maintains speaker-related information.

1) Binary Key Background Model Training: The KBM is a UBM-like acoustic model which is used to convert acoustic features into binary features. The original KBM training procedure for binary speaker modeling [12] uses an initial Universal Background Model (UBM) from which adapted GMMs are obtained by using a set of anchor speakers. Then, the KBM is the result of pooling all GMMs together. Although this approach could be applied to speaker diarization, it is reasonable to think that performance will suffer due to the mismatch between training and testing data. Instead, a novel method for KBM training for speaker diarization was introduced in [11]. This method does not require external data: the test data itself is directly used for training.

The KBM training algorithm for speaker diarization first extracts a pool of single Gaussians from the input features, followed by an iterative process of Gaussian selection in order to select the most complementary and discriminant Gaussians, with the aim of retaining full coverage of the speaker acoustic space of the test data. For Gaussian component extraction, a fixed-length window is used to train single Gaussians, with some window shift (and overlap). The shift value is dependent on the length of the data, and is set in order to obtain several hundreds of components. Once the Gaussian pool is obtained, components are selected iteratively until having the desired number of Gaussians. The first selected component is the one which best models the data segment it was trained from (i.e. $\arg\max_i \mathcal{L}(s_i, \theta_i)$, where $\mathcal{L}$ denotes likelihood, θ_i is the Gaussian trained with the i-th segment s_i). For the iterative Gaussian selection, a global dissimilarity vector v_{KL2} is defined to represent distances between the already selected Gaussians to all others remaining in the pool. This vector is first initialized to ∞ as no component is selected yet. The process then works as follows: (1) Compute the KL2 (symmetric Kullback-Leibler) divergence between the previously selected Gaussian θ' and the rest of Gaussians θ_k still not selected, and set $v_{KL2}[j] = \min(v_{KL2}[j], S_{KL2}(\theta', \theta_j))$; (2) Add to the KBM the Gaussian θ^k with the highest dissimilarity with those already selected (i.e. $\arg\max_k(v_{KL2}[k])$); (3) Go back to (1) until the desired number of components in the KBM N is reached.

It could be argued that a GMM trained on the test data could be used instead of the KBM. However, it has been shown in [12] and [11] that the KBM is able to produce much more discriminative BKs than a classic GMM. A possible explanation to this fact is that the components of a GMM trained, for example, through iterative Gaussian splitting, model the average acoustic space, whilst the KBM components retain acoustic information of the particular speakers, since most of the components are trained on pure speaker data. A second advantage is its lower computational cost compared to the expectation-maximization algorithm.

2) Binary Key Estimation: Once the KBM is trained, any set or sequence of input feature vectors can be converted into a binary key. A binary key $\mathrm{v}_f = \{v_f[1], \ldots, v_f[N]\}, v_f[i] = \{0, 1\}$ is a binary vector whose dimension N is the number of components in the KBM. Setting a position $v_f[i]$ to 1 (TRUE) indicates that the i-th Gaussian of the KBM coexists in the same area of the acoustic space as the acoustic data being modeled. The binary key can be obtained in two steps. First, for each feature vector, the IDs of the best N_G matching Gaussians in the KBM are selected and stored (i.e., the N_G Gaussians which provide highest likelihood for the given feature vector). Second, the count of how many times each component has been selected as a top component along all the features is calculated and stored in a Cumulative Vector (CV). Finally, the binary key is obtained by setting to 1 the positions corresponding to the top M Gaussians at the whole feature set level, (i.e., the M highest positions of the CV). Intuitively, the binary key keeps the components of the KBM which best fit the data being modeled, preserving only the ones with the highest impact. As the KBM is expected to contain Gaussians for all the participating speakers in the input audio file, the activated Gaussians in the BK will ideally be the ones trained on speech regions uttered by the same speaker present in the speech segment being converted. Therefore, the subsets of activated Gaussians per speaker will ideally be highly disjoint. This is the key for producing speaker-discriminative BKs.

Note that this method can be applied to any set of features, either a sequence of features from a short speech segment, or a feature set corresponding to a whole speaker cluster. This fact will make the comparison between two binary keys straightforward, either between segment-cluster binary key pairs or cluster-cluster binary key pairs.

In our speaker diarization scheme, all the input data is transformed into binary keys as follows: first, the data is divided into superframes of fixed length (for example, 1s superframes with some overlap). For each superframe, a BK is estimated following the method explained above. Note that this process requires the extensive computation of likelihoods of all acoustic features against all the Gaussian components in the KBM. However, these likelihoods are computed only once and stored in a matrix which can be re-utilized in later steps. The top N_G Gaussians per frame can also be stored and re-used in the computation of new BKs (cluster BKs). From this point on, all the subsequent processes will be performed over this binary representation, what will result in important computational savings since all operations will be performed on vectors of zeros and ones, involving fast, bit-wise operations.

3) Clustering Initialization: Before switching to the binary processing block, the set of clusters has to be initialized. Clustering initialization in speaker diarization has been extensively addressed in the literature, but the problem is not still solved totally. The original method for cluster initialization for binary key speaker diarization described in [12] takes advantage of the first N_{init} components in the KBM as seed models to obtain a first, maximum-likelihood-based, over-segmented clustering of N_{init} clusters. This method re-uses the likelihoods computed at the beginning when obtaining the BKs, so the initialization is efficient.

B. Binary Processing Module

The binary module implements an agglomerative clustering approach. However, all operations are done with binary data, which makes the process faster than with classic GMM-based approaches.

After estimating BKs for the N_{init} initial clusters by following the method described in Section III-A2, the AHC process proceeds as follows: (1) The input BKs are re-assigned to the current clusters. This is done by calculating similarities between each input BK and each cluster, assigning them to the cluster which provides the highest similarity according to Equation (1):

$$S(\mathrm{v}_{f1}, \mathrm{v}_{f2})) = \frac{\sum_{i=1}^{N}(v_{f1}[i] \wedge v_{f2}[i])}{\sum_{i=1}^{N}(v_{f1}[i] \vee v_{f2}[i])} \tag{1}$$

where v_{f1} and v_{f2} are the BKs being compared, $\wedge$ indicates the Boolean AND operator, and $\vee$ indicates the Boolean OR operation. It is, therefore, a very fast, bit-wise-based, operation between two binary vectors. This similarity metric returns a value between 0 and 1, 0 indicating total dissimilarity, and 1 indicating total similarity (all the elements are equal). (2) Cluster-wise BK similarities are computed, the closest cluster pair is merged, and the BK for the new cluster is computed. (3) The obtained clustering is stored and the algorithm goes back to (1) while the current number of clusters is > 1.

The previous process outputs a set of N_{init} clustering solutions, each one with a decreasing number of clusters. The optimum solution has to be selected from those returned through some clustering selection algorithm. [11] proposed an adaptation of the T-test T_s metric described in [25]. A given clustering solution consists of a set of clusters grouping the equal-sized segments from the input data, represented as BKs. First, the statistics of intra-cluster and inter-cluster similarity distributions (i.e. the distributions of all similarities between binary keys obtained from segments in the same cluster and between all binary keys from segments in different clusters) are calculated. Then, assuming that both distributions are Gaussian-shaped, T_s is calculated as

$$T_s = \frac{m_1 - m_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \tag{2}$$

where m_1, σ_1, n_1, m_2, σ_2 and n_2 are respectively the mean, standard deviation and size of the intra-cluster and inter-cluster distributions. Finally, the clustering which maximizes the T_s value is selected.

C. Baseline System Evaluation

In this subsection we report experimental results of the baseline binary key based speaker diarization system described in the previous paragraphs. This baseline system is similar to the one described in [13]. The differences in the results are mainly due to the way performance is computed. In the former work, Diarization Error Rate was computed excluding audio regions

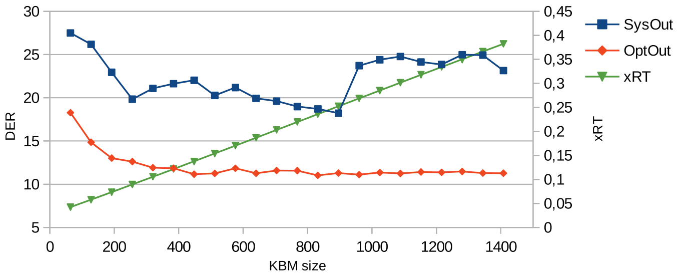

Fig. 1. Baseline Speaker Diarization System performance measured in DER. “SysOut” refers to system output, “OptOut” refers to the best clustering in terms of DER selected manually (performance ceiling), and “xRT” (secondary Y axis) refers to the real-time factor.

TABLE I
EXECUTION TIME (WITH KBM size = 896) IN SECONDS AND xRT FOR TWO AUDIO FILES OF DIFFERENT DURATIONS

	Audio file 14:43 (883s)		Audio file 01:28 (88s)	
Stage	Time (s)	xRT	Time (s)	xRT
KBM training	**66.23**	**0.0749**	**76.94**	**0.8733**
BK estimation	35.54	0.0402	3.14	0.0356
Clustering init.	0.433	0.0004	0.042	0.0004
AHC	25.29	0.0286	3.09	0.0351
Clustering selection	19.44	0.022	0.12	0.0014
Overall	146.933	**0.1661**	83.332	**0.9458**

with overlapping speech. In this work, we opt for measuring performance taking overlapping speech into account (even if our system does not assign additional speaker labels to overlapping regions) in order to enable performance comparison with other works in the literature. A second difference with regard to [13] is in the Speech Activity Detection (SAD) labels used. While in the first work SAD labels were obtained automatically (by using two different methods), in this paper we are using ground-truth SAD labels obtained from the reference speaker labels. This is done to put the focus on the different modules involved in the system, without the impact of the effects introduced by incorrect SAD labels.

The system is evaluated on the REPERE phase 1 test dataset of TV broadcast data (please refer to Section VI for further details). Feature extraction is performed: standard 19-order MFCCs are computed using a 25 ms window, every 10 ms. For training the KBM, single Gaussian components are estimated using a 2s window in order to have sufficient data for parameter estimation. Window rate is set according to the input audio length, in order to obtain an initial pool of 2000 Gaussians. With regard to binary key estimation parameters, the top 5 Gaussian components are taken in a frame basis, and the top 20% of the components at segment level. For cluster initialization, the number of initial clusters is set to 25 in order to assure a number greater than the maximum number of speakers in the database (up to 18 in some excerpts), and the rough clustering is performed by dividing the input features into small chunks of 100 ms and assigning them to the clusters through maximum likelihood. Finally, in the agglomerative clustering stage, binary keys are computed for each 1s segment, augmenting it 1s before and after, totaling 3s.

As mentioned above, performance is measured by calculating DER, which is the most standard metric for speaker diarization. In this work, the evaluation tool developed by LNE for the REPERE evaluation campaign [26] is used. The main difference with the tool developed by NIST is the way the forgiveness collar is applied to the overlapping speech regions. We use a forgiveness collar of 0.25s.

Fig. 1 shows performance of the baseline system measured in Diarization Error Rate (DER), according to the number of components in the KBM (and subsequently, the number of bits of the BKs), for the system output “SysOut” (i.e. the clustering returned by the clustering selection algorithm) and the optimum output “OptOut” (i.e. the clustering solution with lowest DER selected manually after calculating DER of all partial clustering solutions generated at each AHC iteration). Execution time is also provided in terms of Real Time Factor (xRT, secondary Y axis), calculated as the total time taken by the system to process the input data (excluding feature extraction) divided by the speech time (duration of the portions of the input audio labeled as speech).

Comparing DER of optimum and system outputs, it can be seen that the clustering selection algorithm is far from returning the best clustering which the system is actually able to generate (the optimum one in terms of DER). In fact, DER of system output is not very stable, oscillating between 18% and 25% when the number of components is incremented, whilst DER of the selected optimum clusterings converges around 11.5%.

With regard to computation time, xRT increases linearly with the size of the KBM, ranging from 0.042 (23.8 times faster-than-real-time) with 64 components, until 0.382 (2.6 times faster-than-real-time) with 1408 components.

The best system output result is 18.22% DER with xRT = 0.252, obtained with a KBM of 896 components. With this configuration, the optimum clustering selected manually provides a DER of 11.29%. This suggests that a better clustering selection algorithm should be able to return closer-to-the-optimum clusterings.

It is interesting to compare xRT figures achieved in this work with the ones obtained in [13] with the RT-05 database of meetings. In that work, xRT using similar KBM size was around 0.18, which is quite lower than the 0.25 obtained here. The reason is that the iterative Gaussian selection algorithm to train the KBM affects in the same way to the long and short audio files, as the window rate is set to get a constant number of components (around 2000), regardless of the duration of the audio file being processed. Then, the desired number of Gaussians is obtained iteratively. In the case of the RT-05 database, all the excerpts have similar duration, whilst in the REPERE database, audio durations can range between 1 and 15 minutes. The KBM training results in a high penalty in execution time when the audio file is very short, while longer audio files are favored. Table I shows execution time figures for two audio files of different durations, broken down into the involved processes in the diarization system (note that feature extraction xRT of 0.002 is not included, as we keep this stage invariant along all the experiments in this work). It can be appreciated that the KBM training takes even more time for the short audio file than for the long one. This has a big impact on the calculation of xRT for the short audio file.

IV. Recent Improvements

In this section we describe recent work done on improving overall performance of the binary key speaker diarization approach described in Section III, having system efficiency always as a priority. Later, all the improvements are tested together and evaluated in Section VI.

A. KBM Training

We showed in Section III-C that the main bottleneck of our approach is the KBM training process. It has been shown that this part penalizes the execution time, especially when the input audio file is short. Speeding up this part would result in considerable gains in overall execution time.

The Gaussian component selection is done in an iterative way by calculating the KL2 divergences between the last selected Gaussian and the remaining ones. KL2 provides a measure of how different two probability distributions are. D_{KL2}, namely "Symmetric Kullback-Leibler Divergence," of distributions P and Q is defined as $D_{KL2}(P||Q) = D_{KL}(P||Q) + D_{KL}(Q||P)$, where $D_{KL}(P||Q)$ is the Kullback-Leibler divergence of distributions P and Q. Computation of KL (refer to [27] for the closed form of KL for multivariate normal distributions) involves a series of matrix operations, including traces, inversions and determinants. However, we wondered if we could use a simpler and faster, yet useful, method for our purposes. As the aim of the iterative Gaussian selection process is to select the most discriminant and complementary ones, maybe calculating distances between the means (centroids of the distributions) of the Gaussians could be powerful enough to select the most dissimilar components. Following this reasoning, we propose the use of the cosine distance between Gaussian mean vectors as similarity metric. The cosine distance $D_{cos}(a,b)$ is defined as $D_{cos}(a,b) = 1 - S_{cos}(a,b)$, where $S_{cos}(a,b)$ is the cosine similarity between two vectors a and b, defined by Equation (3) as

$$S_{cos}(a,b) = \frac{a \cdot b}{||a|| ||b||} \tag{3}$$

The cosine similarity formulation is considerably simpler than KL2 one, and its computation is faster. We also show in the experimental section that the cosine similarity is discriminant enough and suitable in our Gaussian selection algorithm.

B. Binary Key Similarity Measures

As shown in Section III, BKs are capable of keeping speaker-discriminative information in a vector of binary values. Intuitively, positions equal to one within a BK indicates that those Gaussians of the KBM are the ones that best fit the sequence of feature vectors being converted. Those Gaussians are selected according to the frequency of component activation for the given feature set. In other words, positions in the BK set to one correspond to the top positions in the Cumulative Vector (CV, see Section III-A2). The CV stores the relative weights of each Gaussian of the KBM given the input feature set. These weights are lost in the conversion of the highest positions to the binary values. Thus, it is reasonable to argue that this missing information could also be helpful for discriminating between speakers. Using CVs instead of BKs has already been addressed for speaker verification with success in [28], and we decided to use CVs in this work as well.

An effective similarity measure between CVs has to be found. Each position of the CV contains a positive integer representing the count of how many times its associated Gaussian of the KBM has been selected as a top-scoring component. Therefore, the absolute values depend on the length of the set of input features. However, we are more interested in the relative weights than in the CVs magnitude. It seems a case where the cosine similarity can be suitable for our purposes. The cosine distance S_{cos}, already defined by Equation (3), provides the cosine of the angle between the two vectors. As CVs are vectors of positive integers, the similarity measure will be comprised in the interval (0,1). Note that the use of the CVs as speaker models makes the system no longer "binary". However, CVs are a partial result of the calculation of BKs and are therefore efficient to compute. The cosine distance between vectors of integer/real values is a very efficient operation as well.

C. Clustering Selection

Baseline results showed that the weakest point of the entire binary key speaker diarization system is the selection technique of the best clustering. It has been reported that the T_s based algorithm is far from returning the optimum number of speakers in our system. It has also been shown that there exists a performance ceiling (or DER floor, i.e. DER of the optimum clustering selected manually), which is around 11.5% in the baseline system. The best performance returned by the clustering selection algorithm is 18.22%. This indicates that a better clustering selection will systematically result in an increase of performance. It is a must to propose a different approach that can get closer to the performance ceiling.

In this paper, we propose a clustering selection technique based on the Within-Cluster Sum of Squares (WCSS). To the best of our knowledge, this criterion has not been previously used for speaker clustering. Given a clustering solution C_k composed of k clusters $c_1, c_2, \ldots, c_k$, the WCSS, $W(C_k)$, is defined as

$$W(C_k) = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - \mu_i||^2 \tag{4}$$

where μ_i is the mean of the points of cluster c_i (i.e. the centroid of cluster c_i). The WCSS can be used as an indicator of how good a clustering solution is. Presumably, a good clustering solution yields clusters with small WCSS. Given a set of clustering solutions $C = (C_1, \ldots, C_{N_{init}})$, each one with an increasing number of clusters (from a single cluster to N_{init} clusters), WCSS can be calculated for all clustering solutions and plotted as shown in Fig. 2. When the number of clusters N is less than the optimum number of clusters, WCSS should be high. In the case of $N = 1$, WCSS is maximum, and when increasing the number of clusters, WCSS will exhibit an exponential decay. In some point the decay will show almost linear behavior and WCSS will continue to fall smoothly. The first point which deviates from the exponential model is considered as the elbow, and the associated number of clusters is selected as

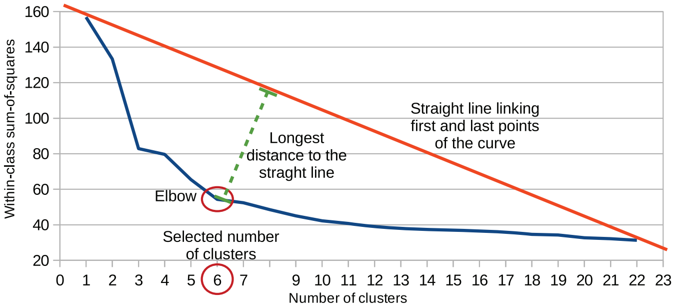

Fig. 2. Example of the elbow criterion applied over the curve of within-class sum-of-squares per number of clusters.

the optimum one. A simple graphic approach to find the elbow is to draw a straight line between the WCSS values of the first (with $N = 1$) and last ($N = N_{init}$) clustering solutions and calculate the distance between all points in the curve to the straight line. The elbow is the point with the highest distance to the line.

In the formulation of Equation (4), the Euclidean distance of each cluster member to its centroid is used. However, we propose to use the cosine distance instead, which we find more adequate to compare CVs.

V. Cross-Show Speaker Diarization

According to the different strategies for cross-show speaker diarization described in Section II-B, our system follows a hybrid approach where first each show is diarized separately using our binary key based speaker diarization system, and at the end all resulting clusters are clustered to conform the final cross-show clustering.

We can separate our cross-show clustering in three parts. First, each returned cluster from the individual diarizations must be converted into BKs/CVs. Second, Inter-Session-Intra-Speaker (ISIS) variability compensation is carried out. Finally, the resulting compensated vectors are clustered to obtain clusters of speakers participating in more than one audio file.

A. Data Preparation

As it has been shown in Section III, the conversion of a given speech segment or cluster into a BK is done thanks to the KBM. The KBM is a UBM-like model trained on the test data itself. That means that each single show is processed using its own KBM. BKs obtained within a show are not directly comparable to BKs from another show generated using its KBM. This implies the need of a new "global" KBM trained on the whole dataset in order to generate cluster BKs globally comparable between them. To achieve that, we propose two different methods. The first one consists in pooling all the audio files and then to train the global KBM following the method described in Section III-A1. The second consists in taking all the individual KBMs (which could be saved within the single-diarization pass) and concatenate them to obtain a pool of Gaussian components. Then the iterative process of component selection described in Section III-A1 is applied to reduce the KBM size to a target number of components.

Once the KBM is obtained, each cluster can be easily converted into BKs/CVs using the computation method described in Section III-A2.

B. Inter-Session Intra-Speaker Variability Compensation

Generally, a given speaker who appears in different audio files would have been recorded under different conditions (different microphones, background noises, etc). Therefore, it is interesting to compensate inter-session variability to avoid the effects of such changing conditions. This way, it is thought that performing speaker linking across shows will be more effective on variability-compensated clusters. In this sense, we propose to apply Nuisance Attribute Projection (NAP) in the Cumulative Vector space.

Nuisance Attribute Projection: NAP [29] assumes that the within-class variability is restricted to a low dimensional subspace. In order to remove this variability, the supervectors are projected onto an orthogonal complementary subspace. First, the within-speaker scatter matrix W is calculated on appropriate labeled data. Then, the projection matrix is obtained as $P = (I - UU^t)$, where U is the matrix of the k eigenvectors associated with the k largest eigenvalues, obtained after solving the eigenvalue problem on W. Finally, the transformation of supervector x is done by calculating $y = Px$.

Estimating the within-class covariance matrix requires labeled data with speaker information. In this work we focus on getting the within-class covariance matrix in an unsupervised manner over the test data itself. We propose to estimate W using the obtained segmentation of individual speaker diarization on each show as speaker labels. Given a show, all speech assigned to speaker i is concatenated and divided into segments of 1 second. Each of those segments are treated as a speaker utterance in the calculation of W. The process is repeated over all speaker clusters of all individual diarizations until obtaining speaker labels for the whole dataset. Before calculating W, CVs have to be estimated for all 1 second segments using the global KBM.

C. Clustering

Recently, an alternative approach to the classical AHC for speaker clustering was presented in [30]. The main argument against AHC is the greedy nature of the technique, whose merging decisions are made based on local maxima. Furthermore, AHC does not allow to recover incorrect merges. The proposed alternative clustering method addresses the clustering as a global process, formulated as a problem of Integer Linear Programming (ILP), with the aim to minimize the number of clusters and the dispersion within them. This approach requires an initial speaker clustering from which i-vectors are computed. Then the ILP clustering is performed over those vectors. Therefore, this approach completely fits the cross-show diarization framework and was successfully applied in [24].

Let's recall how the ILP clustering works: Once a set of N clusters have been converted to N single vectors (multi-dimensional points), the goal is to group them into K clusters while minimizing a given objective function and satisfying some constraints. Some of the N points can act as "centers" of new clusters. The remaining ones (e.g., the ones not selected as centers) must be associated to one of the centers. Intuitively, the objective function consists in minimizing the number K of clusters and the dispersion of the points within each cluster. Regarding the constraints, each point which is not a center can be associated with only one center and its distance to the center must be short enough (below a given threshold).

We have adopted the proposal described in [31] with overall distance filtering, which is much more efficient that the original formulation proposed in [24], since the number of variables and constraints of the ILP problem is reduced significantly. First, computation of the matrix of all pair-wise distances for all points is done, and then the ILP problem is defined only for the candidate points that fulfills the minimum distance requirement. The ILP clustering can be formulated as:

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

Minimize

$$\sum_{k \in C} x_{k,k} + \frac{1}{D} \sum_{k \in K_j} \sum_{j \in C} d(k,j) x_{k,j} \tag{5}$$

Subject to

$$x_{k,j} \in \{0,1\} \qquad k \in K_j, j \in C \tag{6}$$

$$\sum_{k \in K_j} x_{k,j} = 1 \qquad j \in C \tag{7}$$

$$x_{k,j} - x_{k,k} < 0 \qquad k \in K_j, j \in C \tag{8}$$

Equation (5) is the objective function to be minimized, which aims at minimizing the number of clusters and the dispersion of the points within each cluster. The binary variable $x_{k,k}$ is equal to 1 if the point k is a center. $d(k,j)$ is the distance between points k and j. D is a normalization factor equal to the longest distance $d(k,j)$ for all k and j. The binary variable $x_{k,j}$ is set to 1 if point j is associated with center k. Equation (6) forces variable $x_{k,j}$ to be binary. Equation (7) ensures that each point j is associated with only one center k. Finally, Equation (8) ensures that if a point j is assigned to center k, then k is a center.

In our system, we adapt this approach by replacing the i-vectors with the CVs extracted in the stage described in Section V-A, and by using the cosine distance metric.

VI. Experiments and Results

This section evaluates our proposals experimentally. The complete system is implemented in Matlab. The code has been optimized to the extent possible in order to take advantage of all CPU cores. Experiments were conducted on an 2.7 GHz AMD Phenom II X6 processor with 8 GB RAM. First, a brief description of the database used for the experiments is given. Next, experimental results and discussion on single- and cross-show binary key speaker diarization are provided.

A. Database Description

In this work we use the REPERE corpus [32] of TV shows from French TV channels, developed in the context of the REPERE challenge. The phase 1 REPERE corpus consists of a training set, a development set and a test set of around 24, 3 and 3 hours respectively. In this work only the development and test sets are used. All the improvements proposed in this work have been developed and tested using the REPERE test set. However, we wanted to check if the performance is stable when processing new data, thus the REPERE development set is also used as additional unseen data to evaluate the system.

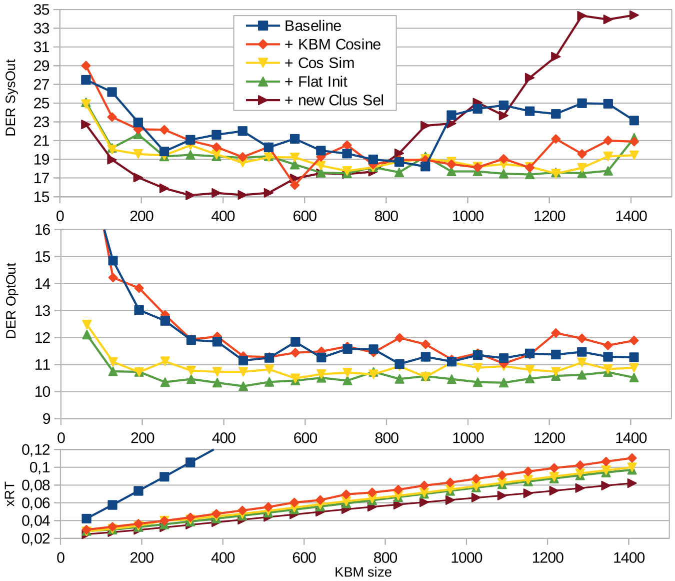

Fig. 3. Diarization performance after applying the proposed improvements one by one: DER of system output, DER of optimum output, and xRT.

The REPERE test set consists of 28 audio files, including different kinds of TV shows, one performing scripted speech, and others performing more spontaneous speech, with a higher rate of overlapping speech regions. The REPERE development set contains 25 audio files of the same nature.

B. Single-Show Speaker Diarization Experiments

The basic settings of the single-show speaker diarization experiments are essentially the same used for the evaluation of the baseline system (see Section III-C). As it is usual in the field of speaker diarization, we use the Diarization Error Rate (DER) as evaluation metric.

Fig. 3 shows system performance on the development data. We have taken an approach in which system performance is evaluated after adding each of the proposed improvements one by one, in order to see how performance evolves. We provide DER of the system output by using the best clustering selection algorithm ("SysOut," upper graph of Fig. 3), and DER of the optimum output the system is able to generate, i.e. all partial clusterings generated by the system are evaluated, ignoring the best clustering selection, and then the best labeling in terms of DER is selected manually ("OptOut" middle graph of Fig. 3). We provide these results in order to establish a performance ceiling which enables us to check how close the clustering selection algorithm is to the optimum clustering. Finally, execution time figures are provided in terms of xRT (lower graph of Fig. 3). All there results are shown with regard to the KBM size in steps of 64 components (X axis).

The first result shown ("Baseline," blue line) is the baseline performance, corresponding to the same figures shown in Fig. 1. The second one ("+ KBM Cosine," orange line) is the same system replacing the KBM training procedure by the one described in Section IV-A. We observe an improvement of SysOut, particularly for KBM sizes greater than 896. However, the best improvement of this modification is undoubtedly in terms of computation. It can be observed that the xRT trend has significantly less slope than the baseline (which gets out of the graph).

The next improvement consists in using the cosine similarity between CVs (“+ Cos Sim,” yellow line). Generally, system performance gets even better in both OptOut and SysOut. As it was hypothesized, the information contained in the CVs results in better speaker discrimination. With respect to execution time, xRT slope decreases a bit more with the use of the new similarity measure, although it can be due to implementation efficiency, as optimized, built-in Matlab functions are used in this case.

The next result corresponds to replacing the cluster initialization strategy by a simple flat initialization, in which the input signal is divided into N_{init} uniforms chunks which conform the initial clusters (“+ Flat Init,” green line). Surprisingly, using such a rough uniform initialization performs better than the method explained in Section III. It has a positive impact in execution time as well.

Up to now, all the modifications have provided improvements both in terms of performance and speed. However, the gap between OptOut and SysOut is still high. The best SysOut is not smaller than 17% DER, whilst the best SysOut is slightly above 10%.

The last modification is with regard to the clustering selection technique. Here, the elbow criterion over the within-cluster sum-of-squares proposed in Section IV-C is applied (“+ new Clus Sel,” brown line). SysOut performance is finally improved, reaching DER values near 15%. The new approach performs better with small sizes of the KBM, contrarily to the trend of the original selection algorithm, which performs better with bigger KBMs. A possible explanation to this fact is that the distances between elements of the same cluster globally become longer when the KBM size is increased. Consequently, WCSS also increases with the number of components in the KBM, and when we get closer to one single cluster, the exponential region is importantly emphasized for bigger KBMs. This results in a tendency to over-cluster the data, producing less clusters than the actual number of speaker, which explains the sudden DER increase when the KBM size becomes higher. Contrarily, when the KBM size is small, the algorithm tends to under-cluster, which also explains the higher values of DER for those KBM sizes. Once again, execution is further sped up, as the number of pair-wise distance calculations required is reduced. Having in mind performance and efficiency, we could select a KBM size of 320 as the best performing configuration, obtaining DER $=$ 15.15% and xRT $=$ 0.0354 (around 28 times faster-than-real-time).

Let’s recall the execution time analysis given in Table I. There, we observed important differences in execution time, especially between short and long audio files. We have measured run time for the same two shows with the new faster system using a KBM of 320 components, and the result is given in Table II. It is observed that xRTs are significantly lower. One of the reasons is that the new modeling allows us to use smaller KBMs, resulting in speed ups in all the stages involved in the diarization. But the main reason is that the component selection algorithm of the KBM training is now much faster. With a KBM size of 320 Gaussian components, the baseline system presents an overall 0.105xRT, while the new improved system provides an overall 0.035xRT. However, according to the results in Table II, we still can observe a difference in the times required to estimate the KBM of short and long audio files. Once again, the reason is the estimation of a fixed number of initial number of Gaussians (2000 components in our case), regardless on the duration of the input audio.

TABLE II
EXECUTION TIME IN SECONDS AND XRT FOR TWO AUDIO FILES OF DIFFERENT DURATIONS AFTER APPLYING SPEEDING IMPROVEMENTS

	Audio file 14:43 (883s)		Audio file 01:28 (88s)	
Stage	Time (s)	xRT	Time (s)	xRT
KBM training	**1.07**	**0.0012**	**1.39**	**0.0158**
BK estimation	10.59	0.0119	1.43	0.0163
Clustering init.	0.0004	0.0000	0.0003	0.0000
AHC	13.83	0.0156	2.34	0.0266
Clustering selection	0.60	0.0006	0.21	0.0024
Overall	26.09	**0.0293**	5.37	**0.0611**

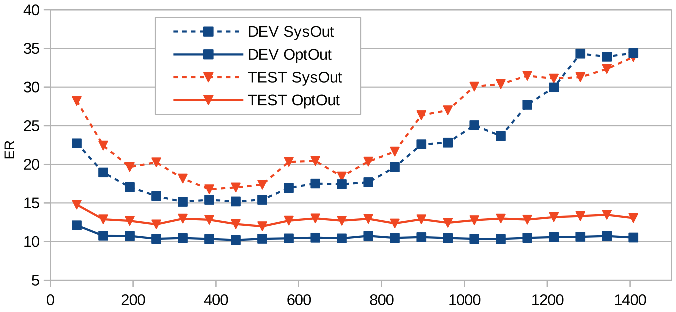

Fig. 4. Comparison of performance of the best performing system when processing the development and the test sets.

To check if the obtained results are stable across different audio data, we have selected the best performing system on the development data and we have conducted speaker diarization over a new dataset not used in the development. This dataset is the REPERE phase 1 development set, which we use here as test data. Fig. 4 shows system DER on the development (“DEV SysOut” and “DEV OptOut”) and test (“TEST SysOut” and “TEST OptOut”) data together. We observe how the system behaves very similarly with the two datasets, but with an approximately constant decrease in performance of around 2% absolute when processing the test set. As far as execution time concerns, xRT are practically identical.

In general, after applying all the proposed improvements we have obtained a very fast yet reasonably accurate diarization system suitable for processing large collections of audio files. In terms of execution time, our approach outperforms those described in Section II. Compared to execution times in [19], our system got a 49.42% relative improvement with respect to the ASR-aided system and an 81.36% relative improvement with respect to the only-acoustic system (although this result should be interpreted with caution since two different databases were used). It is also important to note that no external data is involved in any of the stages of the diarization. We do not have to train total variability matrices nor external UBMs on vast datasets.

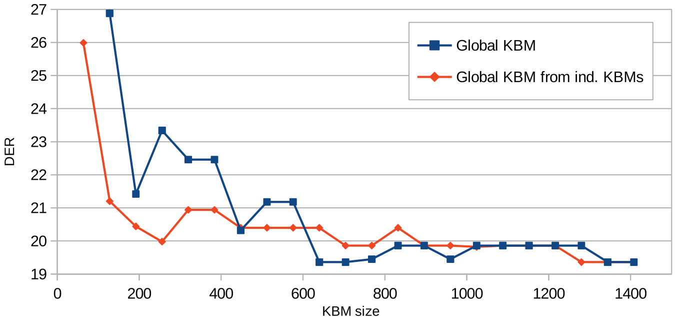

Fig. 5. Cross-show diarization performance for the two proposed methods for KBM estimation on the development set.

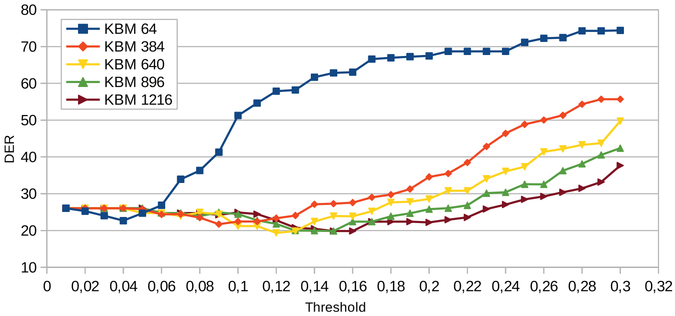

Fig. 6. Cross show diarization performance evolution by varying the ILP threshold δ for different KBM sizes on the development set.

C. *Cross-Show Speaker Diarization Experiments*

Fig. 5 shows cross-show diarization performance measured in DER, for the two proposed methods for global KBM estimation, using different KBM sizes and optimum ILP thresholds δ for each KBM size. It is observed that the KBM obtained from individual KBMs presents better performance with a small number of components, but finally the KBM trained globally on all test data prevails for KBM sizes of 640 and higher. One advantage of the "bootstrapped" KBM is that it is faster to compute, as the original Gaussian pool is smaller than the pool of Gaussians extracted from all the test set. In addition, the bootstrapped pool can be previously stored in the single diarization step, thus they can be reused. However, in our tests, computation of the global KBM is still fast for our purposes, and allows the use of smaller KBM sizes to reach the optimum output. The best DER Fig is 19.36%, obtained with 640 KBM components and $\delta = 0.12$.

In order to assess performance according to the ILP threshold, Fig. 6 shows results for different KBM sizes of 64, 384, 640, 896, and 1216 (using the global KBM estimated over all the test data). It can be observed that the optimum threshold deviates to the right when increasing the number of components in the KBM. We are interested in finding the minimum number of components in the KBM (the smaller the KBM, the faster the system) able to provide optimum performance. Therefore we select KBM size of 640 components, with $\delta = 0.12$ as the best configuration.

Table III presents cross-show diarization results using the best configuration selected above, with and without applying NAP compensation. We found in our experiments on the development data that the best number k of top eigenvectors used to calculate the NAP projection is $k = 6$. Note that values of δ, KBM size, and k have been calculated on the development set and that those values are then used for the test set as well. It can be seen that NAP is beneficial in all tests carried out, providing a slight performance improvement. However, performance measured on the test set suffers a considerable increase of around 6% absolute DER. It is also confirmed that the bootstrapped KBM is less accurate than the one obtained in all data for these values of KBM size and δ. A further analysis indicates that the optimum values of δ, k, and KBM size obtained on the development set are not the optimum ones for the test set. As an example, using a global KBM of 1664 components and $\delta = 0.2$ without NAP results in a DER value of 22.8%, against the 25.04% obtained using the optimum settings tuned on the development set.

TABLE III
CROSS-SHOW DIARIZATION PERFORMANCE WITH KBM SIZE OF 640, ILP THRESHOLD OF 0.12, WITH AND WITHOUT NAP SESSION VARIABILITY COMPENSATION, ON THE DEVELOPMENT AND TEST SETS

	Sing. BS	Global KBM		KBM from ind. KBMs	
Data set		w/o NAP	NAP	w/o NAP	NAP
DEV	15.15	19.36	**18.71**	20.40	**20.04**
TEST	18.17	25.04	**24.34**	27.59	**25.99**

TABLE IV
EXECUTION TIME OF CROSS-SHOW SPEAKER DIARIZATION IN xRT

Stage	**xRT**
Single diarization	0.0354
Training resources for Cross-show	0.0400
Cross-show clustering	0.0002
Total Cross-show	**0.0402**
Single + Cross	**0.0756**

Finally, Table IV shows execution time figures of the Cross-show diarization system using the optimum configuration found above (KBM size of 640, $\delta = 0.12$, NAP order of 6), broken down into the main stages involved. It is clear that most of the time is employed to prepare the resources required, including the global KBM, cluster BKs, and the within-class scatter matrix. This step presents a xRT of 0.04. After that costly step, clustering is very fast, showing a real-time factor of 0.0002. Therefore, the total xRT for the global clustering is 0.0402. If the execution time of the whole previous single-show is added, the final xRT is equal to 0.0756. We find this value still very reasonable to perform the whole single- plus cross-show diarization process.

D. *Comparison of Performance with the State-of-the-Art*

Table V provides results of several works performed over the REPERE Phase 1 test dataset. REPERE A, B and C teams refer to the participant systems in the official REPERE challenge [33]. Descriptions of the systems can be found in [34], [35] and [36]. Single-show performances of the three systems are very good and at the state of the art for this kind of data. DERs of cross-show diarization are very close to their single-show counterparts excepts for system A. Our single- and cross-show systems performances (last row of Table V) are around 3-4% absolute below the official results. We find this difference in performance reasonable, as the participant systems are significantly more complex (see systems' descriptions), involving in some cases supporting modules as overlapping speech detection and handling. The authors have had indications that the winner team (C) corresponds to the system described in [34],

TABLE V
SINGLE-SHOW AND CROSS-SHOW SPEAKER DIARIZATION PERFORMANCE RESULTS OF SEVERAL SYSTEMS IN THE LITERATURE, ON THE REPERE PHASE 1 TEST SET

System	Single DER (%)	Cros DER (%)
REPERE team A	13.70	33.09
REPERE team B	13.35	16.05
REPERE team C	**11.10**	**14.20**
[34] AHC/CLR	17.19	23.95
[34] ILP/i-vector	15.46	19.59
[31] AHC/CLR	16.22	-
[31] ILP/i-vector	14.60	-
Binary key	**15.15**	**18.71**

and that the result achieved in the REPERE evaluation involved the additional use of overlapping speech detection and speaker identification. The authors have also published results of their system in isolation [34], [31]. Such works give results of two different clustering methods: a GMM/CLR based clustering and an ILP/i-vector approach. It can be seen how the ILP clustering outperforms the AHC/CLR one in both tasks of single- and diarization. Our system performs very similarly to those, while probably being faster due to its simplicity. Note that systems described in [34] and [31] perform a BIC speaker segmentation and a first BIC clustering, followed by Viterbi re-segmentation, which are usually computationally expensive.

VII. CONCLUSIONS AND FUTURE WORK

This paper describes the last efforts made in the framework of binary key speaker diarization towards fast single- and cross-show speaker diarization. After giving a brief review of recent research in these lines, the baseline binary key diarization system was described and evaluated. Next, a series of improvements for single-show binary key speaker diarization and a new method for cross-show speaker diarization based on binary keys were proposed. The improvements were validated experimentally, demonstrating the effectiveness of the proposed modifications: although the baseline single-show speaker diarization system was already quite fast, speed was further increased. Now, the system is 28.2 times faster than real time, and presents $\mathrm{xRT} = 0.0354$ with the best configuration found. With regard to performance, the new proposed final clustering selection algorithm outperforms the original one, obtaining a relative improvement of 16.8%, an $\mathrm{DER} = 15.15\%$. Concerning binary key based cross-show diarization, our proposal is effective, getting global DER just above the single-show segmentation used as a starting point. Combining both single- and cross-show stages shows execution times still very competitive. Our system's performance is quite close to those achieved in the REPERE evaluation campaign, but surely exhibiting better execution times. To our knowledge, participating systems used complex combinations of technologies, including overlapping speech detection, not to mention the requirement of vast amounts of training and development data to estimate UBMs and total variability matrices. Our approach trains the required resources on the test data itself.

We think that binary key speaker diarization can be further improved. In the case of short audio excerpts, performance will probably benefit from using adaptive KBM parameters, such as KBM size, window length and overlap, with the belief that smaller audio files will not require so many components since the data to train them is limited. One more aspect concerns the data assignment algorithm. Using fixed-length segments of 1s duration does not allow to obtain very precise segment boundaries and speaker change points. We believe that using a Viterbi-like decoding in the binary key domain would undoubtedly result in better boundary precision, and consequently, in gains in accuracy. But this alternative decoding should be designed having high speed rates always in mind. We have settled a fast single-show diarization system which does not depend on external training data, suitable for further cross-show diarization. However, as the proposed cross-show scheme follows an hybrid approach, it can suffer from memory usage issues with big data collections. In this sense, an incremental system based on binary keys could be designed with likely success.

REFERENCES

[1] H. Jin, F. Kubala, and R. Schwartz, "Automatic speaker clustering," in *Proc. DARPA Speech Recognit. Workshop*, 1997, pp. 108–111.

[2] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA Speech Recognit. Workshop*, 1997, pp. 97–99.

[3] C. Wooters, J. Fung, B. Peskin, and X. Anguera, "Towards robust speaker segmentation: The ICSI-SRI fall 2004diarization system," in *Proc. RT-04F Workshop*, 2004.

[4] S. Meignier, J.-F. Bonastre, and S. Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," in *Proc. Odyssey Workshop*, 2001, pp. 175–180.

[5] C. Fredouille and G. Senay, "Technical improvements of the E-HMM based speaker diarization system for meeting records," in *MLMI*, ser. Lecture Notes in Computer Science, S. Renals, S. Bengio, and J. G. Fiscus, Eds. New York, NY, USA: Springer, 2006, vol. 4299, pp. 359–370.

[6] E. Fox, E. Sudderth, M. Jordan, and A. Willsky, "A sticky HDP-HMM with application to speaker diarization," *Ann. Appl. Statist.*, vol. 5, no. 2A, pp. 1020–1056, 2011.

[7] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 217–227, Jan. 2014.

[8] S. Shum, N. Dehak, R. Dehak, and J. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2015–2028, Oct. 2013.

[9] D. A. van Leeuwen, "Speaker linking in large data sets," in *Proc. Odyssey Workshop*, 2010, p. 35.

[10] H. Ghaemmaghami, D. Dean, R. Vogt, and S. Sridharan, "Extending the task of diarization to speaker attribution," in *Proc. Interspeech*, 2011, pp. 1049–1052.

[11] X. Anguera and J.-F. Bonastre, "Fast speaker diarization based on binary keys," in *Proc. ICASSP*, May 2011, pp. 4428–4431.

[12] X. Anguera and J.-F. Bonastre, "A novel speaker binary key derived from anchor models," in *Proc. Interspeech*, 2010, pp. 2118–2121.

[13] H. Delgado, C. Fredouille, and J. Serrano, "Towards a complete binary key system for the speaker diarization task," in *Proc. Interspeech*, 2014, pp. 572–576.

[14] H. Delgado, X. Anguera, C. Fredouille, and J. Serrano, "Global speaker clustering towards optimal stopping criterion in binary key speaker diarization," in *Proc. IberSPEECH*, 2014, pp. 59–68.

[15] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 356–370, Feb. 2012.

[16] Y. Huang, O. Vinyals, G. Friedland, C. Muller, N. Mirghafori, and C. Wooters, "A fast-match approach for robust, faster than real-time speaker diarization," in *Proc. ASRU*, Dec. 2007, pp. 693–698.

[17] D. Vijayasenan, F. Valente, and H. Bourlard, "An information theoretic approach to speaker diarization of meeting data," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 7, pp. 1382–1393, Sep. 2009.

[18] E. Gonina, G. Friedland, H. Cook, and K. Keutzer, "Fast speaker diarization using a high-level scripting language," in *Proc. ASRU*, Dec. 2011, pp. 553–558.

[19] J. Silovsky, J. Zdansky, J. Nouza, P. Cerva, and J. Prazak, "Incorporation of the ASR output in speaker segmentation and clustering within the task of speaker diarization of broadcast streams," in *Proc. IEEE MMSP*, 2012, pp. 118–123.

[20] H. Ghaemmaghami, D. Dean, and S. Sridharan, "A cluster-voting approach for speaker diarization and linking of Australian broadcast news recordings," in *Proc. ICASSP*, 2015.

[21] Q. Yang, Q. Jin, and T. Schultz, "Investigation of cross-show speaker diarization," in *Proc. Interspeech*, 2011, pp. 2925–2928.

[22] V.-A. Tran, V. B. Le, C. Barras, and L. Lamel, "Comparing multi-stage approaches for cross-show speaker diarization," in *Proc. Interspeech*, 2011, pp. 1053–1056.

[23] G. Dupuy, S. Meignier, and Y. Estève, "Is incremental cross-show speaker diarization efficient for processing large volumes of data?," in *Proc. Interspeech*, 2014, pp. 587–591.

[24] G. Dupuy, M. Rouvier, S. Meignier, and Y. Estéve, "I-vectors and ILP clustering adapted to cross-show speaker diarization," in *Proc. Interspeech*, 2012.

[25] T. H. Nguyen, E. Chng, and H. Li, "T-test distance and clustering criterion for speaker diarization," in *Proc. Interspeech*, 2008.

[26] O. Galibert, "Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech," in *Proc. Interspeech*, 2013, pp. 1131–1134.

[27] J. Duchi, "Derivations for linear algebra and optimization," [Online]. Available: http://www.cs.berkeley.edu/jduchi/projects/general_notes.pdf

[28] G. Hernandez-Sierra, J. R. Calvo, J.-F. Bonastre, and P.-M. Bousquet, "Session compensation using binary speech representation for speaker recognition," *Pattern Recognit. Lett.*, vol. 49, no. 0, pp. 17–23, 2014.

[29] A. Solomonoff, C. Quillen, and W. M. Campbell, "Channel compensation for SVM speaker recognition," in *Proc. Odyssey Workshop*, 2004, pp. 57–62.

[30] M. Rouvier and S. Meignier, "A global optimization framework for speaker diarization," in *Proc. Odyssey Workshop*, 2012.

[31] G. Dupuy, S. Meignier, P. Deléglise, and Y. Estève, "Recent improvements on ILP-based clustering for broadcast news speaker diarization," in *Proc. Odyssey Workshop*, Joensuu, Finland, 2014.

[32] A. Giraudel, M. Carré, V. Mapelli, J. Kahn, O. Galibert, and L. Quintard, "The REPERE corpus: A multimodal corpus for person recognition," in *Proc. LREC*, Istanbul, Turkey, May 2012.

[33] O. Galibert and J. Kahn, "The first official REPERE evaluation," in *Proc. SLAM@Interspeech*, 2013.

[34] 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 *Proc. Interspeech*, Aug. 2013.

[35] C. Barras, X. Zhu, S. Meignier, and J. Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1505–1512, Sep. 2006.

[36] D. Charlet, C. Barras, and J.-S. Lienard, "Impact of overlapping speech detection on speaker diarization for broadcast news and debates," in *Proc. ICASSP*, May 2013, pp. 7707–7711.

Héctor Delgado received the Computer Science Engineering degree from Universidad de Sevilla, Spain, in 2008, and the Multimedia Technologies Master degree from Universitat Autnoma de Barcelona (UAB), Spain, in 2009. He is currently pursuing the Ph.D. degree from UAB. From 2008 to 2015, he was with the Center for Ambient Intelligence and Accessibility of Catalonia (CAIAC), at UAB, contributing to R+D projects related to speech technologies. In 2013, he was with the Laboratoire Informatique d'Avignon (LIA), University of Avignon, France, as a visiting researcher. His current research interests include speaker diarization and recognition, audio segmentation, speech recognition, and automatic audio indexing.

Xavier Anguera obtained his M.S. in electrical engineering at the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 2001, when he also graduated from an European Masters in Language and Speech. In 2006, he obtained his Ph.D. also from UPC University, on speech processing. From 2001 to 2003, he worked for Panasonic Speech Technology Lab in Santa Barbara, CA. From 2004 to 2006, he was a Visiting Researcher at the International Computer Science Institute (ICSI) in Berkeley, CA. Since 2007, he has been with Telefonica Research in Barcelona, pursuing research on multimedia analysis. He has coauthored over 80 academic publications and is the coinventor of several patents. His current research interests include speech/speaker processing and multimedia analysis.

Corinne Fredouille received the Ph.D. degree from the Laboratoire Informatique d'Avignon (LIA), University of Avignon, France, in 2000. She was appointed as an Assistant Professor at LIA in 2003. Her research interests include acoustic analysis, voice quality assessment, statistical modeling, automatic speaker recognition, speaker diarization and, more recently, speech and voice disorder assessment and acoustic-based characterization. She has participated in several national and international speaker diarization and recognition system evaluation campaigns and has published over 20 research papers in this field.

Dr. Fredouille is a member of the International Speech Communication Association (ISCA) and of the French speaking communication association (AFCP), Special Interest Group (SIG) of ISCA, for which she was in charge of the secretariat for six years, from 2008 to 2013.

Javier Serrano graduated in computer science in 1988, and received a Ph.D. (1994) degree in automatic control (computer science program), at Universtat Autónoma de Barcelona (UAB), Spain. Since 1991, he has been an Associate Professor at the Computer Science Department of UAB. In 2000, he joined the new Communications and System Engineering Department. In 2006, he joined the Center for Ambient Intelligence and Accessibility of Catalonia (CAIAC), a research center and a technology transfer node from the Catalan IT network, as Head of Research and Innovation. His main research interests are dialogue systems in human-centric interfaces and speech recognition and understanding facing meta-data extraction from multimedia contents.