Global Speaker Clustering towards Optimal Stopping Criterion in Binary Key Speaker Diarization

Héctor Delgado[1], Xavier Anguera[2], Corinne Fredouille[3], and Javier Serrano[1]

[1] CAIAC, Autonomous University of Barcelona, Barcelona, Spain
[2] Telefonica Research, Barcelona, Spain
[3] University of Avignon, CERI/LIA, France
{hector.delgado,javier.serrano}@uab.cat, xanguera@tid.es, corinne.fredouille@univ-avignon.fr

Abstract. The recently proposed speaker diarization technique based on binary keys provides a very fast alternative to state-of-the-art systems with little increase of Diarization Error Rate (DER). Although the approach shows great potential, it also presents issues, mainly in the stopping criterion. Therefore, exploring alternative clustering/stopping criterion approaches is needed. Recently some works have addressed the speaker clustering as a global optimization problem in order to tackle the intrinsic issues of the Agglomerative Hierarchical Clustering (AHC) (mainly the local-maximum-based decision making). This paper aims at adapting and applying this new framework to the binary key diarization system. In addition, an analysis of cluster purity across the AHC iterations is done using reference speaker ground-truth labels to select the purer clustering as input for the global framework. Experiments on the REPERE phase 1 test database show improvements of around 6% absolute DER compared to the baseline system output.

Keywords: speaker diarization, binary key, ILP, cluster purity.

1 Introduction

Speaker diarization is the task of segmenting an audio file into speaker homogeneous segments. It is well known the importance of speaker diarization as a pre-processing tool for many speech-related tasks which take advantage of dealing with speech signals from a single-speaker. For instance, speech recognition can benefit of speaker diarization to adapt acoustic models to target speakers. Furthermore, searching speech utterances spoken by target speakers within big audiovisual content repositories is increasingly becoming very popular and challenging. Before identifying such speakers by means of speaker identification technology, they must be previously separated adequately. Here, speaker diarization systems should be accurate and fast enough in order to process big quantities of data in a reasonable time period.

J.L. Navarro Mesa et al. (Eds.): IberSPEECH 2014, LNAI 8854, pp. 59–68, 2014.

Most state-of-the-art systems perform a combination of Gaussian Mixture Model (GMM) as speaker models, Bayesian Information Criterion (BIC) as a measure for cluster merging and stopping criterion, and Viterbi decoding for data assignment. All the mentioned algorithms are applied iteratively, imposing a high computational load which results in too long processing times [1] (above 1xRT, being xRT the Real Time factor) for some real-life applications.

The recently proposed speaker diarization approach based on binary key speaker modeling [1] provides a fast system (over 10 times faster than real time) with little performance decrease. DER scores of around 27% with a real time factor of 0.103 xRT were reported using all the NIST RT databases. This technique provides a fast alternative to the use of parallel computing, but using a single CPU. Later, the approach was extended and successfully applied to process TV broadcast audio in [2].

Both works report on the weakness of the stopping criterion being used, which usually does not provide the optimum clustering in terms of DER. Indeed, [2] demonstrates that the diarization system is able to produce better clusterings than the one returned by the stopping criterion. This indicates that improving stopping criterion will systematically produce a gain in performance.

Lately, a global optimization framework to speaker clustering was introduced in [7]. Contrary to classic AHC, the framework tries to find the optimum clustering in a global way, instead of relying on greedy, local-maximum-made decisions as AHC does. Given the weakness of the optimum clustering selection algorithm used in the binary key speaker diarization system, it seems reasonable to think that such an approach, which is able to implicitly determine the optimum number of clusters, can provide an effective alternative to the faulty stopping criterion.

This work follows this direction in order to evaluate the effectiveness of the global clustering technique integrated in the binary key speaker diarization system. First, the approach is adapted to be used in our case. Second, an analysis of cluster purity of the binary key system is performed. And third, the global clustering approach is tested in our system by using the extracted result of the first analysis. Preliminary results show that the global clustering approach outperforms the clustering originally returned by the system stopping criterion. However, it also suffers some robustness issues among test audio files since the global clustering parameters need to be tuned for each input file.

The paper is structured as follows: Section 2 describes the baseline binary key speaker diarization system. Section 3 gives an explanation of the global speaker clustering and proposes an adaptation suitable for the binary key system. Section 4 describes the experimental setup and results. Section 5 concludes and proposes future work.

2 Overview of the Binary Key Speaker Diarization System

The implementation of the binary key diarization system used in this work is described in [2]. First, an acoustic processing block aims at transforming the

acoustic input data into a suitable binary representation. Secondly, the binary processing block takes the binary data from the previous stage to perform an Agglomerative Hierarchical Clustering (AHC) but, unlike the classic approach, all operations are performed in the binary domain. This results in a significant gain in execution time, compared with state-of-the-art agglomerative systems.

As said above, the acoustic processing block transforms the acoustic feature vectors into binary vectors called binary keys. The key element for this transformation is a UBM-like acoustic model, called KBM (binary Key Background Model), which is trained using the own test input data, but in a particular way. A single Gaussian is trained every n seconds (with some overlap), so that at the end a pool of several hundreds of Gaussians is obtained. Proceeding in this way, it is guaranteed that the overall acoustic space of speakers is covered by the pool of Gaussians. The next step consists in taking a subset of N components from the pool so that the selected Gaussians are as complementary and discriminant between them as possible. To achieve that, the Gaussians are selected iteratively by calculating the KL2 (symmetric Kullback-Leibler) divergence between the already selected components and the remaining ones, and the most dissimilar component is selected. The process is repeated until having N components.

Once the KBM is trained, any set or sequence of input feature vectors can be converted into a Binary Key (BK). A BK $\mathrm{v}_f = \{v_f[1], ..., v_f[N]\}, v_f[i] = \{0, 1\}$ 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 ith Gaussian of the KBM coexists in the same area of the acoustic space as the acoustic data being modeled. The BK can be obtained in two steps. Firstly, for each feature vector, the best N_G matching Gaussians in the KBM are selected (i.e., the N_G Gaussians which provide higher likelihood for the given feature), and their identifiers are stored. Secondly, for each component, the count of how many times it has been selected as a top component along all the features is calculated. Then, the final BK is obtained by setting to 1 the positions corresponding to the top M Gaussians at the whole feature set level, (i.e., the Mth most selected components for the given feature set). 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.

The last step before switching to the binary process block is the clustering initialization. This is done at the acoustic level in order to have an initial rough clustering as a starting point. Taking advantage of the KBM trained before, an initial set of N_{init} clusters is build by using the first N_{init}th Gaussians in the KBM. The input data are divided into small segments (e.g., 100ms) and they are assigned to the cluster whose Gaussian provides the highest likelihood.

The binary block implements an AHC clustering approach. However, all operations are done with binary data, which makes the process much faster than with classic GMM-based approaches. First, BKs for the initial clusters are calculated using the method explained in section 2. Then, the input data are reassigned to the current clusters. Data are first divided into fixed length segments and BKs are calculated for all them. Note that these BKs keys will be used along the

iterations of the AHC, so they can be stored and reused. Next, the segments are assigned by comparing their BKs with all current cluster BKs. The similarity metric is given by 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 $\wedge$ indicates the boolean AND operator, and $\vee$ indicates the boolean OR operator. This is a very fast, bit-wise operation between two binary vectors.

Once data are redistributed, BKs are trained for the new clusters. Finally, similarities between all cluster pairs are obtained using equation 1 and the cluster pair with the highest score is merged, reducing the number of clusters by one.

The iterative process is repeated until a single cluster is reached, storing all the partial clusterings. At the end of the process, the final clustering is output by using a modification of the T-test T_S metric proposed in [6]. After the computation of intra-cluster and inter-cluster similarity distributions between segments for each clustering C^i, the selected clustering is the one which maximizes T_S, given by equation 2.

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

where m_1 , σ_1 , n_1 , m_2 , σ_2 and n_2 are the mean, standard deviation and size of intra-cluster and inter-cluster distance distributions, respectively.

3 Global Speaker Clustering

As it has been reported in [1], the final clustering selection (i.e., the stopping criterion) based on the T-test distance does not return the optimum clustering (differences in performance of around 7-8% absolute DER with the REPERE database [2]).

Recently an alternative approach to the classic AHC was presented in [7]. The main argument against AHC is the greedy nature of the technique, which uses local optimums to decide which cluster pair should be merged in each iteration. If an erroneous merging is produced, the error will likely be propagated through the iterations, resulting in impure clusters and, consequently, in loss of performance. Following these thoughts, the proposed alternative clustering method addresses the clustering as a global process, reformulated as a problem of Integer Linear Programming (ILP), in order to minimize a certain objective function, subject to a set of constraints, in a global manner.

The authors of [7] propose to apply this global clustering method immediately after a first BIC-based AHC stage. At this point, it is assumed that the resulting clusters are pure, i.e., each cluster contains speech from a single speaker. However, more than one cluster may refer to a given speaker. This can occur because a given speaker who is speaking over different acoustic conditions (e.g. background music, background noise) may be modeled by different clusters by the

system. It is at this point where the ILP clustering can be used to obtain a final clustering where the several clusters referring to the same speaker are merged together in a single cluster. Additionally, to deal with channel variability, each input cluster is represented by an i-vector. Thus, given an input clustering of N clusters, a set of N i-vectors is obtained. From here on, the clusters are treated as single points.

Given the N points, the goal is to group them into K clusters while minimizing the objective function and meeting the constraints (refer to section 3.1). 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. In the end, there will be as many clusters as 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).

3.1 Adaptation of ILP Clustering to the Binary Key Diarization System

In order to adapt the technique to our framework, some modifications are proposed. First, the points of the problem will be BKs instead of i-vectors. Second, unlike the original work, no channel compensation is applied.

The ILP clustering formulation has been adapted to our framework (refer to [3] for the original formulation), and it is defined as:

Minimize

$$\sum_{k=1}^{N} x_{k,k} - \frac{1}{D} \sum_{k=1}^{N} \sum_{j=1}^{N} d(k,j) x_{k,j} \tag{3}$$

Subject to

$$x_{k,j} \in \{0,1\} \qquad \forall k, \forall j \tag{4}$$

$$\sum_{k=1}^{N} x_{k,j} = 1 \qquad \forall j \tag{5}$$

$$d(k,j) x_{k,j} \leq \delta \qquad \forall k, \forall j \tag{6}$$

Eq. 3 is the objective function to be minimized. As said above, the aim is to minimize the number of clusters and the dispersion of the BKs within each cluster. The binary variable $x_{k,k}$ is equal to 1 if the BK k is a center. The distance $d(k,j)$ between BKs k and j is calculated as $1 - S(k,j)$, where $S(k,j)$ is given by eq. 1 (section 2). 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 BK j is associated with center k. Eq. 5 ensures that each BK j is associated with only one center k. Finally, each BK j associated with a center k must have a distance shorter than a threshold.

The proposed ILP clustering requires an input clustering to start the process. This input clustering will be the result of applying a given number of iterations of the AHC binary key diarization system. Each cluster will be represented as a BK, extracted following the method for BK computation explained in section 2. Ideally, the input clusters should be as pure as possible, since the ILP clustering method is not able to re-allocate misclassified data, so the errors will be propagated to the resulting clusters. For this reason, an analysis of cluster purity across the AHC iterations is conducted previous to the application of the global clustering (section 4.2).

4 Experiments and Results

This section describes experimental setup and results for two different experiments. Firstly, a study of cluster purity across the iterations of the baseline AHC is performed. Secondly, the resulting purest clusterings from the first experiment are taken to be used as input clusters to test the ILP global clustering.

As the aim of this work is mainly to analyze speaker clustering and stopping criterion, it has been decided to use perfect SAD labels. That is, the speaker ground-truth labels have been used to extract the speech activity and to discard nonspeech content. In this way, the analysis can focus on speaker clustering without the effects of additional noise (false alarm speech) and not loosing useful speaker time (miss speech).

Both tests are evaluated on the REPERE phase 1 test dataset of TV data [4]. This database was developed in the context of the REPERE Challenge [5]. It consists of a set of TV shows from several French TV channels.

Previous to experiment descriptions, the experimental setup is explained in the next subsection.

4.1 Experimental Setup

Parameters and settings of the various modules of the binary key speaker diarization system are described here.

Regarding audio processing, the provided single channel is used without further treatment. Next, feature extraction is performed. Standard 19-order MFCCs are computed using a 25ms window every 10ms.

For training the KBM, single Gaussian components are obtained using a 2s window in order to have sufficient data for parameter estimate. Window rate is set according to the input audio length, in order to obtain an initial pool of 2000 Gaussians. Then, 896 components are selected to conform the final KBM following the method described in section 2.

With regard to binary key estimate parameters, the top 5 Gaussian components are taken in a frame basis, and the top 20% components at segment level.

The clustering initialization is done by using the first N_{init} Gaussian components in the KBM as cluster models. Two different values of N_{init} are tested in

the experiments: 25 and 50. Then, 100ms segments are assigned to the different clusters to obtain the first rough, over-segmented clustering.

Finally, in the AHC stage, BKs keys are computed for each 1s segment, augmenting it 1s before and after, totaling 3s.

In order to evaluate performance, the output labels are compared with the reference ones to compute the DER. Since the proposed system does not handle overlap speech, regions with more than one active speakers are ignored in the score computation (note that this is only for evaluation, so that overlapped speech regions are included during the complete diarization process). In addition, as perfect SAD is being used and overlap speech is not being evaluated, false alarm and miss errors are virtually equal to zero, so the analysis can focus only on speaker errors.

4.2 Search of the Purest Clustering

The aim of this analysis is to study the evolution of the cluster purities among the iterations of the diarization system. By using the reference speaker labels, one can determine how much speaker time in the cluster belongs to the different speakers in the reference. In a given cluster there is always a majority speaker, who is the one with most speaker time within the cluster. Considering this speaker as the "main" speaker, the cluster purity can be calculated as the ratio between the cluster time assigned to the main speaker and the total cluster time. However, purity of clusters of different sizes does not affect, globally speaking, in the same way to the system. Due to this fact, the calculation of a time-weighted purity measure is proposed instead by taking into account cluster sizes. The final time-weighted cluster purity is calculated as the cluster purity multiplied by the cluster length, and divided by the total duration of the test audio (after removing nonspeech content). Finally, the time-weighted purity for a whole clustering can be obtained as the average of the time-weighted purity of all clusters in the clustering.

Normally, the purity should start to increase after a few iterations of AHC and will start to decrease when the number of clusters is lower than the actual number of speakers. In table 1, clustering purity is shown for two different clusterings: the one providing highest purity ("highest purity columns") and the one producing lowest DER ("sysOut purity" columns). The experiment is repeated for 25 and 50 initial clusters (N_{init}). Generally, purities reach the optimum in early iterations of the AHC (with a number of clusters significantly higher than the optimum clustering), although the exact iteration is showed to be quite dependent on the show. In addition, optimum purities are higher in the case of 50 initial clusters compared to 25 initial clusters. With regard to the system output, as it could be expected, purity is, in general, inversely proportional to DER. Finally, overall DER of system output with $N_{init} = 25$ (9.47%) is slightly lower than for $N_{init} = 50$ (10.60%). It seems that each of the two configurations works better for certain shows.

Table 1. Results of cluster purity analysis broken down into shows. #spk is the number of actual speaker of each show. N_{init} indicates the number of initial clusters. Column "highest purity" shows purity and number of clusters (#C) of the optimum clusterings in terms of time-weighted overall purity, whilst column "sysOut purity" shows purity and number of clusters of the optimum clusterings in terms of DER.

Show ID	#spk	N_init = 25						N_init = 50					
		Highest purity			SysOut purity			Highest purity			SysOut purity		
		#C	Purity	DER	#C	Purity	DER	#C	Purity	DER	#C	Purity	DER
BFMTV_BFMStory_1	6	19	0.910	24.60	8	0.883	4.37	45	0.923	51.49	6	0.865	6.42
BFMTV_BFMStory_2	18	18	0.891	18.43	18	0.891	18.43	41	0.941	33.32	31	0.913	21.00
BFMTV_BFMStory_3	10	19	0.951	29.77	13	0.937	7.25	35	0.962	33.19	11	0.915	5.58
BFMTV_BFMStory_4	6	11	0.962	11.05	6	0.952	1.59	20	0.963	14.82	6	0.950	1.78
BFMTV_CultureEtVous_1	5	14	0.950	52.13	4	0.891	10.11	21	0.970	52.04	3	0.881	8.88
BFMTV_CultureEtVous_2	6	10	0.925	27.16	4	0.776	21.09	23	0.956	43.30	4	0.780	20.63
BFMTV_CultureEtVous_3	16	22	0.904	63.80	5	0.744	24.44	29	0.851	62.16	4	0.702	23.68
BFMTV_CultureEtVous_4	9	22	0.890	54.37	2	0.640	33.07	41	0.902	70.80	3	0.650	30.63
BFMTV_CultureEtVous_5	6	21	0.905	72.06	3	0.823	9.90	20	0.917	65.02	3	0.823	9.90
BFMTV_CultureEtVous_6	12	18	0.870	52.70	5	0.700	25.37	29	0.890	57.79	5	0.720	21.15
BFMTV_CultureEtVous_7	14	18	0.839	43.19	6	0.701	31.22	34	0.850	68.51	9	0.758	26.67
LCP_CaVousRegarde_1	7	23	0.925	70.01	4	0.823	15.46	48	0.957	82.51	6	0.883	12.11
LCP_CaVousRegarde_2	5	7	0.938	8.00	4	0.903	3.11	14	0.951	9.28	4	0.917	2.64
LCP_CaVousRegarde_3	5	21	0.950	40.21	6	0.836	19.28	37	0.950	55.75	15	0.886	21.94
LCP_EntreLesLignes_1	5	15	0.919	24.07	10	0.909	11.34	19	0.958	19.64	19	0.958	19.64
LCP_EntreLesLignes_2	5	27	0.932	28.44	6	0.891	4.92	26	0.949	24.37	6	0.891	4.44
LCP_EntreLesLignes_3	5	15	0.945	29.45	3	0.823	14.74	26	0.935	29.45	3	0.823	14.74
LCP_LCPInfo13h30_1	16	21	0.890	23.69	14	0.882	10.04	39	0.938	26.83	20	0.902	13.19
LCP_LCPInfo13h30_2	12	18	0.951	16.77	11	0.890	10.87	41	0.953	34.64	23	0.918	17.40
LCP_LCPInfo13h30_3	10	13	0.905	24.55	7	0.871	12.28	27	0.921	24.67	11	0.841	17.10
LCP_PileEtFace_1	3	14	0.921	25.17	3	0.821	8.24	16	0.955	19.52	6	0.921	10.91
LCP_PileEtFace_2	3	15	0.954	29.25	3	0.864	8.11	31	0.960	72.58	2	0.853	8.28
LCP_PileEtFace_3	3	9	0.910	11.18	3	0.797	6.89	24	0.932	37.36	3	0.808	6.89
LCP_PileEtFace_4	3	7	1.000	6.81	6	0.988	3.95	17	1.000	21.56	8	0.988	7.59
LCP_PileEtFace_5	3	18	0.936	53.94	3	0.912	4.20	4	0.936	3.67	4	0.936	3.67
LCP_TopQuestions_1	8	22	0.987	35.89	8	0.976	1.36	12	0.989	7.42	9	0.981	2.11
LCP_TopQuestions_2	5	15	0.985	23.41	3	0.914	8.13	20	0.985	29.11	6	0.959	4.09
LCP_TopQuestions_3	6	11	0.973	21.53	5	0.948	5.17	14	0.973	13.34	5	0.948	5.17
Overall	-	-	-	-	-	-	9.47	-	-		-	-	10.60

4.3 Experiments on ILP Clustering

As stated above, the ILP clustering needs an input set of clusters to work with. Ideally, this input clustering should be as pure as possible, as the technique is not able to recover incorrectly assigned speech. The previous experiment shows that the exact number of iterations to get the purest clustering is dependent on the show. This fact results in a lack of robustness among different audio data. To avoid this issue, in this experiment the clusterings with highest purity of the previous experiment have been selected as input for the current one.

Figure 1 depicts DER of the ILP clustering in function of the threshold θ. For comparison purposes, DER of the baseline system output and optimum clusterings are also plotted. For the initial clustering obtained with $N_{init} = 25$, optimum DER is obtained for $\theta = 0.63$, while for the one of $N_{init} = 50$ is obtained for $\theta = 0.66$. Although the previous analysis shows that the average cluster purity for the case of $N_{init} = 50$ is higher, DER of clustering obtained from $N_{init} = 25$ is slightly lower (15.1% versus 16,26%). This may be due to the higher number of clusters to be merged. As it can be seen, the ILP method outperforms the baseline system with T-test stopping criterion for a range of values of θ, obtaining a gain of around 6% absolute DER with the best configuration. However, performance still

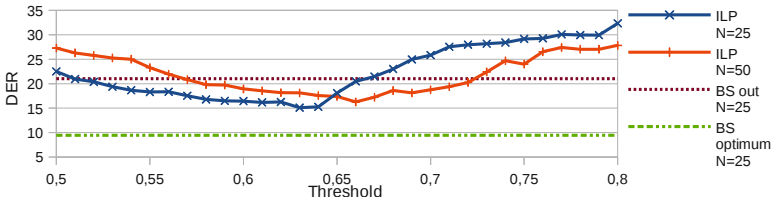

Fig. 1. Overall DER trend of the ILP clustering while varying the threshold θ for N_{init} equal to 25 and 50. Overall DER of the baseline system output (BS out) and optimum clusterings (BS optimum) are also provided for comparison.

Table 2. Results of ILP clustering experiments using the purest clusterings from table 1 for N_{init} equal to 25 and 50 as inputs. For each show, values of the optimum threshold θ_{opt}, resulting number of clusters #C, and DER are shown. The actual number of speakers per show #spk is also provided.

Show ID	#spk	N_init = 25			N_init = 50		
		θ_{opt}	#C	DER	θ_{opt}	#C	DER
BFMTV_BFMStory_1	6	0.63	5	4.52	0.78	6	7.17
BFMTV_BFMStory_2	18	0.50	18	18.17	0.61	24	21.44
BFMTV_BFMStory_3	10	0.57	11	6.42	0.63	11	4.51
BFMTV_BFMStory_4	6	0.62	7	1.78	0.65	7	1.69
BFMTV_CultureEtVous_1	5	0.77	2	18.57	0.82	3	20.35
BFMTV_CultureEtVous_2	6	0.81	3	13.79	0.73	5	17.42
BFMTV_CultureEtVous_3	16	0.77	4	30.31	0.77	6	55.28
BFMTV_CultureEtVous_4	9	0.83	4	39.64	0.89	14	30.56
BFMTV_CultureEtVous_5	6	0.77	4	38.63	0.75	4	33.24
BFMTV_CultureEtVous_6	12	0.76	4	25.74	0.85	4	22.68
BFMTV_CultureEtVous_7	14	0.77	4	29.73	0.82	4	26.48
LCP_CaVousRegarde_1	7	0.69	4	9.61	0.79	5	12.05
LCP_CaVousRegarde_2	5	0.63	4	3.11	0.73	4	2.60
LCP_CaVousRegarde_3	5	0.69	9	19.58	0.74	7	19.31
LCP_EntreLesLignes_1	5	0.63	11	10.08	0.50	19	19.64
LCP_EntreLesLignes_2	5	0.68	5	3.41	0.75	4	9.27
LCP_EntreLesLignes_3	5	0.77	3	15.39	0.76	3	15.00
LCP_LCPInfo13h30_1	16	0.52	15	10.07	0.65	13	10.42
LCP_LCPInfo13h30_2	12	0.53	11	15.26	0.56	22	11.40
LCP_LCPInfo13h30_3	10	0.57	6	13.76	0.57	17	13.59
LCP_PileEtFace_1	3	0.77	2	12.28	0.76	4	10.32
LCP_PileEtFace_2	3	0.74	3	7.15	0.77	2	8.28
LCP_PileEtFace_3	3	0.70	5	8.22	0.73	4	8.82
LCP_PileEtFace_4	3	0.63	6	2.94	0.77	3	1.35
LCP_PileEtFace_5	3	0.74	2	4.98	0.50	4	3.67
LCP_TopQuestions_1	8	0.63	8	1.60	0.63	9	1.01
LCP_TopQuestions_2	5	0.72	4	2.89	0.77	4	2.36
LCP_TopQuestions_3	6	0.66	5	2.79	0.64	7	4.30
Overall	-	-	-	9.57	-	-	10.20

does not reach that of the optimum clustering of the baseline system manually selected(5.63% absolute higher).

In order to demonstrate the dependence of threshold θ on the show, additional results are provided in table 2. Here, DER is shown for the optimum value of θ for each show. Obtained results are quite similar to the ones of the baseline system with optimum clustering manually selected. The reading of these results could be twofold. On one hand, the technique presents a lack of robustness since the number of iterations and threshold must be tuned for each input audio file. On the other hand, even if the threshold is not tuned in a per-show basis, the proposed adaptation of the ILP clustering outperforms the baseline system stopping criterion.

5 Conclusions and Future Work

This work focuses on the exploration of alternative methods to the stopping criterion of the binary key speaker diarization approach presented in [2]. The recently presented global framework for speaker clustering is a candidate to solve this drawback, as the technique implicitly estimates the optimum number of clusters. As this approach needs an input clustering as pure as possible, an analysis of cluster purity of the binary key AHC approach was carried out in order to select the optimum clustering in terms of purity. Then, the original ILP framework was adapted to our needs by replacing the i-vector with the binary key and was tested and compared with the baseline system on the REPERE phase 1 test database. Experiment results show an improvement of performance with respect to the baseline system, but also present some robustness issues according to the audio file being processed. It is thought that an in-depth analysis of the relation between system parameters (number of previous AHC iterations, threshold) and audio nature (audio length, number of speakers) could lead to some guidelines in order to tune system parameters for optimizing the system to the input audio. Finally, DER rates are still high and applying some kind of channel compensation could help to improve performance.

Acknowledgements. This work is part of the project "Linguistic and sensorial accessibility: technologies for voiceover and audio description", funded by the Spanish Ministerio de Economía y Competitividad (FFI2012-31023). This work was partially done within the French Research program ANR Project PERCOL (ANR 2010-CORD-102). This article is supported by the Catalan Government Grant Agency Ref. 2014SGR027.

References

1. Anguera, X., Bonastre, J.F.: Fast speaker diarization based on binary keys. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4428–4431 (May 2011)
2. Delgado, H., Fredouille, C., Serrano, J.: Towards a complete binary key system for the speaker diarization task. In: INTERSPEECH (2014)
3. Dupuy, G., Rouvier, M., Meignier, S., Estéve, Y.: I-vectors and ILP clustering adapted to cross-show speaker diarization. In: INTERSPEECH (2012)
4. Giraudel, A., Carré, M., Mapelli, V., Kahn, J., Galibert, O., Quintard, L.: The REPERE Corpus: a multimodal corpus for person recognition. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey (May 2012)
5. Kahn, J., Galibert, O., Quintard, L., Carre, M., Giraudel, A., Joly, P.: A presentation of the REPERE challenge. In: 2012 10th International Workshop on Content-Based Multimedia Indexing (CBMI), pp. 1–6 (June 2012)
6. Nguyen, T.H., Chng, E., Li, H.: T-test distance and clustering criterion for speaker diarization. In: INTERSPEECH (2008)
7. Rouvier, M., Meignier, S.: A global optimization framework for speaker diarization. In: ODYSSEY (2012)