Constrained speaker diarization of TV series based on visual patterns

Xavier Bost, Georges Linares

CONSTRAINED SPEAKER DIARIZATION OF TV SERIES BASED ON VISUAL PATTERNS

*Xavier Bost, Georges Linarès**

LIA, University of Avignon, 339 chemin des Meinajariès, 84000 Avignon, France

ABSTRACT

Speaker diarization, usually denoted as the "who spoke when" task, turns out to be particularly challenging when applied to fictional films, where many characters talk in various acoustic conditions (background music, sound effects...). Despite this acoustic variability, such movies exhibit specific visual patterns in the dialogue scenes. In this paper, we introduce a two-step method to achieve speaker diarization in TV series: a speaker diarization is first performed locally in the scenes detected as dialogues; then, the hypothesized local speakers are merged in a second agglomerative clustering process, with the constraint that speakers locally hypothesized to be distinct must not be assigned to the same cluster. The performances of our approach are compared to those obtained by standard speaker diarization tools applied to the same data.

***Index Terms*—** Speaker diarization, video structuration, agglomerative clustering

Cite as:
X. Bost & G. Linarès.
Constrained speaker diarization of TV series based on visual patterns.
2014 IEEE Spoken Language Technology Workshop (SLT).
doi: 10.1109/SLT.2014.7078606

1. INTRODUCTION

Speaker diarization is defined as the task of assigning the utterances contained in a spoken document to their respective speakers. For this purpose, two steps are involved, either sequentially or in conjunction: detection of change points between speakers; clustering of the resulting spoken segments in order to assign to the same speaker its own utterances. The clustering step is usually achieved in an iterative hierarchical process, either by agglomerating the closest spoken segments into the same cluster in a bottom-up strategy or by splitting the whole stream of utterances into smaller clusters in a top-down way.

This task is usually performed as an unsupervised one, in particular without allowing any prior knowledge of the number of speakers. This lack of information makes the stop condition of the hierarchical clustering process quite critical.

Speaker diarization (SD) systems were first developed for processing of audio-only streams in adverse – but controlled – acoustic conditions, such as telephone conversations, broadcast news, meetings... Some recent works applied them to videos whose production context is uncontrolled, facing difficulties due to content and environment variabilities.

In [1], the authors apply standard SD tools to the audio source of various kinds of video documents. The reported results exhibit a Diarization Error Rate (DER) much higher than for those classical application fields. The most dramatic decrease in performance is observed when the SD systems are applied to cartoons and movie trailers: among the possible reasons involved, the authors notice the high number of speakers implied in these kinds of stream, as well as the high variability of the acoustic environment (speech and music segments overlapping each other, sound effects). Moreover, as in most of previous related works on audiovisual SD, diarization problem is here addressed by applying audio-only systems to the audio channel of videos, without any integration of the video-related features that could help the diarization system.

However, some recent works focus on multimodal approaches for performing speaker segmentation of video streams: in [2], the authors evaluate a method based on early fusion of audio and video GMMs, and a classical BIC-based agglomerative process on the resulting two-channel information stream. This technique is evaluated on the AMI corpus [3] that consists of audiovisual recordings of four participants playing roles in a meeting scenario.

In this paper, we are interested in diarization of TV series, as a major basic part of a wider project aiming at automatic structuring of fictional videos.

Applying a SD system to TV series, where the speaker number is generally higher than in full-length movies, may thus be expected to be quite challenging. Nevertheless, fictional films exhibit formal regularities at a visual level. For instance, dialogue scenes require that the "180-degree" convention be respected in order to preserve the visual fluidity of the exchange: so that both speakers seem to look at each other when they appear successively on the screen, the first one must look right and the second one must look left, resulting in keeping the two cameras along the same side of an imaginary line connecting them. Such a rule results in a specific visual pattern made of two alternating, recurring shots and highly typical of a dialogue scene.

Relying on such patterns, we propose here to split the speaker diarization process into two steps when applied to fictional films: the first one consists in a local speaker diarization inside the boundaries of the visually detected dialogue scenes; the next one consists in clustering the local hypothesized speakers while preventing speakers locally assumed to be distinct from being merged into the same cluster and propagating this constraint at each iteration of the process.

Such a two-step clustering process is somehow related to what is denoted in [4] as the "hybrid architecture" in the cross-show speaker diarization context. In cross-show SD, diarization is achieved on a set of shows originating from a same source and containing possibly recurring speakers. The shows are first processed independently, before the resulting hypothesized speakers are clustered in a second stage. In [5], the authors make use of speaker diarization in conjunction with face clustering to identify the persons involved in a debate video: the best modality to identify a person is chosen and the iden-

*This work was partially supported by the French National Research Agency (ANR) CONTNOMINA project (ANR-07-240) and the Research Federation *Agorantic*, Avignon University.

tity information acquired for an instance is propagated to its whole cluster. Finally, speaker diarization has already been applied to TV series, but as a mean, among other modalities, to segment the whole video stream into homogeneous narrative scenes. In [6], the performances of mono-modal and multi-modal approaches for the scene segmentation task are evaluated and compared.

In this paper, rather than using speaker diarization to structure the TV movie, we propose to use its structure, as hypothesized from visual patterns, to improve the speaker diarization of such contents. The way such visual patterns are extracted is described in Section 2. The two steps of our speaker diarization approach, as well as the acoustic features used, are described in Section 3. Experimental results are presented and discussed in Section 4.

2. VISUAL PATTERNS DETECTION

The whole video stream can be regarded as a finite sequence of fixed images (or video frames) displayed on the screen at a constant rate to simulate motion continuity. As mentioned in [7], a shot is defined as "an unbroken sequence of frames taken from one camera".

As noticed in Section 1, because of technical narrative constraints, recurring and alternating shots frequently occur in the dialogue scenes of fictional movies, resulting in specific patterns.

In order to automatically extract such patterns, we then first need to split the whole video stream into shots and compare them to detect the recurring ones.

2.1. Shot segmentation and detection of similar shots

Defined by the continuity of the images it contains, a shot can also be defined, in a contrastive way, in opposition to the previous one. Shot segmentation is thus classically performed by detecting the transitions, either abrupt or gradual, between temporally contiguous shots ([7]). Remaining marginal in TV series, gradual transitions are here discarded and only abrupt ones (or cuts) are considered.

A cut between two contiguous shots is hypothesized if two temporally adjacent images differ from each other beyond a given threshold τ_1. Similarly, the present shot and a past one are considered as similar if the difference between the first image of the former and the last image of the latter stays below another threshold τ_2.

Both tasks, shot cut detection as well as shot similarity detection, require that two images be compared. 3-dimension histograms of the image pixel values in the HSV color space are used to describe the image. However, two different images may share the same color histogram, resulting in an irrelevant similarity: spatial information about the color distribution on images is reintroduced by splitting the whole image into 30 pixel blocks, each associated with its own histogram; block-based comparison of the resulting local histograms, as described in [7], is then performed. The similarity between two color histograms is measured by their correlation.

The two thresholds τ_1 and τ_2 respectively needed to achieve both tasks, shot cut detection and shot similarity detection, are estimated by experiments on a development set.

2.2. Shot patterns extraction

Let $\Sigma = \{l_1, ..., l_m\}$ be a finite set of m shot labels, two shots sharing the same label if they are hypothesized as similar as stated in the subsection 2.1.

The whole movie can then be described by a finite string $\mathbf{s} = s_1 s_2 ... s_k$ of shot labels, with each $s_i \in \Sigma$.

For any couple of shot labels $(l_1, l_2) \in \Sigma^2$, the following regular expression $r(l_1, l_2)$ denotes a subset of the set of all the possible shot label sequences $\Sigma^* = \bigcup_{n \geqslant 0} \Sigma^n$:

$$r(l_1, l_2) = \Sigma^* l_1 (l_2 l_1)^+ \Sigma^* \quad (1)$$

The set $\mathcal{L}(r(l_1, l_2))$ of strings denoted by such a regular expression corresponds to all the shot label sequences containing l_2 inserted between two occurrences of l_1 with a possible repetition of the alternation $l_2 l_1$, whatever be the previous and following shot labels. This regular expression formalize the intuition of the "two-alternating-and-recurring-shots" pattern mentioned in section 1 and typical of dialogue scenes.

Figure 1 shows a sequence of shots captured by the regular expression 1, as well as it illustrates the "180-degree" rule mentioned in section 1.

Fig. 1. *Example of shot sequence ...$l_1 l_2 l_1 l_2 l_1$... captured by the regular expression 1 for two shot labels l_1 and l_2.*

For a given movie described by a sequence $\mathbf{s} = s_1 s_2 ... s_k$ of shot labels, a set $\mathcal{P}(\mathbf{s}) \subseteq \Sigma^2$ of shot patterns is extracted by considering all the couples of shot labels $(l_1, l_2) \in \Sigma^2$ such that $\mathbf{s} \in \mathcal{L}(r(l_1, l_2))$:

$$\mathcal{P}(\mathbf{s}) = \{(l_1, l_2) \mid \mathbf{s} \in \mathcal{L}(r(l_1, l_2))\} \quad (2)$$

In other words, $\mathcal{P}(\mathbf{s})$ contains all the label pairs which occur as recurring subsequences of the form showed on Figure 1 in the whole movie sequence $\mathbf{s}$.

The set of utterances $\mathbf{u}(l_1, l_2)$ covered by the pattern (l_1, l_2) are then all these which occur whenever the two shots alternate with each other according to rule 1.

In order to increase the coverage of the patterns included in $\mathcal{P}(\mathbf{s})$ and reduce their sparsity, two extensions or the condition 1 are introduced.

1. In addition to rule 1, isolated expressions of the two alternating shots of the form $(l_1 l_2 | l_2 l_1)^+$ are taken into account, increasing the total amount of speech captured by the patterns.
2. The number of patterns is reduced while the average pattern coverage is increased by iteratively merging in a new pattern two patterns (l_1, l_2) and (l_1, l_3) with at least one label in common. As showed in Figure 2, such situations frequently occur during dialogues when one of the speakers (here the one appearing on the shots l_2 and l_3) is alternatively filmed from two distinct cameras. The resulting pattern gather all the utterances $\mathbf{u}(l_1, l_2)$ and $\mathbf{u}(l_1, l_3)$ covered by the merged patterns.

Table 1 reports the total coverage of the patterns extracted from the movies of our corpus (described in subsection 4.1), expressed as the ratio between the amount of speech covered by the patterns and the total amount of speech. The average duration of the speech covered by each pattern is also indicated, as well as the average number of speakers by pattern. These data are both computed by applying the basic version of the regular expression r, as given in equation 1, and by using the extended expression of r.

Fig. 2. *Shot sequence ...$l_1 l_2 l_1 l_3 l_1$... at the boundary of two adjacent patterns (l_1, l_2) and (l_1, l_3) with one shot in common.*

Table 1. *Shot patterns and speech: statistical data*

	coverage (%)	spch/patt (s.)	# of spk/patt
r	49.51	11.07	1.77
ext. r	**51.99**	**20.90**	**1.86**

As indicated in Table 1, the extracted visual patterns cover in average a bit more than half (51.99%) of the total amount of speech contained in the TV movies of our corpus.

69.85% of the patterns contain 2 speakers, 8.09% three and 22.06% only one. However, most of these one-speaker patterns correspond to short scenes, where the probability that one of the speakers remains silent increases.

Figure 3 shows the ability of such visual patterns to capture the main characters of a narrative movie: 97.96% of the characters speaking at least 5% of the time are involved in such patterns.

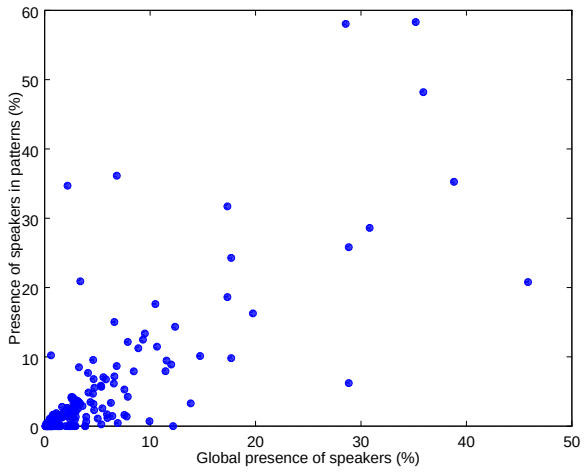

Fig. 3. *Global vs pattern presence of each speaker*

3. SPEAKER DIARIZATION

Speaker diarization is performed in two steps: speaker diarization is first achieved locally by clustering the set of utterances $\mathbf{u}(l_1, l_2)$ covered by the pattern (l_1, l_2); in a second stage, the locally hypothesized speakers are clustered in order to merge recurring speakers.

3.1. Acoustic features

Easily available, the subtitles of the movie are here used as an way to estimate the boundaries of the corresponding speech segments. As an exact transcription of the speech uttered, the subtitles temporally match it, despite a slight and variable latency before they are displayed on the screen and after they disappear. When the latency was too large, the subtitle boundaries were manually adjusted.

Moreover, a subtitle generally corresponds to a spoken segment uttered by a single speaker; on the remaining ones that cover two speech turns, the boundaries of each utterance are indicated, allowing to split the whole subtitle into two shorter ones.

The detection of change points between the possible audio sources, as a prerequisite of most of the diarization systems, is thus here avoided, allowing us to focus on the clustering process.

The acoustic parameterization of the resulting spoken segments is achieved by extracting 19 cepstral coefficients plus energy, completed by their first and second derivatives.

As a state of the art approach in the speaker verification field, i-vectors are used to retain the relevant acoustic information from each spoken segment ([8]). I-vectors are extracted by using a 512-components GMM/UBM and a total variability matrix trained on a development set.

The initial set of instances to cluster is then made of 60-dimension normalized i-vectors, each corresponding to a speech segment uttered by a single speaker.

3.2. Agglomerative local clustering

A first step of agglomerative clustering is processed within each local dialogue scene as hypothesized by the use of the visual patterns described in subsection 2.2.

For the set of utterances $\mathbf{u}(l_1, l_2)$ covered by the pattern (l_1, l_2), the bottom-up clustering algorithm relies on the following:

- The Mahalanobis distance is chosen as a similarity measure between the i-vectors corresponding to the spoken segments, resulting in a matrix M of similarity between the utterances contained in $\mathbf{u}(l_1, l_2)$.

  The covariance matrix used to compute the Mahalanobis distance is the within class covariance matrix of the training set, as mentioned in [9] and computed as follows:

$$W = \frac{1}{n} \sum_{s=1}^{S} \sum_{i=1}^{n_s} (\mathbf{u_i^s} - \overline{\mathbf{u_s}})(\mathbf{u_i^s} - \overline{\mathbf{u_s}})^T \qquad (3)$$

  where n denotes the number of spoken segments of the training set, S the number of speakers and n_s the number of segments uttered by the speaker s; $\overline{\mathbf{u_s}}$ is the mean of the i-vectors corresponding to utterances of speaker s and $\mathbf{u_i^s}$ denotes the i-vector corresponding to the $i-$th utterance of speaker s.

- The Ward's aggregation criterion is used during the agglomeration process to estimate the distance $\Delta I(c, c')$ between the clusters c and c'; it is computed as follows:

$$\Delta I(c, c') = \frac{m_c m_{c'}}{m_c + m_{c'}} d^2(g_c, g_{c'}) \qquad (4)$$

  where m_c and $m_{c'}$ are the respective mass of the two clusters, g_c and $g_{c'}$ their respective mass centers and $d(g_c, g_{c'})$ the distance between the mass centers.

- Finally, the Silhouette method is used to cut the dendogram resulting of the clustering process and obtain the final partition of the spoken segments. Described in [10], the Silhouette method allows to automatically choose a convenient partition of the instance set by evaluating the quality of a each possible partition resulting from the clustering process. For a

given partition, if instances appear closer to another cluster than to their own, the quality measure tends to decrease, and to increase if the instances are appropriately assigned to their respective clusters.

3.3. Constrained global clustering

Once the speaker diarization is performed inside each dialogue scene, a second stage of clustering is performed in order to merge the recurring speakers.

The set of segments locally clustered as uttered by the same speaker are extracted in order to be modelled by a speaker normalized i-vector $\mathbf{s_i}$ of 60 components.

The global clustering of the resulting set is processed in the same way than the local one, using Mahalanobis distance based on the W covariance matrix, Ward's aggregation criterion and the Silhouette method to extract the final partition of speakers.

However, this second step is guided, at each agglomeration step, by the structural information given by the visual segmentation of the movie into dialogue scenes as described in Section 2: the global clustering step has to prevent speakers locally hypothesized to be distinct from being assigned to the same cluster during the iterative agglomeration process.

The integration of such a constraint in the bottom-up clustering algorithm is achieved in the following way:

- In the initial matrix M of the distances between the i-vectors corresponding to the locally hypothesized speakers, the distance $d(\mathbf{s}, \mathbf{s}')$ between two instances $\mathbf{s}$ and $\mathbf{s}'$ is set to $+\infty$ if the corresponding two speakers appear together in the same dialogue scene:

$$d(\mathbf{s}, \mathbf{s}') = +\infty \Leftrightarrow \exists (l_1, l_2),\ \mathbf{u}(\mathbf{s}) \cup \mathbf{u}(\mathbf{s}') \subseteq \mathbf{u}(l_1, l_2) \quad (5)$$

  where (l_1, l_2) denotes a dialogue pattern, $\mathbf{u}(l_1, l_2)$, the set of utterances covered by the pattern (l_1, l_2) and $\mathbf{u}(\mathbf{s})$ the set of utterances assigned to the speaker $\mathbf{s}$ during the local clustering step.

- The distance $\Delta I(c, c')$ between the clusters c and c' is set to $+\infty$ if at least one instance of the first cluster is located at an infinite distance from an instance of the second one:

$$\Delta I(c, c') = +\infty \Leftrightarrow \exists (\mathbf{s}, \mathbf{s}') \in c \times c',\ d(\mathbf{s}, \mathbf{s}') = +\infty \quad (6)$$

  where $\mathbf{s}$ and $\mathbf{s}'$ denote i-vectors corresponding to hypothesized speakers.

The application of rules 5 and 6 prevents two distinct speakers from being clustered when choosing at each iteration of the agglomerative process the two closest instances to merge.

Figure 4 illustrates both the application of these rules at the initial step of the agglomerative process and how this "different-speakers" property is inherited by the newly created cluster. Local dialogue scenes are surrounded by dotted rectangles; each node s_{ij} represents the $i-$th speaker hypothesized in the $j-$th dialogue; the edges between two nodes represent their distance; the absence of edge between two nodes corresponds to an infinite distance. Merging the two closest nodes s_{11} and s_{12} results in an isolated cluster $s_{11}s_{12}$ that inherits both from the distinction between the two speakers of the first scene and from the distinction between those of the second one: the hypothesized recurring speaker in the two scenes has indeed to be different from both the speakers he is respectively talking to.

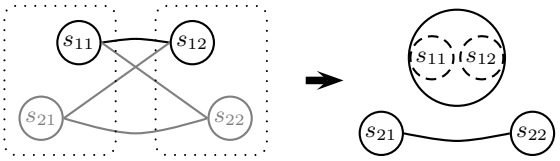

Fig. 4. *First iteration of constrained global clustering*

Such a "different-speaker" property, as propagated at each step of the agglomerative process, is expected to prevent the speakers involved in a same dialogue to be prematurely clustered: the background music of a dialogue may for instance hide the inter-speaker variability and cause such an early clustering.

Moreover, the main consequence of respecting such a constraint is to block the clustering process before assigning all the instances to the same cluster. In the small example of Figure 4, only one more step of the agglomerative process could be achieved, by clustering s_{21} and s_{22}: the narrative structure (two dialogues with two speakers each) remains indeed compatible with such a clustering. The resulting dendogram is then split into two distinct trees.

Figure 5 shows dendograms corresponding to agglomerative clustering of local speakers. The one figuring on top is obtained in a classical way, but may be difficult to cut automatically to extract the best partition of the instances. The bottom part of the figure, obtained with the same data by integrating the "different-speakers" property to the clustering process, shows five trees corresponding to five incompatible groups of speakers; each one is made of a group of narratively consistent speakers, with possibly many occurrences of the same one.

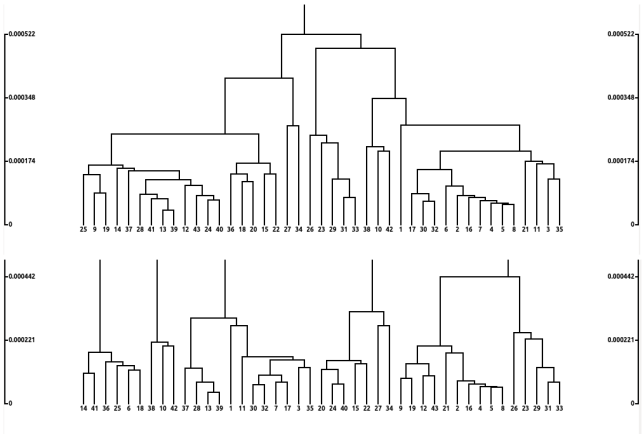

Fig. 5. *Dendograms obtained by agglomerative clustering on local speaker hypotheses, unconstrained (top); constrained (bottom)*

Each of these remaining trees of compatible speakers is finally cut using the Silhouette method described in subsection 3.2 and the final partition of the instance set is obtained by the union of the partitions obtained for each tree.

However, this constrained global clustering step remains dependent of the outputs of the local one. If a single speaker is wrongly split into two clusters during the local clustering step, the two re-

sulting utterance groups will never be merged during a global clustering embedding the "different-speakers" property. Nevertheless, even during an unconstrained clustering process, such groups would be merged lately, possibly after the best partition is reached.

4. EXPERIMENTS AND RESULTS

4.1. Corpus

For experimental purpose, we acquired the first seasons of three TV series: *Breaking Bad* (abbreviated *bb*), *Game of Thrones* (*got*), and *House of Cards* (*hoc*). We manually annotated three episodes of each series by indicating shot cuts, similar shots, speech segments as well as the corresponding speakers.

The total amount of speech in these nine episodes represents a bit more than three hours (3:12).

A subset of six episodes (denoted DEV) was used for development purpose, the remaining three ones (denoted TEST) being used for test purpose.

4.2. Shot cut and shot similarity detection

The evaluation of shot cut detection relies on a classical F1-score ([11]) based on recall (% of retrieved cuts among the relevant ones) and precision (% of relevant cuts among the retrieved ones). For the shot similarity detection task, an analogous F1-score is used: for each shot, the list of shots hypothesized as similar to the current one is compared to the reference list of similar shots; if both lists intersect in a non-empty set, the shot is considered as correctly paired with its list. Results on DEV and TEST sets are reported in Table 2.

Table 2. *Results obtained for shot cut and shot similarity detection*

	shot cut	**shot sim**		
	F1-score	precision	recall	F1-score
bb-1	0.93	0.88	0.81	0.84
bb-2	0.99	0.90	0.83	0.86
got-1	0.97	0.88	0.84	0.86
got-2	0.98	0.89	0.90	0.90
hoc-1	0.99	0.91	0.92	0.92
hoc-2	0.98	0.93	0.97	0.95
avg. DEV	**0.97**	0.90	0.88	**0.89**
bb-3	0.98	0.83	0.84	0.83
got-3	0.99	0.92	0.89	0.91
hoc-3	0.99	0.98	0.96	0.97
avg. TEST	**0.99**	0.91	0.90	**0.90**

The results obtained in both the image processing tasks, particularly for the shot similarity detection one (F1-score amounting to 0.90) are thus expected to provide a firm base for guiding speaker diarization of narrative movies. Precision is slightly more important than recall, resulting in some missed similarities between shots but with fewer false positives. As a result, the dialogue patterns are slightly less covering when based on automatic similarity detection (49.70% of the part-of-speech *vs* 51.99% when shot similarity is manually indicated) but appear highly reliable.

4.3. Local speaker diarization

The DER used to evaluate the local clustering step is computed independently in each episode dialogue before averaging the obtained scores according to each dialogue duration (*single-show* DER, as mentioned in [12]). The results are reported in Table 3, when using both the reference (denoted *ref.*) and the automatically detected (denoted *auto.*) similar shots. For the sake of comparison, agglomerative clustering (denoted AC), is compared to a "naive method" relying on a strong assumption of synchronization between the audio and video streams: clustering of local utterances is performed by assigning each spoken segment the label of the current shot, assuming the two alternating shots match exactly the speaker turns.

Table 3. *Single-show* DER *by episode obtained for the local diarization step*

	input auto.		**input ref.**	
	naive	AC	*naive*	AC
bb-1	30.26	**19.11**	22.81	21.00
bb-2	22.06	22.51	19.78	**19.14**
got-1	22.16	23.70	19.46	**15.78**
got-2	26.19	18.78	22.80	**16.61**
hoc-1	17.23	13.36	16.31	**11.84**
hoc-2	30.66	**18.18**	31.87	19.12
avg. DEV	24.76	19.27	22.17	**17.25**
bb-3	40.45	21.15	24.31	**12.15**
got-3	33.45	17.43	35.43	**12.80**
hoc-3	24.44	12.83	22.95	**12.82**
avg. TEST	32.78	17.14	27.56	**12.59**

The results obtained by performing an audio-based clustering of the utterances of each dialogue scene appear better than those obtained by applying the naive image-based method.

Moreover, the automation of the previous step, though slightly degrading performances in speaker diarization, does not really impact it, which confirms the reliability of the visual modality.

4.4. Global speaker diarization

Table 4 reports the results obtained during the clustering of the local speakers, achieving the second step of the speaker diarization process.

Table 4. DER *obtained for the global diarization step*

	input auto.		**input ref.**		**spch ref.**	
	2S	*cst. 2S*	*2S*	*cst. 2S*	LIA	LIUM
bb-1	51.36	56.00	52.66	**48.10**	72.06	67.21
bb-2	**41.83**	65.07	58.76	49.49	77.03	76.79
got-1	70.13	**52.79**	70.67	53.87	65.57	58.49
got-2	67.28	**38.85**	70.32	41.24	65.29	60.80
hoc-1	**50.04**	55.61	52.70	52.15	60.26	62.37
hoc-2	64.91	56.40	63.65	**37.09**	67.05	59.00
avg.	57.59	54.11	61.46	**46.99**	67.88	64.11
bb-3	60.41	**33.94**	59.22	42.64	60.61	55.56
got-3	74.71	**49.31**	70.34	63.17	61.33	52.89
hoc-3	**57.68**	59.87	67.52	67.41	70.55	67.05
avg.	64.13	**47.71**	65.69	57.74	64.16	58.50

Results are given both in taking as input the local speakers hypothesized in each dialogue scene during the previous step (*input*

auto.) as well as the real speakers manually annotated (denoted *input ref.*). In both cases, the second step of clustering is performed in an unconstrained way (denoted *2S*), allowing any local speakers to be clustered during the agglomerative process, and in a constrained way (denoted *cst. 2S*), by preventing it. For the sake of comparison, the results of two standard speaker diarization tools (denoted LIA, described in [13], and LIUM, described in [14] and [12]), are also reported: these tools receive in input all the spoken segments covered by the dialogue patterns.

Though still high, the DER is generally reduced by integrating to the clustering process the structural information based on visual patterns. By stopping the clustering before all the instances can be gathered, the "different-speakers" property allows to cut the resulting dendogram at a suitable level, providing an early stop condition of the process, when only a few mutually exclusive groups of instances remain. By contrast, unconstrained clustering has to face the critical issue of finding the optimal partition of the instances.

Table 5 reports the average number of speakers involved in the dialogue scenes considered, as hypothesized by the different systems.

Table 5. DER *Average number of hypothesized speakers*

	truth	*2S*	*cst. 2S*	LIA	LIUM
bb	**10.3**	7.3	**11**	6	25.7
got	**25.3**	4.7	15.7	9.3	**24**
hoc	**20.7**	3.7	**24**	6	27

As can be seen, two systems (unconstrained 2-step clustering and LIA), tend to cut the clustering dendogram at a high level, resulting in a few number of too wide classes. Conversely, LIUM, by cutting the tree at a low level, overestimates in two cases the number of speakers. The constrained clustering approach (*cst. 2S*), resulting in disjoint dendograms, offers an reasonable approximation of the number of speakers and prevents early as well as late cuts of the clustering tree.

5. CONCLUSION AND PERSPECTIVES

In this paper, we proposed to achieve speaker diarization of narrative movies by relying on the structural information they carry. By detecting similar shots, some covering patterns, typical of dialogue scenes, can be extracted and a first step of speaker diarization can be locally performed inside each dialogue boundaries. A second step of clustering, aiming at detecting the recurring speakers, is then applied to the locally hypothesized speakers: at each iteration of this global clustering process, the constraint that speakers locally assumed to be different must not be clustered is propagated; as a result, the agglomerative process is blocked far before all the instances are clustered, allowing a more convenient partition of the initial set than when applying an unconstrained approach.

Despite the coverage of the visual patterns, there still remains some sparse spoken segments outside their boundaries (near than a half of the total amount of speech). A specific study of the shot patterns involved in the dialogue scenes could allow to increase their coverage. The labelling of the remaining spoken segments could then be achieved by assigning them to the – possibly noisy – speaker models resulting from the SD process. Finally, visual information could be used during the local clustering of the dialogue utterances by exploiting the way the shots alternate with each other.

6. REFERENCES

[1] Pierre Clément, Thierry Bazillon, and Corinne Fredouille, "Speaker diarization of heterogeneous web video files: A preliminary study," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4432–4435. 1

[2] G. Friedland, H. Hung, and Chuohao Yeo, "Multi-modal speaker diarization of real-world meetings using compressed-domain video features," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, April 2009, pp. 4069–4072. 1

[3] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner, "The ami meeting corpus: A pre-announcement," in *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, Berlin, Heidelberg, 2006, MLMI'05, pp. 28–39, Springer-Verlag. 1

[4] Viet-Anh Tran, Viet Bac Le, Claude Barras, and Lori Lamel, "Comparing multi-stage approaches for cross-show speaker diarization.," in *INTERSPEECH*, 2011, pp. 1053–1056. 1

[5] Meriem Bendris, Benoit Favre, Delphine Charlet, Géraldine Damnati, Gregory Senay, Rémi Auguste, and Jean Martinet, "Unsupervised face identification in tv content using audio-visual sources," in *Content-Based Multimedia Indexing (CBMI), 2013 11th International Workshop on*. IEEE, 2013, pp. 243–249. 1

[6] Hervé Bredin, "Segmentation of tv shows into scenes using speaker diarization and speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2377–2380. 2

[7] Irena Koprinska and Sergio Carrato, "Temporal video segmentation: A survey," *Signal processing: Image communication*, vol. 16, no. 5, pp. 477–500, 2001. 2

[8] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011. 3

[9] Pierre-Michel Bousquet, Driss Matrouf, and Jean-François Bonastre, "Intersession compensation and scoring methods in the i-vectors space for speaker recognition.," in *INTERSPEECH*, 2011, pp. 485–488. 3

[10] Peter J Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987. 3

[11] John S Boreczky and Lawrence A Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–128, 1996. 5

[12] Mickael Rouvier, Grégor Dupuy, Paul Gay, Elie Khoury, Teva Merlin, and Sylvain Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," in *INTERSPEECH*, 2013, number EPFL-CONF-192762. 5, 6

[13] Simon Bozonnet, Nicholas WD Evans, and Corinne Fredouille, "The lia-eurecom rt'09 speaker diarization system: enhancements in speaker modelling and cluster purification," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4958–4961. 6

[14] Sylvain Meignier and Teva Merlin, "Lium spkdiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, 2010, vol. 2010. 6