Hierarchical Framework for Plot De-interlacing of TV Series based on Speakers, Dialogues and Images

Philippe Ercolessi, Christine Sénac,
Sandrine Mouysset
IRIT – Université Paul Sabatier
118 route de Narbonne – F-31062 Toulouse
{ercolessi, senac, mouysset}@irit.fr

Hervé Bredin
LIMSI – CNRS
BP 133 – F-91403 Orsay Cedex
bredin@limsi.fr

ABSTRACT

Since the 90's, TV series tend to introduce more and more main characters and they are often composed of multiple intertwined stories. In this paper, we propose a hierarchical framework of plot de-interlacing which permits to cluster semantic scenes into stories: a story is a group of scenes not necessarily contiguous but showing a strong semantic relation. Each scene is described using three different modalities (based on color histograms, speaker diarization or automatic speech recognition outputs) as well as their multimodal combination.

We introduce the notion of *character-driven episodes* as episodes where stories are emphasized by the presence or absence of characters, and we propose an automatic method, based on a social graph, to detect these episodes. Depending on whether an episode is character-driven or not, the plot-de-interlacing -which is a scene clustering- is made either through a traditional average-link agglomerative clustering with speaker modality only, either through a spectral clustering with the fusion of all modalities. Experiments, conducted on twenty three episodes from three quite different TV series (different lengths and formats), show that the hierarchical framework brings an improvement for all the series.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods; H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Experimentation

Keywords

video structuring, plot de-interlacing, scene clustering, spectral clustering, multimodal fusion

ACM MM AMVA Workshop 2012 Nara, Japan

1. INTRODUCTION

In our digital broadcasting era witnessing the rise of new content broadcasting channels such as the web and mobile phones, digital libraries are growing exponentially with very little or no manual labeling at all. It becomes necessary to provide users with efficient search and browsing tools.

This paper focuses on a particular type of digital libraries: collections of TV series episodes that can be automatically recorded when aired or downloaded from the web. The ultimate goal is to provide efficient automatic video abstraction tools – for a fast and easy overview of a whole series or a summary of (potentially missed) previous episodes. Similarly to *Sundaram & Chang* [12], we aim at generating meaningful video skims that take into account the complexity and temporality of the actual video content.

Based on the idea that TV series are segmented into narrative themes at post-production, we present a system able to discover the original structure of an episode. Since the 90's, TV series tend to introduce more and more main characters and they are often composed of multiple intertwined stories. The main objective of plot de-interlacing is to cluster scenes back into stories – where a story is a group of scenes showing a strong semantic coherence.

The top row of Figure 1 shows what kind of output is expected from a perfect plot de-interlacing algorithm, applied on an episode of the *Malcolm in the Middle* TV series. The episode is segmented into 26 scenes which are divided into three stories (numbered #1, #2 and #3) of various lengths. All three stories converge at one point in scene #14 and then diverge again. Four scenes do not belong to any actual story. They correspond to credits at the beginning and end of the episode, transition scenes or even sketches.

The main difference with previous story segmentation techniques is that scenes from the same story do not have to be contiguous. As opposed to existing story segmentation works that focus on specific types of programs with a rigid structure (such as broadcast news [6] or soccer games), plot de-interlacing is not a temporal segmentation task, it is a clustering task.

2. OVERVIEW

This paper is an extension and improvement over our previous work on the subject [4]. This previous work basically provided a comparison of multiple scene clustering algorithms applied to mono-modal scene descriptors. In this paper, we propose a multi-modal approach to scene clustering for plot de-interlacing and design a hierarchical framework that automatically selects the best clustering approach

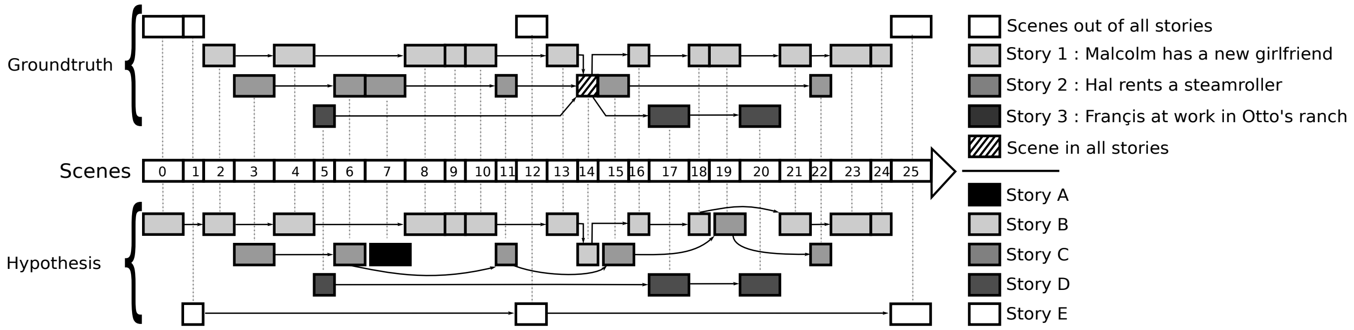
Groundtruth
Hypothesis
Scenes
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
Scenes out of all stories
Story 1 : Malcolm has a new girlfriend
Story 2 : Hal rents a steamroller
Story 3 : Françis at work in Otto's ranch
Scene in all stories
Story A
Story B
Story C
Story D
Story E

Figure 1: Expected and automatic plot de-interlacing

depending on the type of stories of the processed episode.

Section 3 first introduces three monomodal distances between scenes (based on speakers, dialogues and image content) that are then combined into a multimodal distance used as input of spectral clustering.

The notion of Character-Driven Episodes (CDE) is introduced in Section 4. Those are episodes where each story focuses on a small fixed set of characters and where very few characters are part of more than one story. We describe a method to detect CDE automatically, using a state-of-the-art community detection algorithm applied on the speaker social graph. We also show that the optimal clustering approach depends on whether the episode is a CDE or not. Therefore, a hierarchical framework for plot de-interlacing is introduced in Section 4.1. It is illustrated in Figure 2.

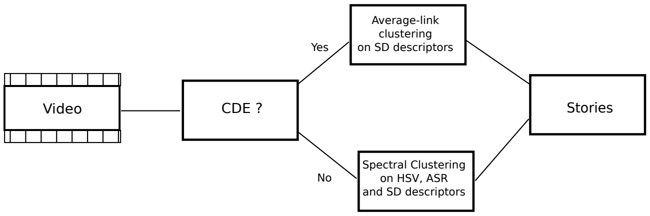
Video
CDE ?
Yes
No
Average-link clustering on SD descriptors
Spectral Clustering on HSV, ASR and SD descriptors
Stories

Figure 2: Plot de-interlacing framework

Section 5 shows the whole set of experiments performed on a collection of 23 episodes from 3 different TV series. Section 7 concludes the paper.

3. SCENE CLUSTERING

As illustrated in Figure 1, plot de-interlacing consists in separating each story from each other and grouping pre-segmented scenes altogether. A scene is a group of consecutive shots describing temporally continuous and semantically coherent events. Though we did propose an efficient technique for automatic segmentation of TV series into scenes in [3], we rely on manual segmentation in this paper.

As previously stated in the introduction, we tackle the plot de-interlacing task as a scene clustering problem. First, a distance measure is obtained for every pair of clustered entities (scenes, in our case). Section 3.1 describes the ones we used. Then, using the sole knowledge of the resulting distance matrix, the actual clustering algorithm finds homogeneous groups of entities (stories). Both average-link agglomerative clustering and spectral clustering are investigated in Section 3.3.

3.1 Mono-modal distances

As proposed in our previous publication [4], and inspired by the classical unities of action, place and time in theater, we propose to use three types of distance between scenes, based on three different modalities: visual content, speakers and dialogues.

3.1.1 Visual content (HSV)

Two scenes that take place in the same location are more likely to be part of the same sub-story than if they happen in completely different places. The first proposed distance measure is therefore based on the visual content.

1000-dimensional HSV color histograms ($10 \times 10 \times 10$ bins) are extracted every second. The distance d_{ij}^{HSV} between two scenes i and j is defined as the average minimum distance between all pairs of histograms:

$$d_{ij}^{\text{HSV}} = \begin{cases} \dfrac{1}{|H_i|} \displaystyle\sum_{h \in H_i} \min_{g \in H_j} d_M(h,g) & \text{if } |H_i| > |H_j| \\ d_{ji}^{\text{HSV}} & \text{otherwise.} \end{cases}$$

where H_k is the set of histograms extracted from scene k, $|H_k|$ its cardinal and d_M the Manhattan histogram distance.

3.1.2 Speakers (SD)

Similarly, it is expected that a story will tend to closely follow one particular character (or one small set of characters) and what is happening to them.

A person who is speaking usually implies that he/she is part of the current story. This is why our second distance measure relies on the output of an automatic speaker diarization (SD) system [1]. Speaker diarization is the process of partitioning the audio stream into homogeneous segments, based on the identity of the speaker.

Zero, one or more speakers may speak during each scene. Therefore, we introduce a distance d_{ij}^{SD} based on the count of common speakers in scenes i and j. All pairs of scenes are sorted in decreasing order based on the number of common speakers. The motivation behind this ranking comes from the observation that two scenes are more likely to be part of the same story if they have a greater number of characters in common. Ties are sorted in increasing order based on the total number of speakers involved in each pair. The distance d_{ij}^{SD} between scenes i and j is defined as follows:

$$d_{ij}^{\text{SD}} = \frac{R_{ij}}{N} \tag{1}$$

where N is the total number of pair of scenes in the episode and R_{ij} is the rank of the pair made of scenes i and j in the over-mentioned ranking.

3.1.3 Dialogues (ASR)

This third distance aims at narrowing the so-called semantic gap by incorporating the actual subjects of discussion between characters. It relies on the output of an automatic speech recognition (ASR) system [5].

The ASR output is processed by TreeTagger [10] in order to extract the lemma of each recognized word. Each scene is then described by a D_{ASR}-dimensional TF-IDF feature vector, where D_{ASR} is the total number of unique lemmas recognized by the ASR system in the episode.

The ASR-based distance d_{ij}^{ASR} between scenes i and j is defined as the cosine distance between their respective TF-IDF feature vectors.

$$d_{ij}^{\text{ASR}} = d^{cos}(\text{TF-IDF}_i, \text{TF-IDF}_j), \tag{2}$$

where d^{cos} is the cosine distance between two vectors.

3.2 Multimodal distance

In order to get the most out of these three complementary modalities, we also define a multimodal Gaussian distance:

$$d_{ij}^{FUS} = e^{-\frac{\|V_{ij}\|_2^2}{2\sigma^2}} \text{ with } V_{ij} = [d_{ij}^{\text{HSV}}, d_{ij}^{\text{SD}}, d_{ij}^{\text{ASR}}]$$

where $\sigma = \max\limits_{i,j \in \{1,..,N\}} \|V_{ij}\|_2$ and $\|.\|_2$ is the euclidean norm.

The use of the Gaussian kernel gives the opportunity to define clusters without a priori on shapes. This means that data that can not be easily separated in the original space can be clustered into homogeneous groups in the implicitly transformed high dimensional feature space.

3.3 Clustering approaches

In [4], we investigated the use of several clustering approaches. In this paper, the proposed hierarchical framework uses two different clustering algorithms: an *average-link* agglomerative clustering (that gives the best results in [4]) and the *spectral clustering* that we introduce below.

Spectral clustering consists in selecting dominant eigenvectors of the Gaussian similarity matrix in order to define a low-dimensional data space in which data points are easy to cluster [9, 11]. In the following, we present the adapted spectral clustering method to the framework fusion and we present an heuristic method to automatically determine the number of clusters, denoted k and thus, provide a fully-unsupervised spectral clustering.

The Gaussian similarity matrix is set to be the multimodal fusion measure d_{ij}^{FUS} defined in Section 3.2. After a normalization step, the k dominant eigenvectors are extracted where k is the number of clusters. By stacking the eigenvectors associated with the k largest eigenvalues, data points plotted in the spectral embedding are grouped in k clusters via the *K-means* method. The partition in the spectral embedding directly provides the partition in the original data space.

To define the number of clusters k, the Gaussian similarity matrix is exploited [7]. After indexing data points per cluster for a value of k, the indexed Gaussian similarity matrix, whose diagonal affinity blocks represent the affinity within a cluster and the off-diagonal ones the affinity between clusters, is defined. The ratios between the Frobenius norm of the off-diagonal blocks and that of the diagonal ones is evaluated. Among various values for k, the final number of clusters is defined so that the affinity between clusters is the lowest and the affinity within cluster is the highest.

4. DETECTION OF CHARACTER-DRIVEN EPISODES

As shown in Section 5.3 and paper [4], we tested every combination of modality (HSV, SD, ASR or their fusion) and clustering approach (spectral or average-link) on a set of 23 episodes from 3 different TV series.

One combination appears to lead to the best performance for almost two third of the episodes: the speaker modality combined with average-link clustering. These episodes usually contain 3 or 4 independent stories centered on disjoint character communities: we call them *Character-Driven Episodes* (`CDE`).

These types of episodes are not specific to a particular genre of TV Series. `CDE` episodes can be found in *sitcoms* (*Friends*, *Malcolm in the Middle*), dramas (*Ally McBeal*), fantasy (*A Game of Thrones*) or thrillers (*CSI: Crime Scene Investigation*). A `CDE` is an episode with two or more stories and where separated set of characters are involved in different stories.

The remaining episodes have stories that are centered on a particular topic rather than a set of characters: they are denoted $\overline{\texttt{CDE}}$. As illustrated in Figure 2, we propose to automatically classify episodes into `CDE` or $\overline{\texttt{CDE}}$.

Our approach is inspired by the characters social interaction graph introduced by *Weng et al.* in [13].

It aims at establishing the relationships between characters of the episode. Figure 3 shows its construction. Each character (or speaker, in our case) is associated to a node in the graph. An edge between two nodes means that corresponding characters both appear in at least one common scene. Edges are weighted by the number of scenes characters have in common. This leads to the creation of an undirected graph representing the social interactions between the characters (or speakers) of the episode.

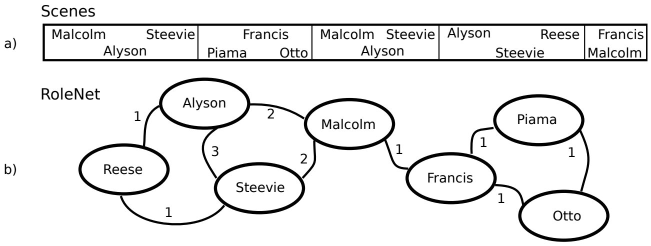

Figure 3: (a) List of characters for each scene. (b) Characters social graph.

Based on this graph, we propose to apply a state-of-the-art algorithm for community detection: the so-called *Louvain* approach recently proposed by *Blondel et al.* [2]. It is a heuristic method based on the maximization of a quantity called modularity and denoted $\mathcal{Q}$:

$$\mathcal{Q} = \frac{1}{\sum_{i,j} A_{ij}} \sum_{i,j}^{N_c} \left[A_{ij} - \frac{\sum_k A_{ik} \sum_k A_{kj}}{\sum_{i,j} A_{ij}} \right] \delta_{ij} \tag{3}$$

where $\delta_{ij} = 1$ if node i and j are members of the same detected community, 0 otherwise, N_c is the number of characters and A_{ij} is the weight between nodes i and j. $\mathcal{Q}$ can be seen as a measure of the quality of the detected communities. It increases when communities have stronger intra-community and weaker inter-community edges [8].

Starting with as many communities as there are nodes, the *Louvain* approach looks at all nodes for a potential change of community resulting in a higher modularity. Once modularity can no longer be improved, a new graph is built – in which every community is a node and edges are weighted by the sum of the corresponding edges in the original graph. This process is repeated until the maximum of modularity is reached. For a more detailed description and analysis of the algorithm, the interested reader might want to have a look at reference [2].

The higher $\mathcal{Q}$ is, the more disjoint character communities are and the more likely the episode is character-driven. We propose to use this value to automatically detect character-driven episodes. Episodes with higher modularity than a threshold are said to be `CDE`, while the others are classified as $\overline{\texttt{CDE}}$.

The optimal threshold is automatically determined by leave-one-out cross-validation in order to maximize the average performance of the global hierarchical framework described in the next paragraph.

4.1 Hierarchical framework

As highlighted in Table 1, average-link agglomerative clustering based on speaker distance leads to the best performance for character-driven episodes. Similarly, spectral clustering in the multimodal domain tends to obtain the best result for episodes that are not driven by characters.

Therefore, the global framework (illustrated in Figure 2) will select the optimal clustering method on a specific episode depending on the result of the output of the `CDE` detection algorithm.

5. EXPERIMENTS

5.1 Corpora

In order to evaluate our proposed algorithms on actual TV series, we manually annotated 7 episodes of the TV series called *Ally McBeal*, 7 episodes of the one called *Malcolm in the Middle*, and 9 of the one called *A Game of Thrones*. Manual annotations include scenes boundaries and stories defined by the list of their scenes.

All in all, the *Ally McBeal* dataset lasts approximately 5.5 hours, with 304 scenes gathered into 20 stories (about 2.5 stories per episode on average) while the *Malcolm in the Middle* dataset is 2.5 hours long, with 196 scenes and 24 stories (about 3.4 stories per episode), and the *Game of Thrones* dataset lasts 8.5 hours, with 274 scenes and 38 stories (about 4 per episode).

There are two american comedies and one fantasy series. They differ in their format: different durations of episodes, different numbers and lengths of scenes.

5.2 Evaluation method

We introduce a new evaluation method borrowed from the *speaker diarization* community: the *Diarization Error Rate* (DER).

The output of a speaker diarization system consists of a list of speech segments described with starting time, ending time and speaker cluster name (this list is called hypothesis). It is evaluated against a manually annotated groundtruth (called reference). The evaluation looks for an optimum one-to-one mapping between the hypothesis segments and the reference segments so that the total overlap time between the reference speaker and the corresponding mapped speaker cluster returned by the hypothesis is maximized. Labels between reference and hypothesis do not have to be the same. The DER is defined as the sum of three errors:

$$DER = \frac{\text{False alarm} + \text{Misses} + \text{Speaker error}}{\text{Episode total duration}} \qquad (4)$$

False alarm is the total duration of segments where speech is present in the hypothesis but not in the reference,

Misses is the total duration of segments where speech is present in the reference but not in the hypothesis,

Speaker error is the total duration of segments where the mapped reference is not the same as the speaker found by the system in the hypothesis.

The DER can be directly translated into the plot de-interlacing domain using the following analogy: each story corresponds to a speaker and each scene to a speech segment.

The distinction between speech and non-speech segments is not obvious in our case – as scenes cover the entire duration of the episode. This would systematically lead to False alarm = Misses = 0.

However, some particular transition scenes are not part of any story. They could be considered as non-speech segments in the computation of the diarization error rate.

Unlike F-measure, the DER measures the error of the method (not the accuracy). A perfect clustering will reach a DER of 0 while the worse clustering will get a DER close to 1. It is illustrated in Figure 4.

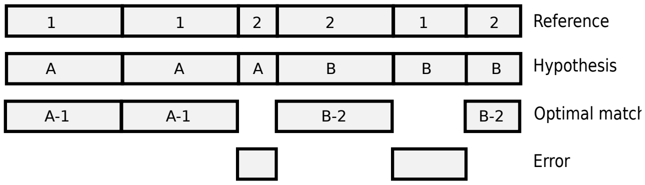

Figure 4: Diarization Error Rate. $DER = 0.23$

5.3 Results

Table 1 summarizes results obtained on various subsets of the evaluation corpus: for each TV series, for `CDE` or $\overline{\texttt{CDE}}$ episodes only and for the whole corpus.

The table allows to compare the performance of each approach on every collection of episodes – using the SD modality only, the combination of SD and ASR (+ASR) or the combination of all three modalities (+HSV). Column FW (FrameWork) contains the performance of the overall framework (with `CDE` detection).

The column entitled *Random* is obtained using average-link clustering based on random distances between all scenes. Random values are the average DER of 100 random clusterings minus three times the standard deviation (all values smaller than this value are better than 95% of the random clusterings). The *Best* column shows the best score that can be obtained with our proposed approaches. As a matter of fact, none of them allows a scene to be part of

COLLECTION	Average-link			Spectral Clustering			FW	Random	Best
	SD	+ASR	+HSV	SD	+ASR	+HSV		baselines	
Ally McBeal	0.49	0.63	0.77	0.49	0.54	0.47	0.45	0.64	0.18
Malcolm	0.24	0.65	0.64	0.39	0.41	0.32	0.23	0.54	0.14
Game of Thrones	0.30	0.61	0.65	0.43	0.36	0.42	0.30	0.55	0.03
CDE	**0.26**	0.65	0.61	0.43	0.39	0.43	0.26	0.51	0.05
$\overline{\text{CDE}}$	0.56	0.66	0.75	0.55	0.51	**0.48**	0.48	0.61	0.19
All series	0.34	0.63	0.68	0.44	0.43	0.40	**0.32**	0.53	0.11

Table 1: Average-link clustering vs. spectral clustering using different modalities. None of our approaches is able to be better than the Best baseline as they do not allow overlapping stories. FW = Framework (diarization error rate)

more than one story. Thus, in the best case scenario, if at least one scene belongs to two or more stories, DER will be greater than zero.

Table 1 confirms previous results obtained in [4]: SD modality gives the best results with an average-link clustering for almost all collections. Another interesting result (not shown in this table) is that, considering mono-modal distances, only SD modality gives better results than the random baseline. However, combining SD modality with the other ones leads to the best result when considering only spectral clustering on the $\overline{\text{CDE}}$ collection.

For CDE episodes, average-link clustering brings an improvement of 13% compared to spectral clustering, whereas spectral clustering outperforms average-link clustering by 8% for $\overline{\text{CDE}}$ episodes. Therefore, the hierarchical framework, based on the classification of episodes into CDE or $\overline{\text{CDE}}$, permits to choose the more appropriate clustering and gives an average improvement of 2% on the global collection compare to the best system (average-link with SD modality).

An extensive analysis of results shows that the best clustering method is not correctly selected for 4 out of 23 episodes. Note that annotation of these episodes was very tricky and the various annotators did not agree at first.

One source of error comes from the automatic selection of the optimal number of stories in the clustering methods. We have compared the average results obtained when the correct number of stories (i.e. clusters) is known in advance and when the optimal number of stories is automatically computed by the algorithm itself. We found that knowing the correct number of clusters brings an improvement of 7% when considering the spectral clustering method. Likewise, the framework performance is improved by 3%.

Another source of error is due to the fact that neither average-link agglomerative clustering nor spectral clustering is able to generate partially overlapping stories (i.e. to detect scenes belonging to more than one story). Making two stories converge at one time and then diverge again (as illustrated in Figure 1) is a common practice in the construction of modern TV series. For instance, our evaluation corpus is made of 478 scenes, 50 of which belong to at least two stories. The average DER value of 0.11 in column *Best* in Figure 1 says it all: it is the lowest error rate that can be achieved by such exclusive scene clustering approaches.

Figure 5 shows the improvement of fusing the three modalities using spectral clustering on an episode of *Ally McBeal*. It represents $K - means$ results in the spectral embedding space by using respectively SD distance and multimodal distance FUS . Each symbol (*squares*, *triangles* and *diamonds*) indicates the scenes which have to be in the same annotated story whereas the *circles* symbols are scenes out of all stories and *plus* are scenes which belongs to multiple stories. The circles C_i, for $i \in \{1, 2, 3\}$, describe the three clusters defined by the $K-means$ method. It shows that the plotted scenes in the spectral embedding are better separated with the fusion of the three modalities **b)** than considering only the SD one **a)**.

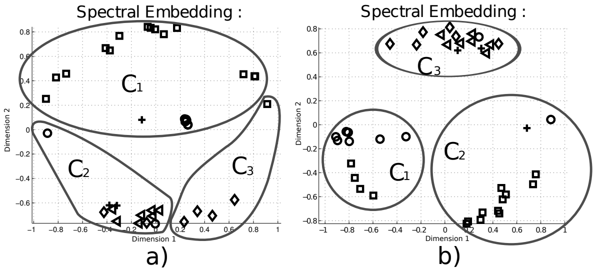

Figure 5: Scenes plotted in the embedding space: a) $K-means$ result on scenes described by SD modality. b) $K-means$ result on scenes described by the fusion of the three modalities.

6. STOVIZ VISUALIZATION TOOL

Both for visualization and evaluation, we designed StoViz – a web-based interface for browsing TV series episodes by story and comparison of manual and automatic plot de-interlacing results.

As ASR and SD computation time is quite low, StoViz only allows a visualization of results, not a computation on new videos. When the user chooses a video, a timeline appears, showing the scene segmentation of the episode. Four buttons are proposed to the user : the first one shows the scenes segmentation of the video (selected by default). The second one rearranges the scenes of the timeline in several timelines, each timeline showing an annotated story. The third button does the same thing but each timeline showing an automatically detected story using our framework. In this representation of the video, all incorrectly classified scenes will appear in red while correctly classified scenes will appear in grey, allowing the user to easily see the classification performance. The fourth one shows an abstract composed of one scene selected from each detected story.

The user can play and pause the video, click on a timeline to navigate quickly in the video, and play each story independently.

A demonstration is available online at `http://stoviz.niderb.fr`. Figure 6 contains a representative screenshot.

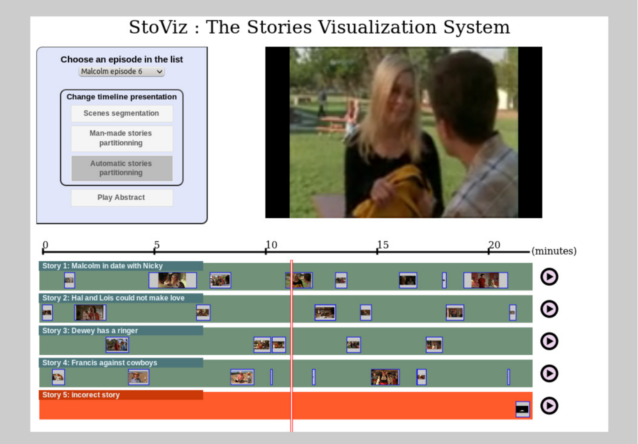
StoViz : The Stories Visualization System
Choose an episode in the list
Malcolm episode 6
Change timeline presentation
Scenes segmentation
Man-made stories partitionning
Automatic stories partitionning
Play Abstract
0
5
10
15
20
(minutes)
Story 1: Malcolm in date with Nicky
Story 2: Hal and Lois could not make love
Story 3: Dewey has a ringer
Story 4: Francis against cowboys
Story 5: incorect story

Figure 6: StoViz : a visualization and evaluation tool for plot de-interlacing.

7. CONCLUSION

Since the 90's, TV series tend to introduce more and more main characters and they are often composed of multiple intertwined stories. Following previous works, in this paper we have proposed a hierarchical framework in order to rearrange the scenes of an episode into semantically coherent stories. Each scene is characterized by different modalities: this includes visual (color histograms) or audio (outputs of speaker diarization system and outputs of an automatic speech recognition system) and their different combination.

We have introduced the notion of character-driven episodes and we give a method to automatically detect those episodes. It is based on a graphical representation of the interactions between characters appearing in an episode and on its automatic processing, using *Louvain* state-of-the-art algorithm, to detect and measure the strength of communities of characters. Depending on whether an episode is character-driven or not, the hierarchical framework allows to automatically select the plot de-interlacing approach which is optimal for this episode. So, plot-de-interlacing uses an average-link agglomerative clustering with speaker modality for character-driven episodes or a spectral clustering with the fusion of all modalities for other episodes. Experiments, conducted on three quite different TV series, show that the hierarchical framework brings an improvement for all the series.

In order to have an effective and visual comparison between manual and automatic plot-de-interlacing, we have developed a web-application, StoViz , which allows also a user to browse a video by following a specific story which takes place into its narration.

As the plot-de-interlacing system is generic and does not rely on models, we plan to test it for other types of video than TV series, such as archives of TV news providing a structure that can be dissociated into several stories.

Finally, as the work presented in this paper is part of a larger project which aims at providing the end-user with tools for fast and easy overview of one episode, one season or the whole TV series, we plan to work on the summarization aspect.

8. REFERENCES

[1] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain. Multi-Stage Speaker Diarization of Broadcast News. *IEEE TASLP*, 14(5):1505–1512, September 2006.

[2] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast Unfolding of Community Hierarchies in Large Networks. *Computing Research Repository*, abs/0803.0, 2008.

[3] H. Bredin. Segmentation of TV Shows into Scenes using Speaker Diarization and Speech Recognition. In *ICASSP 2012, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012.

[4] P. Ercolessi, H. Bredin, C. Sénac, and P. Joly. Toward Plot De-interlacing in TV Series Using Scenes Clustering. *IEEE CBMI*, 2012.

[5] J. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System. *Speech Communication*, 37(1-2):89–109, 2002.

[6] C. Ma, B. Byun, I. Kim, and C. Lee. A Detection-based Approach to Broadcast News Video Story Segmentation. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1957 –1960, april 2009.

[7] S. Mouysset, J. Noailles, D. Ruiz, and R. Guivarch. On A Strategy for Spectral Clustering with Parallel Computation. *High Performance Computing for Computational Science–VECPAR 2010*, pages 408–420, 2011.

[8] M. E. J. Newman. Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582, June 2006.

[9] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[10] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, 1994.

[11] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[12] H. Sundaram and S. Chang. Condensing Computable Scenes Using Visual Complexity and Film Syntax Analysis. *IEEE International Conference on Multimedia and Expo, 2001. ICME.*, pages 273 – 276, 2001.

[13] C.-Y. Weng, W.-T. Chu, and J.-L. Wu. RoleNet: Movie Analysis from the Perspective of Social Networks. *Multimedia, IEEE Transactions on*, 11(2):256 –271, feb. 2009.