VIDEO SCENE SEGMENTATION OF TV SERIES USING MULTI-MODAL NEURAL FEATURES

AMAN BERHE, CAMILLE GUINAUDEAU, CLAUDE BARRAS

Name Aman Berhe
Academic centre LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay
E-mail address aman.berhe@limsi.fr

Name Camille Guinaudeau
Academic centre MSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay
E-mail address camille.guinaudeau@limsi.fr

Name Claude Barras
Academic centre LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay
E-mail address claude.barras@limsi.fr

KEYWORDS
Scene segmentation; TV series; Neural features; Multimodal fusion; Unsupervised.

ABSTRACT
Scene segmentation of a video, a book or TV series allows them to be organized into logical story units (LSU) and is an essential step for representing, extracting and understanding their narrative structures. We propose an automatic scene segmentation method for TV series based on the grouping of adjacent shots and relying on a combination of multimodal neural features: visual features and textual features, further augmented with the temporal information which may improve the clustering of adjacent shots. Reported experiments compare the combination of different features, video frames sub-sampling and various shot clustering algorithms. The proposed method achieved good results, using different metrics, when tested on several seasons of two popular TV series.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

1. INTRODUCTION

We witness narrative structures everywhere in our daily lives. Movies, books or TV series are major sources of narrative structures. They help us to interpret our actions and give meaning to our lives. Understanding the narrative structure of TV series through automatic processing is thus a very interesting problem despite its complexity, and researchers on computational narratives believe that it may become a new area of artificial intelligence.

Extracting the narrative structure of a TV series can be addressed by dividing it into simple and specific modules that perform part of the task. One of the basic modules is the segmentation of a TV series into scenes. Having a solid scene segmentation system will have a vital role for performing scene analysis and understanding the narrative structure.

Our work focuses on an automatic scene segmentation system of TV series, using multimodal features extracted through deep neural networks. The multimodal nature of the video makes scene segmentation tasks difficult. On top of that, the definition of a scene can differ based on the TV series we are dealing with and depending on the people analyzing the problem.

Many papers define a scene differently according to the problem they are dealing with. We compose our definition from Bost (2016) and Kumar, Rai, Pulla, & Jawahar (2011) who defined a scene as a set of contiguous shots which are connected by a central concept or theme or coherent subject.

Even if some works rely on speaker diarization (i.e. characters occurrences within a scene) for scene segmentation Ercolessi, Bredin, Sénac, & Joly (2011), scene definition cannot be based on the set of characters. In most TV series, like *Game of Thrones* (2011-2019), when several new characters appear while others disappear, it is usually a serious hint of a scene change. However, there is a lot of counterexamples -- where the topic changes while the set of characters stay the same or where the topic stays the same even if some characters have left. Besides, some sitcoms include some characters that appear in almost all of the scenes of the TV series, like *The Big Bang Theory* (2007-2019).

Del Fabro and Böszörmenyi (2013) believed that "finding scenes in TV series and sitcoms is simpler than finding scenes in movies". But we believe this may not always be the case. Some TV series are very complex and can, in fact, be more complicated than a standalone movie. TV series and sitcoms are typically characterized not only by a fixed group of actors and a limited set of locations where the plot takes place, as explained by Del Fabro and Böszörmenyi (2013), but they also present a different range of stories and different parallel stories within each episode. Their characteristics, especially the protagonists, may remain the same across all episodes even as they evolve through the episodes, whether mentally, behaviorally and physically.

The segmented scenes will have an important role for extracting and understanding the narrative structure of a TV series. Furthermore, it will be used to create links between different scenes and connect the intertwined stories. If a scene segmentation method achieves good results for one episode it is likely that it will work well with the rest of the episodes.

The main contribution of our paper includes the following core points.

1. We propose a method for automatically segmenting a video into scenes using multimodal features.
2. We propose to use well-known, pre-trained deep neural network models to extract features from the frames of the video and we combine them with the word embedding of video's subtitles based on a shot. We also use each shot's temporal information as a feature to group shots that are closer to each other. Using all the features makes it possible to use the multimodal nature of a video.
3. We design a sequence-splitting algorithm to group together shots from the clustering step. This results in sequences of shots belonging to each scene and makes it possible to perform further processing at the scene level.

Our paper is organized as follows. In Section 2, we will cover prior work on video scene segmentation. Next, we will discuss our methods in Section 3. In Section 4, we will present the dataset used and introduce our results. Finally, the conclusion and future work are presented in Section 5.

2. RELATED WORK

Del Fabro and Böszörmenyi (2013) surveyed 20 years of video scene segmentation, discussing the methods investigated by many researchers using different algorithms. They categorized the approaches based on the combination of three classes of low-level features – visual, audio and textual – resulting in seven categories.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

Deep learning recently gained popularity for visual features extraction (Baraldi, Grana and Cucchiara 2015, Protasov et al. 2018). For example, Protasov et al. (2018) compute deep convolutional features using the Places205-AlexNet image classification network, motivated by the idea that the shots of a scene share a common surrounding. We want to extend this visual surrounding with the verbal context of a scene. Bost (2016) developed a video scene segmentation framework that segments the video content into story units. Bost's (2016) framework is formulated in a statistical fashion and uses the Markov Chain Monte Carlo (MCMC) technique to determine the boundaries between video scenes. He tried to use visual content and speech content of the video.

It is also possible to transform the scene segmentation problem into a graph problem, as did Yeung, Yeo and Liu (1998), Sidiropoulos et al. (2009), and Ercolessi et al. (2011). They used minimum edge detection for grouping adjacent shots into scenes. Kumar et al. (2011) propose bag of visual words of a shot and a post clustering based on a graph. They used a color histogram with a threshold to detect the shot boundaries, then picked key frames and did clustering based on their histogram and finally computed the similarity of shots with their neighbors. But we believe that the colors do not carry all the information needed for scene segmentation.

On the other hand, the segmentation problem can be considered based on text only, the most famous text topic segmentation algorithms being Texttiling and C99. Texttiling (Hearst 1997), which mainly subdivide texts into multi-paragraph units that represent subtopics, makes use of patterns of lexical co-occurrence and distribution, and C99 (Choi 2000) used ranking scheme and the cosine similarity measure as their main step for text segmentation. Utiyama and Isahara (2001) have used a statistical approach that tries to find the maximum probability of a text's segmentation. Their method does not require training data and they claim it be applied to any text. Guinaudeau, Gravier, and Sébillot (2012) proposed modifications of the computation of the lexical cohesion to make the algorithm proposed by Utiyama and Isahara more robust to TV programs automatic transcripts peculiarities (compared to written text). Scaiano et al. (2010) perform scene segmentation in a movie using the text of the subtitles. They used a vector of Synsets instead of a vector of words with the cosine similarity.

Various metrics can be used for the evaluation of scene segmentation. Purity and coverage, for example, which are borrowed from clustering evaluation metrics, are used by Ercolessi et al. (2011) and Del Fabro and Böszörmenyi (2013). Recall and Precision, from Information Retrieval evaluation, are also used to evaluate segmentation algorithm, by Baraldi et al. (2015) and Chasanis, Likas and Galatsanos (2009), for example. Recall and precision are used in order to estimate how accurate the detected boundaries are. There is an argument that these metrics are not quite appropriate for segmentation systems. WindowDiff (Pevzner and Hearst, 2002) and Pk (Beeferman et al., 1997) measures were defined especially for topic segmentation evaluation. Beeferman et al. (1997) defined Pk as the probability that two sentences drawn randomly from the corpus are correctly identified as belonging to the same document or not. Pevzner and Hearst (2002) define WindowDiff that counts the number of boundaries between the two ends of a fixed-length window and compare this number with the number of boundaries found in the same window of text in the reference segmentation. Pk and WindowDiff values increase in case of over- or under-segmentation, and decrease for improved segmentation. The evaluation metrics will be discussed in Section 4.2.

3. METHOD

Recently, features extracted through deep neural networks have gained widespread interest thanks to their very competitive performance in a large range of applications, especially in image processing and natural language processing. So we intend to use well-performing pre-trained models for video frames feature extraction and textual feature representation.

Our processing workflow is organized as follows. First, the video is analyzed into frames and split into shots thanks to a shot boundary detection method. Frame-level visual features are then computed and aggregated for each detected shot. At the same time, the textual features are generated from the subtitles for each shot. The temporal information of each shot, both the starting and ending time are also considered to help the clustering method consider the closeness of the shots. The features from each modality are combined in different ways. An inter-shots similarity matrix is computed, based on the resulting features, and allows for a shot-based threading algorithm to assign a cluster to each shot. Like C99 segmentation technique by Choi (2000), we also apply a ranking to the similarity matrix, where the rank is the number of neighbouring elements having a lower similarity value within a neighbourhood window of 5. Finally, adjacent shots are grouped into scenes using Algorithm 1. The whole general method is depicted in Fig. 1. The steps are discussed in detail in the following subsections.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

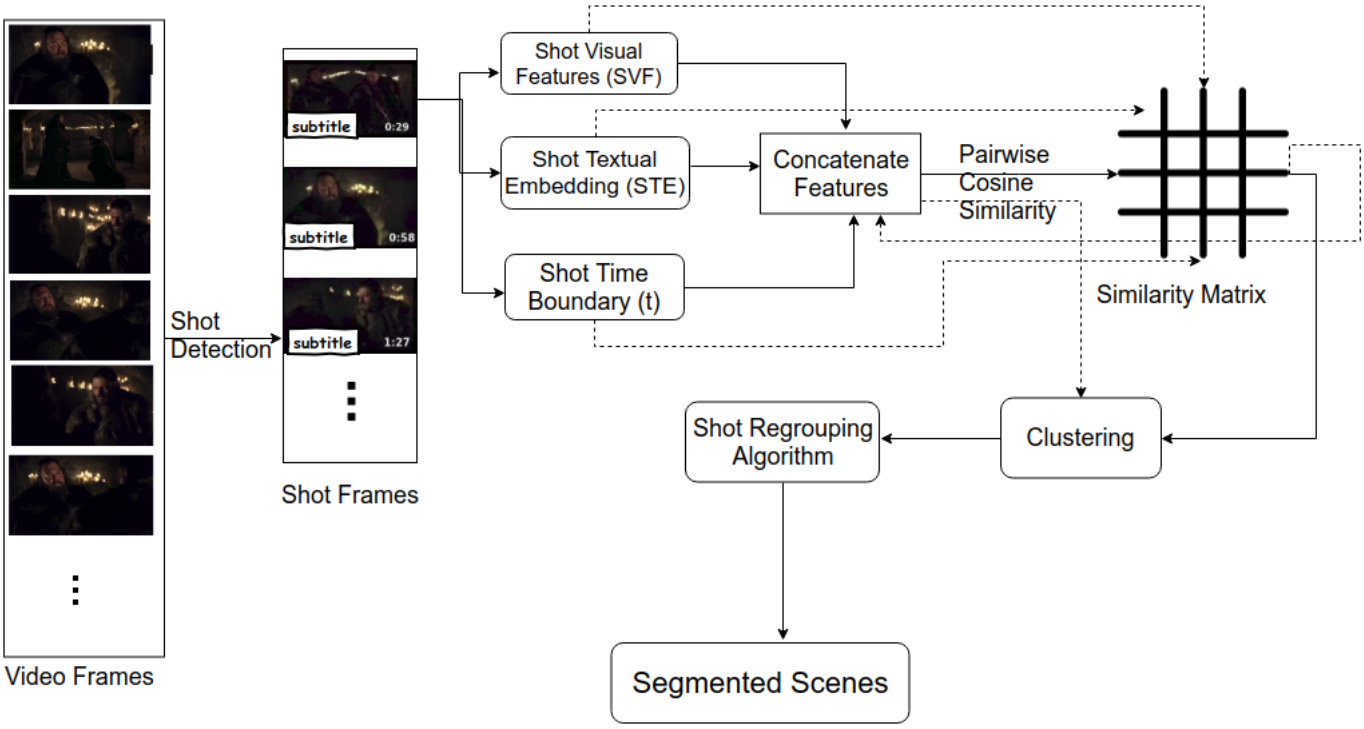

FIG. 1. SCENE SEGMENTATION METHOD

The broken lines show the late fusion of the features, where similarity matrices are computed for each feature set and then combined to be fed into the clustering module. The straight lines show early fusion of features, meaning that the similarity matrix is computed using the combination of the features.

3.1 **Shot detection**

We use a shot boundary detection (SBD) algorithm implemented in the open-source Pyannote-Video toolkit (Bredin 2015), which is based on displaced frame differences (DFF) and uses landmark features of the frames. It depends on some hyper parameters like the frame height, the duration of context window and a threshold on the similarity measures between shots.

We use a manually annotated TV series corpus described in Bost (2016) to optimize the above parameters. Bost (2016) provided manually annotated shots of the first seasons from each of the TV series reported in Table 1. The shot detection technique had an average shift of 0.04 seconds for the correctly detected shot boundaries, which is equivalent to one frame per shot. The accuracy of the optimized automatic SBD is 85% and 81% for *Breaking Bad* (2008-2013) and *Game of Thrones* (2011-2019), respectively.

3.2 **Features Extraction and Shot Representation**

The visual stream of a video V, is a sequence of frames, which can be further processed into a stream of visual features. In our experiments, we use the VGG16 pre-trained model provided by Simonyan and Zisserman (2014) to extract deep visual features for each frame and VGG16-places365 by Zhou et al. (2017) to extract scene's features of a frame. Thus, for the set of frames belonging to each shot, visual features are extracted following the above method; we refer to them as Shots Visual Features (SVF)[1].

The subtitles of the audio stream of a video also carry important semantic information. Therefore, we built a word2vec model for the word representation of each

1 SVF is the shot visual features, we have used CFF which is the central frame shot features. SVF and CFF can be used interchangeably in this document.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

word in the subtitle using the well-known Gensim word-2vec model from Řehůřek and Sojka (2010). We compute the textual features of a shot, in the same way as the visual feature, and refer to it as Shot Text Embedding (STE), using a word-embedding model built using all the subtitles of the respective TV series. In the case of *Game of Thrones* (2011-2019), we also use the text of the books and the pre-trained Gensim word2vec model to build our own word embedding model.

Furthermore, we add the temporal information of the shots for closeness by taking the start and end time of a shot. Consecutive shots that have short duration may have less time difference, therefore the temporal information will help to capture that and in return helps the clustering part. The temporal feature is normalized with regard to the total length of the video in order to present values between 0 and 1.

3.3 **Feature Selection and Augmentation**

While the length of each shot is variable, shot-level features have a fixed dimension. In our experiments, the dimensions for a video are as follows: Nx25088 for the SVF, Nx300 for the STE and Nx1 for the temporal feature, where N is the number of shots detected from the video. Given the variable number of frames within a shot, we test two aggregation methods for combining the frame-level visual features into fixed-size shot-level features, but taking the central frame of a shot performs better. Then we flatten the frame features and get the dimension of Nx25088. Next, we augment all the shot features as depicted in equation 1.

$$F(S) = [f(Fi)] \oplus E(S) \oplus time(S) \quad (1)$$

Where n is the total number of frames in shot S, i refers to a frame in the shot S, the $[f(Fi)]$ refers to the selected shot features and f is a function representing the deep features (features extracted from VGG16 or VGG16-places365 pre-trained model) of shot i and E is the text embedding (features) of shot S, the $\oplus$ operation represents the concatenation operation of the features. We have tried averaging the SVF and taking the average of the frame features inside a shot, by replacing $[f(Fi)] as \left[\frac{1}{n}\sum_{i=1}^{n} f(Fi)\right]$ in equation 1. The features subsampling is performed by selecting M frames within a shot. We test both random and uniform sub-sampling; in the latter case we use a step value S where S = N/M. Though, we perform the above combination of features, taking the central frame of a shot and extracting its features performed better and computed faster than averaging and subsampling of the frames of a shot.

3.4 **Shot threading**

Shot threading is important because a scene typically consists of an intertwining of shots, with alternate points of view on the characters and on the set. Thus, shot threading is a meaningful intermediate step between the shot segmentation and the scene segmentation, rather than directly clustering the shots into scenes.

With the concatenated shot features F(S) obtained so far, which is a late fusion of shot features, we compute a similarity measure between each pair of shots using the cosine distance and build the inter-shots similarity matrix. On the other hand, we also perform early fusion of features and then compute the similarity matrix of the shots. We compare three different clustering algorithms, i.e., K-Means, Spectral clustering and Affinity propagation. We report the results using these algorithms on Tables 2 and 3.

3.5 **Shot Grouping**

We propose the following, Algorithm 1 to group the shots labeled after the shot threading together into scenes. The motivation behind it is the fact that the result of shot threading is a sequence of labeled shots and at this stage, we can consider the scene segmentation problem as a problem of grouping sequence of adjacent shots together (Vendrig and Worring 2002), with the objective of maximizing the coherence of the resulting segment.

The algorithm performs the grouping of a sequence of shot threads based on the parameters K and C (where K is the sliding window size which is used to slide through the sequence of shot threads and C is the number of different shot threads) into a set of similar threads which are the scene or logical story unit. To our knowledge this algorithm is original, even if sequence grouping algorithms were proposed for other tasks like Vendrig & Worring (2002), which was motivated by biological sequence alignment of proteins, but ours presents a lower complexity.

Figure 2 depicts an example of sequence grouping into scenes, to illustrate algorithm 1. In this example, we have 6 clusters (c1 – c6) for the 15 shots. The K and C values are set to 3. The window slides until the end and the algorithm checks the number of different clusters (C) inside

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

ALGORITHM 1. SHOTS SEQUENCE GROUPING

```
procedure SCENE_SEGMENTATION (S, K, C)
S: sequence of labels,. K: window size,.
C: number of different speakers
tempList ← S[0 : 2] . initialize tempList
by the first 2 labels from S
sepPos ← [] detected scene boundaries
count ← 0
if len(S) <= 2 then
return sequence too short
else
for i in range(2,len(S)) do
        if S[i] = tempList[–1] then
                continue
        else if S[i] in tempList then
                tempList.pop(0)
                tempList.append(S[i])
        else
                count ← count + 1
                if len(tempList) < K then
                        tempList.append(S[i])
                end if
        if count = C then
                sepPos.append(i – C)
                count ← 0
        end if
    end if
  end for
end if
return sepPos
end procedure
```

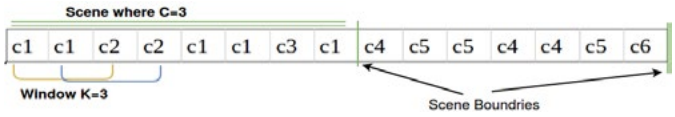

FIG. 2. EXAMPLE OF SHOTS GROUPING INTO SCENE

the window (size K). Therefore, according to the algorithm, the shots are segmented into two scenes as can be seen on Fig. 2.

4. EXPERIMENTS

We perform weakly supervised video scene segmentation of TV series using the techniques discussed in Section 3, comparing various clustering algorithms and different features as it will be discussed in Section 4.3. First, we present the dataset and the experimental setup.

4.1 Dataset and experiment setup

The dataset consists of 2 TV series – *Game of Thrones* and *Breaking Bad* – which we believe have complex and intertwined scenes and stories. We use the first 5 seasons of *Game of Thrones* and the first 3 seasons of *Breaking Bad*. For both series, we evaluate our systems using the manual annotation into shots for the first season and the manual annotation into scenes for all the dataset provided by Bost (2016)[2].

The data is split into a development and a test subset, as shown in Table 1. The first 3 seasons of *Game of Thrones* and the first 2 seasons of *Breaking Bad* are used as development set and the rest of the data for each series is used as test data[3]. We use the shot detection method discussed in Section 3.1 and evaluate its performance on the manual shot annotation provided by Bost (2016). This resulted in an accuracy above 85% and 81% for *Breaking Bad* and *Game of Thrones* respectively, which we found reliable enough for further processing.

TABLE 1. CONTENT OF THE DEVELOPMENT AND TEST DATASETS

Development Dataset

	Game of Thrones		*Breaking Bad*	
	Quantity	Average Time (h/m/s)	Quantity	Average Time (h/m/s)
Season	3	7.1h	2	7.6h
Episode	27	46.07m	18	45.7m
Scene	753	126.9s	459	120s
Shots	27396	3.4s	10814	5.1s

Test Dataset

	Game of Thrones		*Breaking Bad*	
	Quantity	Average Time (h/m/s)	Quantity	Average Time (h/m/s)
Season	2	7.9h	1	9.8h
Episode	20	46.07m	13	45.7m
Scene	460	140s	270	131s
Shots	17913	3.5s	5875	6s

2 Dataset: https://ndownloader.figshare.com/articles/3471839/versions/3

3 There are some missing episodes. In *Game of* Thrones, Season 02 Episodes 03 and 09, and Season 04 Episode 01; in *Breaking Bad* Season 01 Episode 05.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

4.2 **Evaluation Metrics**

As many papers have used the coverage and purity clustering evaluation metrics, we have reported purity and coverage measures in our tables.

We also use the frequently used topic segmentation metrics, WindowDiff and Pk. Both WindowDiff and Pk use a sliding window over the segmentation; each window is evaluated as correct or incorrect or as true or false. Equation 2 and Equation 3 show how the Pk and WindowDiff are computed, respectively.

$$P_k = \frac{1}{N-k}\sum_{i=1}^{N-k} f\big(f(ref_i, ref_{i+k}), f(hyp_i, hyp_{i+k})\big) \quad (2)$$

where ref and hyp are the manual segmentation (ground truth) and automatic segmentation, respectively. N is the total number of shots of an episode. k is the window size which is set to half of the average true segment size, according to Beeferman et al. (1997). In our case, we have set to 20 for *Game of Thrones* and 11 for *Breaking Bad*. The function f is 1 if the arguments are equal and a 0 if not.

Pevzner & Hearst (2002) claimed that Pk is unintuitive and come up with WindowDiff. WindowDiff is an amended metric of P_k, as can be seen in Equation 3.

$$WindowDiff(ref, hyp) = \frac{1}{N-k}\sum_{i=1}^{N-k}(|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0) \quad (3)$$

where b(i,j) represents the number of boundaries between positions i and j. The remaining symbols are the same with the above P_k symbols.

We compute recall and precision measures at shot level[4] of the scene segmentation of the video. Recall is the fraction of correctly grouped shots over the total amount of correct shot levels. Precision is the fraction of correct shots among all correctly grouped shots. Both are combined into the F1-score, defined as the harmonic average of precision and recall.

$$Recall = \frac{\#correctlydetectedshotsofhyp}{|ref|} \quad (4)$$

$$Precision = \frac{\#correctlydetectedshotsofhyp}{|hyp|} \quad (5)$$

where $|ref|$ is the total number of shots inside a boundary of the reference (ground truth boundaries) and $|hyp|$ is hypothesis (automatically segmented boundaries).

The shot level recall and precision may not indicate how good the scene segmentation is. Since a single frame (0.4s) or a single shot shift from the ground truth may cause an incorrect boundary at the scene level. As Baraldi, Grana and Cucchiara (2015) stated, "precision and recall fail to convey the true perception of an error". Therefore, we have introduced tolerance to the recall and precision measurements at a scene level. We set a tolerance of 3 shots[5] to the left or to the right of the boundary. In this shot tolerance, the boundary of a scene automatically generated will be considered as correct, if its boundary is 3 shots away from the ground truth.

4.3 **Results**

We compare the different shot features and their multimodal fusion. The results presented in Table 2 are based on central frame features (CFF) of a shot considering late fusion (combining the features after we compute the similarity matrix) and Table 3 shows the comparison of segmentation results with Bost (2016) work using the same data. The features show good performance with all metrics and for all clustering algorithms. The Kmeans and spectral clustering give the best results. In case of affinity propagation, the number of clusters is set by the algorithm itself and it may result in a large number of clusters. Thus the result of affinity propagation varies from episode to episode, whereas the spectral and Kmeans clustering are set to 40 clusters as optimized using the development dataset and its result is stable.

TABLE 2: RESULTS OF SCENE SEGMENTATION

using CFF, TSE and temporal information (t) based on early fusion, with the similarity matrix computed from the combined features

Game of Thrones

Clustering	WinDiff	Pk	Coverage	Purity	Recall	Precision	F1
Spectral	**0.03**	**0.01**	0.49	**0.91**	0.53	0.29	0.39
Kmeans	0.03	0.01	**0.61**	0.86	0.55	**0.47**	**0.51**
Affinity	0.03	0.01	0.43	0.91	0.50	0.23	0.31

Breaking Bad

Clustering	WinDiff	Pk	Coverage	Purity	Recall	Precision	F1
Spectral	0.07	0.04	0.61	**0.82**	0.59	0.51	0.54
Kmeans	**0.05**	**0.03**	0.63	0.80	**0.61**	**0.53**	**0.57**
Affinity	0.07	0.05	**0.65**	0.78	0.51	0.48	0.49

4 Shot level precision and recall around 0.95 are very high values but they don't guarantee that a scene boundary is correctly detected.

5 Using this 3 shot tolerance, the average distance between automatically generated boundaries and ground truth equal 16.5 and 15.4 seconds (with our best system), for *Game of Thrones* and *Breaking Bad*, respectively.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

The results on Table 3 shows that the method we use, CFF frames of VGG 16 and Kmeans clustering, performs better than Bost (2016). We have also compared our method with C99[6] by Choi (2000), one of the famous segmentation techniques, and still our method performs better.

TABLE 3. COMPARISON OF RESULT OF SCENE SEGMENTATION BETWEEN BOST (2016) AND OUR METHOD ON THE SAME DATA

	Bost (2016)			**Our Method**		
Game of Thrones	Recall	Precision	F1	Recall	Precision	F1
S01E01	0.47	0.20	0.28	0.41	0.30	0.35
S01E02	0.64	0.23	0.34	0.59	0.34	0.43
S01E03	0.72	0.27	0.39	0.73	0.32	0.45
Breaking Bad	Recall	Precision	F1	Recall	Precision	F1
S01E01	0.76	0.24	0.36	0.53	0.37	0.44
S01E02	0.56	0.14	0.22	0.53	0.29	0.38
S01E03	0.55	0.01	0.17	0.50	0.15	0.25

We compare all the features and their impact on the segmentation of a video into scenes. Table 4 compares the results of different features using the Kmeans clustering method, which show consistently good results for most of the features. CFF is the central frame features of a shot, STE is the text embedding of the shot and t is the timing of the shot in the video. The "VGG-CFF_Rank" are the features after ranking has been performed on the similarity matrix of the central frame features of the shot extracted from VGG16 deep pretrained model and we use VGG16-places365[7] features in our experiments. We also investigate the results of our method with and without augmentation with the temporal information. Table 4 shows the results using the features.

The method performs a little bit better in *Breaking Bad*, though we have used more data for *Game of Thrones*. In *Breaking Bad*, the combination of CFF and STF increases the results but the augmentation of temporal information did not show any significant improvement. In *Game of Thrones*, the combination of CFF, STF and temporal information (t) gives the best result. However, CFF alone performs better than combined with textual features. This can be explained by the fact that in *Game of Thrones* many shots do not contain any speech and are therefore associated with 0 subtitles and 0 STF values[8].

Similarly to C99 (Choi, 2000), we compute the rank of the similarity matrices. Ranking seems to improve the coverage in *Game of Thrones* and coverage and purity in *Breaking Bad*. But it does not improve the other measures.

TABLE 4. RESULTS USING DIFFERENT FEATURES AND THEIR COMBINATIONS BASED ON SPECTRAL CLUSTERING

Game of Thrones

Features	WinDiff	Pk	Coverage	Purity	Recall	Precision	F1
VGG-CFF	**0.03**	**0.01**	0.57	**0.86**	**0.59**	0.44	0.50
VGG-CFF_Rank	0.03	0.01	0.60	0.86	0.55	0.41	0.47
VGG-CFF $\oplus$ STE	0.03	0.01	**0.64**	0.84	0.52	0.44	0.48
VGG-CFF_ Rank $\oplus$ STE	0.03	0.01	0.63	0.85	0.45	0.37	0.40
VGG-CFF $\oplus$ STE $\oplus$ t	0.03	0.01	0.61	0.86	0.55	**0.47**	**0.51**
VGG-CFF_ Rank $\oplus$ STE $\oplus$ t	0.03	0.01	0.63	0.85	0.52	0.43	0.47

Breaking Bad

Features	WinDiff	Pk	Coverage	Purity	Recall	Precision	F1
VGG-CFF	**0.05**	**0.03**	0.6	0.79	0.57	0.49	0.54
VGG-CFF_Rank	0.05	0.03	**0.64**	**0.81**	0.51	0.46	0.48
VGG-CFF $\oplus$ STE	0.05	0.03	0.63	0.80	**0.61**	**0.53**	**0.57**
VGG-CFF_ Rank $\oplus$ STE	0.05	0.03	0.63	0.79	0.54	0.47	0.50
VGG-CFF $\oplus$ STE $\oplus$ t	0.05	0.03	0.64	0.79	0.59	0.52	0.55
VGG-CFF_ Rank $\oplus$ STE $\oplus$ t	0.05	0.03	0.63	0.80	0.44	0.40	0.41

5. CONCLUSIONS

Using the multimodal features of an episode from a TV series, we have designed a method for automatic scene segmentation. Our method shows detecting the shots and using shot

6 C99 generates highly over segmented results, which make the recall very high and the precision very low and, in consequence, a very low F1 measure. The average recall and precision computed using C99 are 0.13 and 0.05 respectively and the F1 measure is 0.08 for season 01 episode 01 of *Game of Thrones*.

7 Features extracted using VGG16-places have quite similar results with VGG16 features which are reported on Table 4.

8 The percentage of shots without subtitles in the test data is equal to 57% in *Breaking Bad* and 66% in *Game of Thrones*.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

level features are helpful for this purpose. In this work we have used shot visual features (SVF) and shot textual embedding (STE), which are both deep features. Since our method is based on shots, and the shots are based on visual features, the SVF performs better when used alone or in combination of other modalities.

We have tried to show the quality of our work using different metrics discussed in Section 4.3. Our method shows a good WindowDiff and Pk which shows the segmentation performs well. It also shows good purity values which can be interpreted as the grouping of shots into a scene is quite pure.

These preliminary results leave space for improvement. The textual features can be represented in a way comfortable for segmentation and in more suitable ways rather than just shot-based textual embeddings. Second, we believe that including audio features like music or prosodic information can improve the segmentation of a video into scenes. Speaker diarization can divide the audio of the video into segments according to the character's identity in a scene. Thus results from speaker diarization may also help improve the results of the scene segmentation by including each character in the sequence.

Scene segmentation is a fundamental task for many jobs that try to analyse and understand a video. In the near future, we plan to use our scene segmentation method for creating a meaningful link between scenes of the same episode and with scenes of other episodes of the same TV series. The link created between scenes will help to extract the parallel and intertwined stories of a TV series. There are different kinds of relationships between scenes that try to convey a message or tell a story to the audience. Our method is good enough to be used as scene segmentation whenever we want to do some scene-based analysis of a video.

In our future work we will be dealing with the extraction and the description of the narrative structure from TV series. The scene linking that will be created using our segmentation method will be helpful to understand the main theme of a scene and then can lead us to extract and describe the narrative structure of TV series. In the case of TV series, the narrative elements follow the same sequence in almost all series. Most episodes have a similar structure and each narrative element is located within a scene. A scene can have one or more of the basic acts of a narrative structure which will capture the important elements of narratives in TV series.

REFERENCES

Baraldi, Lorenzo, Costantino Grana and Rita Cucchiara (2015). "A Deep Siamese Network For Scene Detection In Broadcast Videos." Association for Computing Machinery International Conference on Multimedia. http://dx.doi.org/10.1145/2733373.2806316.

Bost, Xavier. (2016). *A Storytelling Machine?: Automatic Video Summarization: The Case of TV Series*. Doctoral dissertation. Avignon: Université d'Avignon.

Beeferman, Doug, Adam Berger and John Lafferty, J. (1997). "Text Segmentation Using Exponential Models." Computing Research Repository. https://arxiv.org/abs/cmp-lg/9706016v3 (last accessed 26-06-19).

Bredin, H. (2015). Pyannote Video: A Toolkit For Shot Detection, Shot Threading And Face Trucking. https://github.com/pyannote/pyannote-video (last accessed 30-04-19).

Chasanis, Vasileios T., Aristidis C. Likas and Nikolaos P. Galatsanos, N. P. (2009). "Scene Detection In Videos Using Shot clustering And Sequence Alignment." *Institute of Electrical and Electronics Engineers Transactions on Multimedia* 11(1): 89-100. https://doi.org/10.1109/TMM.2008.2008924.

Choi, Freddy Y. Y. (2000). "Advances in domain independent linear text segmentation." In Proceedings of North American Chapter of the Association for Computational Linguistics. https://dl.acm.org/citation.cfm?id=974309 (last accessed 26-06-19).

Del Fabro, Manfred and Laszlo Böszörmenyi (2013). "State-of-the-art And Future Challenges In Video Scene Detection: A Survey". *Multimedia Systems* 19(5): 427-54. https://doi.org/10.1007/s00530-013-0306-4.

Ercolessi, Philippe et al. (2011). "Segmenting TV Series Into Scenes Using Speaker Diarization." WIAMIS 2011, 12th International Workshop on Image Analysis for Multimedia Interactive Services, 2011, Delft, Netherlands. https://hal.archives-ouvertes.fr/hal-01987819 (last accessed 26-06-2019).

Guinaudeau, Camille, Guillaume Gravier and Pascale Sébillot (2012). "Enhancing Lexical Cohesion Measure with Confidence Measures, Semantic Relations and Language Model Interpolation for Multimedia Spoken Content Topic Segmentation." *Computer Speech & Language* 26(2): 90-104. https://doi.org/10.1016/j.csl.2011.06.002.

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X

Kumar, Niraj et al. (2011). "Video Scene Segmentation with a Semantic Similarity." Indian International Conference on Artificial Intelligence.

Pevzner, Lev and Marti A. Hearst (2002). "A Critique and Improvement of an Evaluation Metric for Text Segmentation." *Computational Linguistics* 28(1): 19-36. https://doi.org/10.1162/089120102317341756.

Protasov, Stanislav (2018). "Using Deep Features for Video Scene Detection and Annotation." *Signal, Image and Video Processing* 12(5): 991-9. https://doi.org/10.1007/s11760-018-1244-6.

Řehůřek, Radim and Petr Sojka (2010). "Software Framework for Topic Modelling with Large Corpora.". In Proceedings of the International Conference on Language Resources and Evaluation Workshop on New Challenges for NLP Frameworks. http://www.fi.muni.cz/usr/sojka/papers/lrec2010-rehurek-sojka.pdf (last accessed 26-06-19).

Scaiano, Martin and Diana Inkpen (2012). "Getting More from Segmentation Evaluation." In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technologies. https://www.aclweb.org/anthology/N12-1038 (last accessed 26-06-19).

Scaiano, Martin et al. (2010). "Automatic Text Segmentation for Movie Subtitles." Canadian Conference on Artificial Intelligence. https://doi.org/10.1007/978-3-642-13059-5_32.

Sidiropoulos, Panagiotis et al. (2009). "Multimodal Scene Segmentation Using Scene Transition Graphs.". In Proceedings of the Association for Computing Machinery International Conference on Multimedia. https://doi.org/10.1145/1631272.1631383.

Simonyan, Karen and Andrew Zisserman (2014). "Very Deep Convolutional Networks for Large-scale Image Recognition." International Conference on Learning Representations. https://arxiv.org/abs/1409.1556v6 (last accessed 26-06-19).

Tapaswi, Makarand, Martin Bäuml and Rainer Stiefelhagen (2014). "Storygraphs: Visualizing Character Interactions as a Timeline". Institute of Electrical and Electronics Engineers Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2014.111.

Todorov, Tzvetan (1977). *The Poetics of Prose*. Ithaca: Cornell University Press.

Utiyama, Masao and Hitoshi Isahara (2001). "A Statistical Model for Domain-independent Text Segmentation." In Proceedings of the Association for Computational Linguistics. http://dx.doi.org/10.3115/1073012.1073076.

Vendrig, Jeroen and Marcel Worring (2002). "Systematic Evaluation of Logical Story Unit Segmentation.". *Institute of Electrical and Electronics Engineers Transactions conference on Multimedia* 4(4):492-9. https://doi.org/10.1109/TMM.2002.802021.

Yeung, Minerva, Boon Lock Yeo and Bede Liu (1998). "Segmentation of Video by Clustering and Graph Analysis". *Computer Vision and Image Understanding* 71(1);94-109. https://doi.org/10.1006/cviu.1997.0628.

Zhou, Bolei (2017). "Places: A 10 million image database for scene recognition." *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence* 40(6): 1452-64. https://doi.org/10.1109/TPAMI.2017.2723009.

TV series cited

Game of Throne (2011-2019)
Breaking Bad (2008-2013)
The Big Bang Theory (2007-2019)
House of Cards (2013-2018)

DOI https://doi.org/10.6092/issn.2421-454X/8967
ISSN 2421-454X