Content-based unsupervised segmentation of recurrent TV programs using grammatical inference

Bingqing Qu, Félicien Vallet, Jean Carrive, Guillaume Gravier

Noname manuscript No.
(will be inserted by the editor)

Content-based unsupervised segmentation of recurrent TV programs using grammatical inference

Bingqing Qu · Félicien Vallet · Jean Carrive · Guillaume Gravier

Received: date / Accepted: date

Abstract TV program segmentation raised as a major topic in the last decade for the task of high quality indexing of multimedia content. Earlier studies of TV program segmentation are either highly supervised (e.g., event detection) or too specific to a certain type of program (e.g., cluster-based methods), which is not practically usable for indexing tasks because of the lack of generality of programs types. In this paper, we address the problem of unsupervised TV program segmentation by leveraging grammatical inference, i.e., discovering a common structural model shared by a collection of episodes of a recurrent TV program by finding an optimal alignment of structural elements across episodes. Structural elements referring to a video segment with a particular syntactic meaning with respect to the video structure. The use of symbolic representation of structural elements makes grammatical inference feasible to be applied on TV program modeling, and makes TV program segmentation possible to rely on only minimal domain knowledge. The proposed approach is operated in two phases. The first phase aims at obtaining a symbolic representation of each episode, where the elements relevant to the structure are discovered based on recurrence mining. The second phase is that of grammatical inference from the symbolic representation of episodes. We investigate two inference techniques, one based on multiple sequence alignment and one relying on uniform resampling, to infer structural grammars for TV programs. A model of the structure is derived from the structural grammars and used to predict the structure of new episodes. Comparative evaluation on two gram-

Bingqing Qu
E-mail: bingqing.qu@unifr.ch

Félicien Vallet
E-mail: fvallet@cnil.com

Jean Carrive
E-mail: jcarrive@ina.fr

Guillaume Gravier
E-mail: guig@irisa.fr

mar inference approaches demonstrates that the models obtained can reflect the structure of the program and predict the structure of unseen episodes, which is the main application of the proposed approach in industry, i.e., to assist librarians for segmentation tasks.

Keywords Multimedia mining · video segmentation · grammatical inference · multiple sequence alignment · uniform resampling · hierarchical clustering · experimental evaluations · practical applications

1 Introduction

The last decade has seen a rapid increase in multimedia content, with large scale audiovisual archives being made available for users and content providers. Consequently, data organization tools to efficiently manipulate and manage multimedia archives are needed. For instance, The French National Institute of Audiovisual (INA) is a broadcast archive institution that gathers, stores and shares professional multimedia content. INA has around fifteen million hours of radio and television programs stored. Such a large collection is useless in practice if content is not described and indexed so as to search for material and to provide easy access to a particular item. Based on this insight, a general description of a document (e.g., title, time, persons, summary) is no longer sufficient, and a video-fragment-level index describing fragments of content is needed, i.e., to facilitate information access or archive exploration. For instance, a piece of news described with just headlines is almost useless for retrieval usage, while information about the news generated through time (e.g., where can one find a scene) is indispensable. Thus, temporally segmenting documents into their constitutive parts (e.g., shots, scenes, ...), or temporally locating a given sequence or excerpt, is a crucial task of audiovisual content indexing in the context of large-scale audiovisual archives. Currently, audiovisual content segmentation in practice heavily depends on manual operations by librarians. According to INA's recorded data, segmenting and annotating correctly half an hour of news may take up to four hours, a significant part of that time being devoted to locating the start and end points of events of interest that librarian actually document. The segmentation step consumes important amounts of time and is costly in human resources, specially for a task where the skills of librarians are not essential. Therefore, automatic segmentation in recent years has been used to assist the manual operations, and can spare a lot of manual and tedious work.

The most traditional way to realize automatic TV program segmentation is either classification strategies [10,27] or event detection approaches [4,17], which are mostly supervised approaches. Unsupervised approaches for program segmentation were also addressed recently, where audiovisual consistency [5] and clustering-based methods [15] are considered. In particular, [5] proposed a multimodal event mining technique to discover repeating video segments exhibiting audio and visual consistency, and [15] clustered the keyframes based

on a statistical distance of Pearson's correlation coefficient to detect anchorperson shots. However, these approaches are not enough practical for librarians, because they are either highly supervised or too specific to a particular type of programs. Practically, there exist various programs of different types and adopting an automatic segmentation approach that can just deal with a small number of programs is not a wise choice. In this context, we address the challenge of an unsupervised automatic TV program segmentation approach that covers a wide range of programs, focusing on programs with multiple episodes regularly broadcasted, e.g., daily or weekly. Such programs are said to be *recurrent*. Recurrent programs are numerous on most TV channels, including news programs, magazines and entertainments. Let us take news program as an example: broadcast news usually start with a brief outline of the reports, followed by an alternation of anchorperson's announcement of the upcoming topic and of the corresponding news report; and most news programs end with interview segments, sports highlights or program teasers. We designate the constitutive elements of the program—i.e., headline, report, interview, trailer in the news example—as structural elements referring to a video segment with a particular syntactic meaning with respect to the video structure.

A key feature of recurrent programs is that, by construction, most episodes follow the same editorial structure: across episodes, the same structural elements are involved in almost the same order and with comparable duration. These two properties, namely, *repetitiveness* and *temporal stability* of structural elements, result from editorial choices, where the episodes of a recurrent program are constructed following a predefined temporal structure and a given principle of content that characterize the program. As a consequence, a stable temporal structure can usually be found for a recurrent program, whatever the type of program. The stability of the structure across episodes suggests to mine the underlying temporal structure of the program by working on a collection of episodes (i.e., a small quantity of episodes) to discover the regularities across episodes, infer the structure of the program and build the corresponding model. We propose to leverage grammatical inference to discover the logical organization of structural elements within a program from a number of unlabeled episodes and build a model of the structure of the program. A structural model of a program is composed of a number of structural elements, along with their temporal organization, temporal positions as well as presence probabilities. The interest in casting the task of structure modeling of recurrent TV programs as a grammatical inference task lies in the abundant literature on grammatical inference that offers a large choice of models to choose from and from which unsupervised inference can be made. Adopting grammatical inference techniques to multimedia content structure modeling however requires a symbolic representation of the programs that is suited for grammatical inference. Moreover, we require that this representation is obtained in an unsupervised manner so as to be applicable to a large variety of recurrent programs. The symbolic representation step is achieved thanks to the recurrence analysis and pattern mining across episodes combined with very limited domain knowledge. Showing that recurrent TV programs can be

turned into sequences of symbols in an unsupervised way using content-based analysis is also a key contribution that opens the door to the use of any symbolic knowledge discovery technique, of which grammatical inference is just an example.

In this paper, we leverage the grammatical inference techniques to achieve structure modeling task, which is conducted in two phases, i.e., structural element determination and structural grammar inference. First, we discover elements that are relevant to the structure of the program (e.g., jingles, typical separators, anchor shots) based on recurrence mining. Second, grammatical inference is leveraged to infer a structural grammar, then build a structural model for the program. Practically, building a structural model has specific meaning for TV program segmentation tasks in industry. By adopting a small quantity of episodes, without manual annotations, we generate a structural model of the program that can further be used. For instance, the model can be used as a structural reference for librarians to gain an overall understanding of the structure of a program and facilitate the identification of relevant parts to index. The model, having temporal information regarding the program, can also be used to actually segment episodes and save low-reward work for librarians. Note also that apart from TV archives, inference of a structural model could be used in other domains, e.g., to structure video captured by security camera in a public place, where regularities can be observed over days across similar time periods.

The rest of this paper is organized as follows. Previous work on TV program structuring is discussed in Section 2. Section 3 gives a global view of the proposed approach, before detailing the two main phases: Section 4 describes the determination of structural elements; Section 5 introduces the two grammatical inference techniques considered in this study. Experimental evaluations are reported in Section 6. At last, we discuss improvement and future research directions in Section 7 before concluding remarks in Section 8 .

2 Related work

TV program structuring focuses on structuring a certain program into its constitutive components. Based on whether prior knowledge of the program structure is adopted or not, previous work in the abundant literature on TV program structuring can be classified in two categories: prior-knowledge-based methods and prior-knowledge-free methods

Numerous studies use prior knowledge of the program structure to segment programs or to build models. For instance, [27,16] use extensive prior knowledge of sports and editing rules to model the structure of sport videos. [27] targets soccer videos and relies on low-level features to detect particular actions or states in soccer games. Hidden Markov models are used in [16] for tennis video structuring, relying on prior information about tennis video content and production rules. Some recent studies work on diving and jump games to recognize the player action by continuous hidden Markov models [19] or to de-

tect and represent player's action in panoramic background by RANSAC techniques [30]. A temporal structural model is used in [29] to identify the different news stories in broadcast, while [8] uses machine-learning-based techniques to classify the shots of news video into predefined categories, e.g., *anchor*, *interview*, *forcast*. [10] proposes a method for the automatic segmentation of TV news videos into stories, where a temporal context and machine learning methods are used to perform the story boundaries detection from multimodal features. There exists some work focusing only on speech recognition of TV programs using Deep Neural Networks (DNN), for example, [24] uses generalized discriminant analysis for acoustic feature extraction and [11] represents acoustic features by an i-vector before adopting DNN techniques. Most of these approaches are however highly supervised, requiring extensive prior knowledge of the program structure.

Some noticeable studies address the problem of program structuring with minimal, even without, prior knowledge of programs, shifting from supervised to unsupervised techniques for program structuring. For instance, a frequent pattern approach is used in [13] for anchorperson detection and other related purposes, where a matrix of similarity between different time points in a video is firstly computed and a K-means clustering is adopted to find patterns in the matrix. A recent piece of work [7] presents a robust framework to detect the anchorpersons for different types of programs by extracting speaker identity features from the audio data. Some methods focus on scene segmentation for various types of TV programs. For example, [23] proposes a novel approach to video temporal decomposition into semantic scenes jointly exploiting low-level and high-level features automatically extracted from the visual and the auditory channel, where a fast scene transition graph (STG) approximation and a generalized STG-based technique are proposed for multimodal scene segmentation. [31] focuses on grouping video content into semantic segments and classifying semantic scenes into different types based on the temporal constraint of video content and visual similarity between shot activities. A novel work [20] proposes a method for automatic storyboard segmentation of TV news using image retrieval techniques and content manipulation, where image re-ranking based on neighborhood relations and temporal variance of image locations is adopted to construct a unimodal cluster for anchorperson detection and differentiation.

Beyond traditional approaches, a few approaches target program structuring based on recurrence detection, saving the use of prior knowledge of programs. For example, event repetitiveness is leveraged by considering visual/audio recurrence in [1,2] to detect separators; [28] proposes a method to model and analyze video syntactical structure based on short video repeat detection. In our previous work [21,22], we also take advantage of the property of recurrence and the underlying structure of recurrent programs is explored with grammatical inference techniques.

Our contribution falls in the unsupervised category and builds on the seminal idea of [1] to exploit regularities in recurrent programs. Abduraman *et al*. focus on detecting a particular structural element called separator—i.e.,

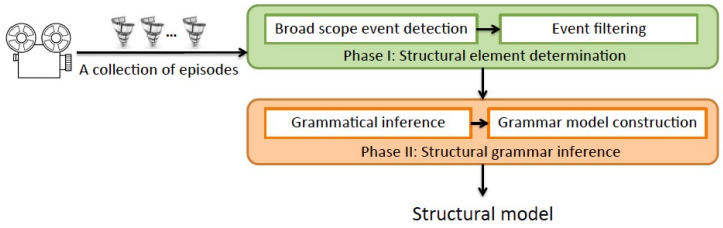


Fig. 1: Principle of grammatical inference for recurrent TV programs

short audio/visual sequences that appear before or after the events of interest to signal their starts or ends—considering repetitiveness across episodes. This results in an efficient method to structure recurrent programs that is however not practical enough for librarians for two main reasons: (a) the method is limited to the detection of separators and fails at adressing a number of situations, e.g., two consecutive structural elements with no separators, or identifying the common structure across episodes and the corresponding variations; (b) the method does not extend to unseen episodes that keep on arriving, not being able to create an actual model of a program or of a separator. We thus extend the seminal idea of [1], considering a variety of structural elements and building an actual model of the structure of the program so as to be able to segment future episodes.

3 Principle of the proposed approach

Our objective is to ultimately create a model of a recurrent TV program given a collection of episodes. In this scenario, we assume no prior knowledge on the structural elements that might be present in a recurrent program and very limited knowledge on the program type.

The principle of the proposed approach is illustrated in Figure 1, where two distinct phases are considered. The first phase identifies structural elements that are relevant to the program structure. The second phase infers a grammar and the corresponding model of the program considering the structural elements identified in the first phase. In the first phase, we first leverage a large number of audiovisual detectors to detect general events, i.e., audiovisual segments with basic features or information of video content, such as, monochrome image sequences, silence segments, etc. Some of these general events are relevant to the structure while others are not, where relevant events are those that occur in most of the episodes at about the same time instant. We thus filter general events based on their occurrences across episodes using temporal density analysis, retaining only the recurring ones. At last, the structural elements of a program are determined from the recurring events using limited domain knowledge, turning each episode into a sequence of symbols (a symbol is a particular structural element). Although recurrent TV programs exhibit temporal stability across episodes, differences still exist among differ-

ent episodes. To discover a common pattern shared across episodes, grammatical inference is leveraged to find an optimal alignment of structural elements across episodes, taking advantage of the symbolic representation obtained in the first phase. We consider two basic grammatical inference methods in this work, namely, multiple sequence alignment [21] and uniform resampling [22]. Finally, a structural model is constructed by adding temporal information to the structural elements that appear at different positions in the grammar obtained in the grammatical inference step. The interest of the model with respect to the grammar is its ability to process new episodes, e.g., to provide a segmentation that is usable by librarians.

While we are aware that far more elaborate grammatical inference methods do exist, we limited ourselves to basic methods as the goal of this work is first and foremost to demonstrate that the task of unsupervised modeling of recurrent TV programs can be addressed as a grammatical inference task. We also wanted to limit ourselves to state-based grammars from which it is easy to construct a model of the content.

4 Identifying structural elements

The first phase is to determine structural elements that are relevant to the structure of the program using a number of broad-scope audiovisual event detectors. With the main assumption that there is very limited knowledge about the type of program and no prior knowledge about the structural elements that may be present, determining structural elements must be performed in an unsupervised manner. To skirt the unsupervised issue, we search for elements and events that repeat across episodes with relative temporal stability, giving priority to the basic and common elements. Practically, we apply a large amount of audiovisual detectors on the collection of episodes of a program and select events relevant to the structure by analyzing their temporal distribution across episodes.

4.1 Broad scope event detection

To determine structural elements generic enough for various types of programs, a large number of event detectors should be adopted to detect general purpose events that may potentially be relevant to the program structure and from which we can identify structural elements easily with only minimal domain knowledge. Considering a trade-off between the type of programs, the genericity of the method and the complexity at run time, nine audiovisual event detectors are considered. Among them, seven are visual detectors: shot detector, dissolve detector, monochrome image detector, text region detector, motion activity detector, person clustering, as well as shot-reverse-shot detector. Besides, two audio detectors, i.e., speech/music/silence detector and audio recurrence detector, aim at detecting generic audio features for program structuring. More detailed descriptions of these event detectors can be found in [21,

22]. By applying all the above event detectors on the collection of episodes, we obtain the occurrences of each general event with their temporal positions across episodes.

A considerable amount of general events is detected in this first step, however, not all are relevant to the structure. For instance, a short sequence of black frames could indicate a separator inserted between two successive parts of the program. It could however also be found in a night scene. We thus need to select those events that are relevant to the program structure among the general events detected. The key idea that we exploit for the determination of the structural elements is that of repetitiveness. We therefore seek events that repeat across episodes and select those whose occurrences are concentrated around the same instant in most of the episodes.

4.2 Temporal density filtering

Repetitiveness is measured by means of temporal density filtering. However, before applying density filtering, a complementary strategy, i.e., role recognition, is performed to detect the dominant person of each episode, e.g., anchorperson or conductor, as it is an important feature of program structures.

4.2.1 Role recognition

Role recognition is adopted to further characterize the outcome of person clustering and identify the most important person of each episode, such as the conductor or the anchor, which is clearly a strong cue with respect to program structure. We use five measures to characterize each person cluster with the idea of finding the person that covers at best a significant amount of time in an episode [14]: total duration of appearance; total number of distinct appearances, i.e., number of non consecutive segments; duration of the longest segment in which the person appears; time range between the first and last occurrence; duration in which the speaker is engaged in a dialog. To account for varying episodes and program lengths, all five measures are scaled to $[0, 1]$. Decision on the dominant person is made based on the sum of the five normalized measures, the cluster having the maximal sum being identified as the dominant person. Once the dominant person of each episode is identified, the shots containing the dominant person are considered as a general event and are further analyzed using temporal density filtering, as all other general events.

4.2.2 Event filtering

The key of event filtering is to first find the events that repeat across episodes, i.e., *repeated events*, before selecting the occurrences of repeated events that have a high rate of occurrence across episodes at a given time. These segments, designated as *repeated occurrences*, are the occurrences significantly repeated and thus deemed relevant to the program structure.

Repeated event selection A prior filtering step is firstly adopted to remove the general events that do not exhibit the property of repetitiveness. In particular, a general event is deemed as not repeating across episodes, if one of the two cases occurs. First, the occurrences of a general event only come from a small minority of episodes, i.e., less than one third of the total number of episodes in the collection. For example, a sequence of monochrome images just found in one single episode is very unlikely to be relevant to the program structure. It is probably just a night scene. Second, a general event whose highest value of the temporal density (which will be detailed later) does not reach a given value, i.e., about 30% of episodes in the collection. For instance, the music sequence with its occurrences appearing in all episodes but not found in similar temporal positions across episodes, is also not considered as a repeated event. It may be just different scenes with musical accompaniment. We thus eliminate the general events exhibiting no repetitiveness, then analyze the temporal density of occurrences of a repeated event over the set of episodes to select the repeated occurrences.

Repeated occurrence selection For each type of repeated event, we project onto the same temporal axis its occurrences across episodes in the collection, and measure the empirical density of occurrences across episodes at very time instant. In other words, we estimate from all episodes the probability that a repeated event appears at a given time instant. Prior to density estimation, we normalize the length of each episode to a common value so as to account for variations in episodes lengths. Moreover, to avoid artifacts and account for slight time variations within episodes, a kernel-based density estimator is used in practice.

Formally, for an event, we compute at each time instant the density as

$$f(t;h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{t - t_i}{h}\right) \quad , \tag{1}$$

where n is the number of instances of the event considered, t_i is the time position of the i^{th} instance calculated as the mean of the start and end times of the i^{th} instance. K is a zero mean unit variance Gaussian kernel function whose optimal bandwidth h is automatically chosen as in [6]. In plain words, $f(t;h)$ measures how frequently an event occurs around time t across episodes. Detecting repeated occurrences is finally performed by thresholding $f(t;h)$, considering only frequent occurrences as structurally relevant and ignoring sporadic ones. The threshold is empirically set as a fraction of the mean of all the peak values in the density function, defined as:

$$\text{threshold} = \frac{\text{mean of peaks}}{p} \tag{2}$$

The selection of the threshold (i.e., the selection of parameter p) will be discussed in the experiment part in Section 6.2.1. Figure 2 illustrates temporal density filtering on a collection of episodes, depicting the occurrences of three detected general events (colored rectangles). The red and green events are the

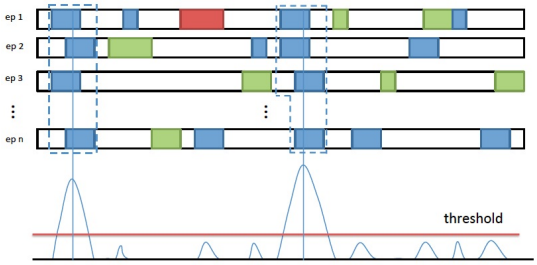


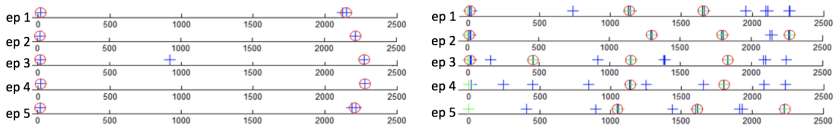
Fig. 2: Illustrative example of repeated events and repeated occurrences filtering

ones exhibiting no repetitiveness across episodes and need to be removed. The red event is found just in one episode and the occurrences of the green event do not appear in similar temporal positions across episodes. The blue event is a repeated event with its occurrences found in similar temporal positions across episodes. After applying the density filtering, the repeated occurrences of the blue event (enclosed in dashed boxes) are selected by thresholding $f(t; h)$.

The selected events do not have semantic characterization for program structures, so the next step is to endow the syntactic meaning with those events using minimal domain knowledge and identify the structural elements relevant to the program structures.

4.3 Structural element identification

From the selected repeated events, we identify the structural elements of a program using a set of rules based on minimal common domain knowledge in TV programs. As opposed to repeated events that have no particular meaning per se, structural elements are syntactic units that compose the different parts of the program, similar to words in a grammatical inference task. For instance, a structural element corresponding to a sequence of white images (the event) is a separator (the structural element), while a long duration shot containing the dominant person at the beginning of the program (event) is deemed as an anchorperson's opening (structural element). A realistic example is given in Figure 3(a). For NEWS, monochrome images were found to be often around the same temporal positions. Additionally, two short sequences of monochrome images are found in each episode, resp. at the beginning and end of the episode. By leveraging domain knowledge in TV programs, we deem the repeated short sequences of monochrome images as separators. There are cases where we need more than one repeated event to identify a structural element. Commercials is a typical such structural element, identified by jointly considering three repeated events, i.e., silences, monochrome images and shots with short duration. For example, in Figure 3(b), for GAME monochrome frames and dissolve transitions are repeated almost at the same temporal positions across episodes, so we considered these two repeated events jointly occurring



(a) NEWS (b) GAME

Fig. 3: Examples of separators for five different episodes of GAME and NEWS, where "+" in green (resp. blue) represents monochrome images (resp. dissolve) and "o" represents separators

as separator for GAME. More details of structural element identification will be further reported in the experiment section. After the identification of structural elements, each is represented as an alphabet symbol, e.g., S for separators and C for commercials.

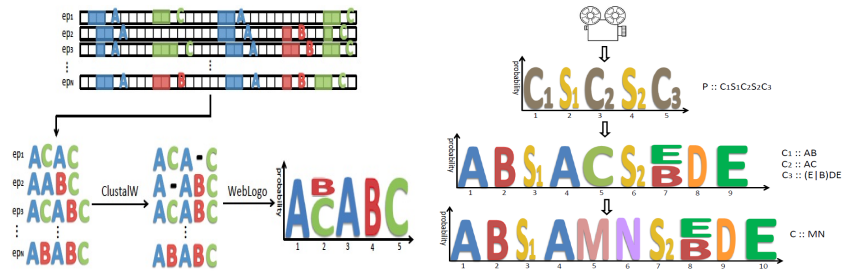
We emphasize that the identification of structural elements is the only step in the whole process of grammatical inference where domain knowledge is required. Adopting minimal prior knowledge in this step makes it possible to apply the proposed method on numerous categories of programs. Furthermore, the symbolization of structural elements provides a possible way to adopt various symbolic data mining algorithms on a collection of TV programs, e.g., to infer shared patterns among episodes, or to classify or cluster episodes, etc. Grammatical inference, as considered in this paper, is only one example among others of the opportunities offered by the lightly supervised symbolization of recurring episodes.

5 Building the structural grammar model

Given a set of structural elements and their occurrences across episodes, the next phase is to infer a structural model for the program, the key of which is to discover a common structure shared by the episodes. Using grammatical inference techniques, a structural grammar is firstly inferred for a program, based on which a corresponding structural model can be constructed by combining the temporal boundaries of structural elements and their presence probabilities, so as to be utilized in practical, e.g., comparing different inference methods or structuring upcoming episodes of the same program.

5.1 Grammatical inference

Although the temporal stability of recurrent TV program results in similar episodes in terms of the structure, slight differences still exist among different episodes. Therefore, the key idea of grammatical inference is to find an optimal alignment of the structural elements between episodes. The alignment can be done in various ways. In this paper, the two methods that we previously introduced, i.e., multiple sequence alignment [21] and uniform resampling [22],



(a) Multiple sequence alignment from a collection of episodes

(b) Hierarchical architecture for multiple sequence alignment

Fig. 4: Illustration of grammatical inference by multiple sequence alignment

are considered. We briefly recall the underlying principles of each method and discuss their pros and cons in contrast. Readers are referred to the seminal papers for more details. Note that other grammatical inference methods could be used with very limited adaptation.

5.1.1 Multiple sequence alignment

Multiple sequence alignment techniques, originally designed to align any set of symbolic sequences, clearly serve our case, i.e., aligning a collection of episodes labeled with structural elements. We used ClustalW [25], a general purpose alignment tool from the field of bioinformatics that can perfectly align a set of sequences with different length. Alignment of the symbolic sequences is done in the way that alphabet symbols in a given position are homologous, superposable or play a common functional role, thus allowing to derive a shared pattern of the episodes from the aligned sequences. The process of multiple sequence alignment is illustrated in Figure 4(a), where three structural elements are identified and represented by different symbols, i.e., A, B, C. In order to visualize the aligned episodes and have a concise understanding of the program structure, the WebLogo [9] representation is adopted. A stack of symbols is used to illustrate each position in the grammar: the height of objects within the stack indicates the relative frequency of each symbol while the stack width is proportional to the fraction of valid symbols in that position.

Multiple sequence alignment can be effectively used for modeling recurrent programs with concise structure, i.e., the ones with short duration or limited numbers of structural elements. However it may fail in the case of programs with more complex structures, i.e., the ones with long duration or a large number of structural elements. The reason is obviously that increasing the number of structural elements and the structure complexity also increases ambiguity in the alignment process. To reduce this ambiguity, the whole procedure can be repeated in a hierarchical manner to yield complex grammars with tractable computational burden. The idea of the hierarchical method is that multiple sequence alignment is independently applied to short segments

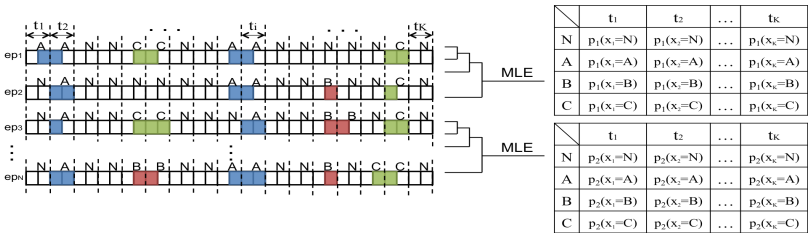


Fig. 5: Illustration of grammatical inference by uniform resampling

of a program. The key point of the idea is to replace aligning long sequences of entire programs by aligning short sequences of certain segments of a program. We rely on separators to divide a program into chapters, i.e., the set of segments between two separators, because common separators can be easily identified, e.g., monochrome images and short repeated audiovisual sequence. This hierarchical architecture is illustrated in Figure 4(b). Adopting multiple sequence alignment, a coarse-grain structure is first obtained considering only separators (S_i) and chapters (C_i). For a structure at a finer grain, multiple sequence alignment is applied independently to each chapter, i.e., considering the sequences of symbols that belong to the same chapter across episodes. This hierarchical architecture allows facilitating the alignment for programs with complex structures and obtaining structural grammars with few ambiguities.

5.1.2 Uniform resampling

Practically, some recurrent programs have not only one structure, but can rather have multiple structures, e.g., depending on the day of the week or on the phase of a game. Considering the TV news example: The days when there are invited people, the show usually ends with the interview, while the days when a new film is released, the show usually ends with the film trailer. Multiple structures can mislead sequence alignment and therefore need be detected before inferring the program structural grammar and the corresponding model. A rather straightforward way to identify multiple structures is to run clustering on the episodes. However, clustering with multiple sequence alignment is costly and difficult, in particular because episodes do not have the same length, thus requiring numerous dynamic alignments between sequences. To circumvent this high computational cost, we propose to infer the structural grammar in the way of uniform resampling.

The general idea of the uniform resampling is illustrated in Figure 5. The episodes labeled with structural elements are segmented into a fixed number of time intervals after normalizing the length of the episode. Each time interval is represented by the symbol of the corresponding structural element, if any, present in the interval. Arbitrarily, the specially defined symbol N denotes the absence of any structural element. After turning a collection of episodes into a set of fixed-length sequences, the next step is to find a common structure shared by the sequences. To do this, we first assume that a program has a

unique structure shared by all episodes, before moving to the step of multiple structure identification. For each time interval, we compute the probability distribution of each possible structural element using maximum likelihood estimation (MLE). Therefore, we can obtain a distribution matrix to represent the structural grammar of the program. In the practical case of multiple structures, we identify multiple structures by a clustering technique using a hierarchical agglomerative clustering technique to group episodes [22].

5.2 Structural model construction

The previously inferred structural grammars are essentially the aligned symbolic sequences, which are not usable in practice to segment new episodes. Hence, we propose to construct structural grammar *models* based on the structural grammars obtained with different inference methods. The constructed structural model should represent all the identified structural elements with their sequential order, the duration of the elements as well as their presence probabilities. With all these elements, models do not only provide a general understanding of the program structure, but can also be utilized in practical use. In this paper, we consider a segmentation use-case where unseen episodes are automatically segmented into their constitutive structural elements using the model inferred from the result of grammatical inference. Note that in this work, models do not account for content-based features. The main reason for this is that visual and audio content may vary significantly from one episode to the other, thus making it difficult to build a content-aware model. The lack of determinism to map (repeated) events to structural elements and the difficulty to provide a stochastic model of this mapping also governed this choice.

Formally, a structural model is defined as follows:

$$\begin{cases} \mathbf{E} = \{E_1, E_2, ..., E_i, ..., E_m\},\ where\ E_i \in \mathbf{S}^d \\ \mathbf{T} = \{\mathbf{T_1}, \mathbf{T_2}, ..., \mathbf{T_i}, ..., \mathbf{T_m}\},\ where\ \mathbf{T_i} = \{T_{start}, T_{end}\} \\ \mathbf{P} = \{\mathbf{P_1}, \mathbf{P_2}, ..., \mathbf{P_i}, ..., \mathbf{P_m}\},\ where\ \mathbf{P_i} = \{P_1, P_2 ..., P_d\} \end{cases} \tag{3}$$

where $\mathbf{E}$ is a sequence of structural elements composing the program, and m represents the number of positions containing identified structural elements in the model. $\mathbf{S}$ is a set of symbols representing the identified structural elements of the program, and d is the number of the identified structural element types present in the model. $\mathbf{T}$ is the temporal positions of the structural elements present in the model. Particularly, $\mathbf{T_i} = \{T_{start}, T_{end}\}$ is the start and end instants of the structural element at a given position. $\mathbf{P}$ is the presence probability of the structural elements, where $\mathbf{P_i} = \{P_1, P_2 ..., P_d\}$ is the presence probability of all possible of structural elements at a given position.

In Figure 6, we show two examples of the graphical representation of the structural models. We propose to use a rectangle filled with a symbol to illustrate each structural element for the program: The height of rectangles indicates the relative probability of each element while the width is proportional to its duration. Note that in spite of the fact that the structural grammar

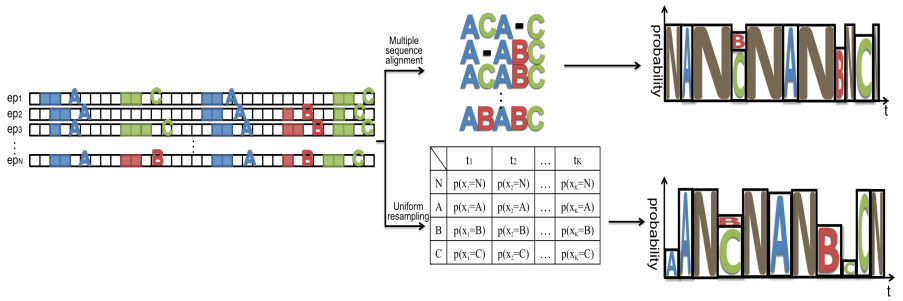


Fig. 6: Example of structural model construction

representation resembles the WebLogo representation, the two differ in their interpretation.

In the case of multiple sequence alignment grammars, the boundaries of each symbol in the grammar are taken as the average time positions of the occurrences of the elements in the same position across episodes. In the case of multiple elements present in one position, we do not differentiate them: the start and end time instants of the position are obtained considering all the structural elements in that position, regardless of their label. Evidently, as in Figure 6, adjacent elements are not necessarily contiguous because of segments with no particular semantic interpretation appearing between two structural elements. In practice, we consider such segments as a particular structural element denoted by the symbol N. The presence probability of each element in the same position is computed separately by counting the number of occurrences contributing to the same element. In the example illustrated in Figure 6, there are three types of structural elements ($d = 3$), which are found in five positions ($m = 5$).

In the case of uniform resampling grammars, the grammar represented by a probability distribution matrix has very limited abstraction capabilities and is not concise enough owing to information redundancy, i.e., consecutive time intervals repeating with the same structural elements and similar probability distribution. We thus propose to separate the time intervals into states, where a state refers to consecutive time intervals having the same structural elements with similar probability distribution. To this end, we verify between two successive time intervals the variations of two indicators: The composition of structural elements and the Jensen-Shannon (JS) divergence, which measures the similarity between two probability distributions. If one of the two indicators has a significant change, we consider that there is a rupture of state. Specifically, the time intervals are segmented as coherent states based on the positions where the JS divergence has a local maximum value or where the combination of structural elements changes. Based on the segmented time intervals, we build a structural model. Since each state (i.e., each position in the model) may consist of several time intervals, the start time of the state is taken as the start time of the first interval in the state, while the end time

of the state is taken as the end time of the last interval in the state. The probability distribution of each state is computed as the average value of the probability distribution (the average value of each structural element is computed separately) of all time intervals. In the example illustrated in Figure 6, there are three types of structural elements ($d = 3$) found at seven positions ($m = 7$).

6 Experimental results

We conduct experiments on four different types of recurrent programs, viz., news, game, talk show and magazine. Firstly, we analyze the influence of the threshold applied on the density filter, as well as the size of the collection for inferring structural grammars. Secondly, we examine the performance of structural models relying on a segmentation use-case.

6.1 Data set description

Experiments make use of four recurrent programs of different types. Global statistics for each data type are provided in Table 1. *20h News* (NEWS) follows a very standard pattern for a daily news show. *Que le meilleur gagne* (GAME) is a game show having four parts divided by separators. The program, hosted by a conductor, mainly contains interview scenes and question/answer scenes with full text segments. The episodes of GAME were taken over two years (1991 and 1992). *Le grand journal* (TALK) is a recent talk show, whose episodes are taken from the first months of 2014. The talk show is hosted by a conductor and mainly contains news reports, talks, weather reports and musical performances, between which separators are inserted. The magazine *Telematin* (MAGZ) was taken with episodes selected from the year 1989. MAGZ is a morning program proposing news and topics about culture and daily life. Different topics of the program are separated by separators. For the four programs, we manually segmented and annotated the structural elements of each episode with their corresponding types and start/end times. We annotated program structures with an accuracy at the frame level.

For each program, the dataset comprises 24 episodes, divided into two sets: one set for inferring structural grammars (*inference set*), the other for the use-case application (*test set*). Due to the limited quantity of data and to avoid experimental biases, experiments are conducted using a cross-validation strategy: For each fold, part of the episodes is randomly selected for inference from the 24 episodes, the remaining ones being used for the segmentation use-case. The size of the inference set for the segmentation experiments will be fixed later, after a study on the influence of the number of episodes on grammar inference. Quantitative results reported hereunder are averaged over 5 folds.

Dataset	Date	Episodes	Type	Average duration	No. of element types in the annotations
NEWS	2007	24	TV news	37.9 m	3
TALK	2014	24	Talk show	71.3 m	3
GAME	1991 - 1992	24	Game	31.9 m	4
MAGZ	1989	24	Magazine	61.9 m	6

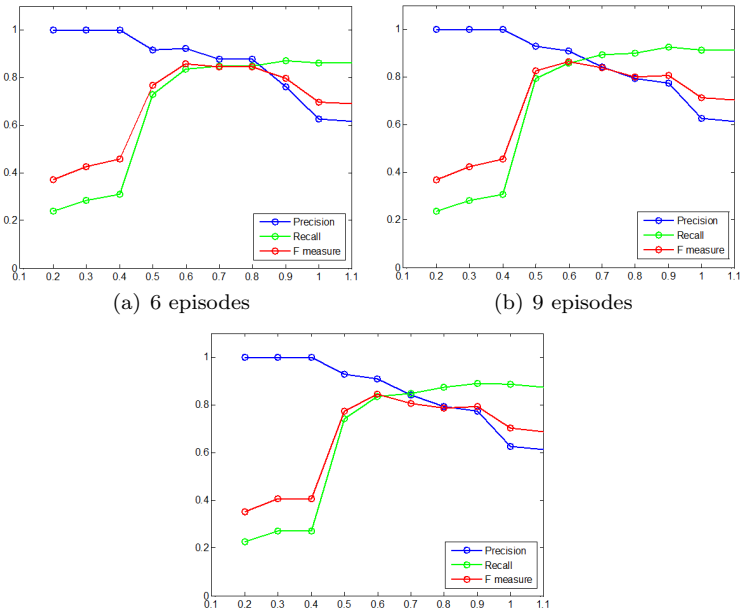
Table 1: Description of the datasets used for evaluation

6.2 Experiments on structural element determination

Two main factors impact the detection of repeated events and thus the obtention of structural elements. The first factor is the threshold applied on the temporal density function, the second one being the number of episodes on which the detection of repeated events operates. We study in turn these two factors.

6.2.1 Threshold of the density function

Thresholding the temporal density function relies on the threshold given in Equation 2, governed by parameter p where the greater the value of p, the more repeated events are found. Three evaluation metrics, i.e., precision, recall and F measure, are adopted to examine the influence that the parameter produces on the selected occurrences of repeated events. Precision measures the fraction of repeated occurrences composing the structural elements over the total number of occurrences selected by the given threshold. Recall measures the percentage of the repeated occurrences composing the structural elements over the repeated occurrences of structural element in the ground truth. In other words, we look at how accurate the repeated events found are to find the structural element of the ground truth. Results, averaged over the four data sets, are reported in Figure 7, considering different numbers of episodes to estimate the temporal density function. The vertical axis represents resp. precision, recall and F-measure while the horizontal axis shows the value of p. As shown, the increase of the parameter p (decrease of the threshold) results in an increase of recall and a decrease of precision, as more sporadic occurrences are involved. Consequently, F measure exhibits a decrease after an initial increase. We consider the F1 measure to provide a good trade-off in term of structural element determination. Based on the observations of different numbers of episodes, when the parameter p lies between 0.6 and 0.8, the F measure shows its highest values. Among them, when $p = 0.6$, the average of F measures stabilizes to a maximum value. Consequently, from now on, we will use $p = 0.6$ for the threshold for the temporal filters.



(a) 6 episodes (b) 9 episodes

(c) 12 episodes

Fig. 7: Precision, recall and F measure as a function of the parameter p, averaged over show types, for different number of episodes.

6.2.2 Size of the inference set

The size of the inference set strongly influences the number of repeated events as a direct consequence of the repetitiveness property. In order to choose the size of the inference set, we fix the threshold and analyze the effect that the number of episodes in the inference set may produce on the number of repeated events that is determined. Evidently, a very small amount of episodes is not sufficient to conduct density filtering with high confidence. On the contrary, repeated events may be drowned in a large number of episodes. Figure 8 reports the number of repeated events determined when varying the number of episodes involved in the density filtering step. For all four data sets, with a small quantity of episodes in the inference set, i.e., less than nine episodes, the number of events increases along with the number of episodes. For GAME and TALK, the number of repeated events tend to be stable as the amount of episodes increases above 9. For MAGZ, however, the number of events discovered drops down after 12 episodes. This can be explained by the fact that some events, such as the ones corresponding to separators, are very short audio sequences. In particular, the two drowned recurrent audio sequences have very short length (less than 5 seconds), so in the case of MAGZ they are drowned when the size of inference set having more than 11 episodes. Based on these observations, 11 or 12 episodes are chosen for the inference set.

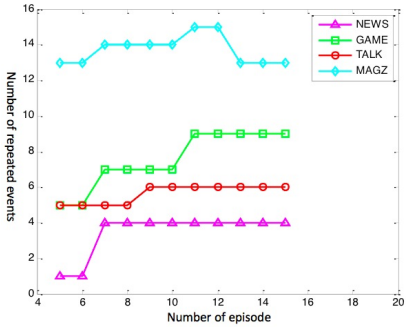


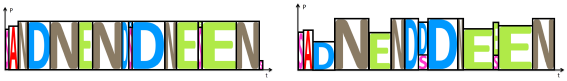
Fig. 8: Number of events determined for the four types of programs

Structural element	Symbol	SD	DT	MI	CT	MA	DP	SRS	SMS	AR
Separator NEWS	S			yes						
Separator GAME	S		yes	yes						
Separator TALK	S									yes
Separator MAGZ	S									yes
Dialog	D							yes		
Anchor's monologue	A	long					yes			
Music/song	M								music	
Commercials	C	short		yes					silence	
Outline NEWS	T	short							music&speech	
Full screen text	E				yes	low				

Table 2: Structural elements determination using a broad scope of detectors, where SD: Shot duration, DT: Dissolve transition, MI: Monochrome image, CT: Centralized text, MA: Motion activity, DP: Dominant person, SRS: Shot-reverse-shot, SMS: Speech/music/silence, AR: Audio recurrence

(a) MSA grammar (b) UNR grammar 1 (c) UNR grammar 2

Fig. 9: Structural models of TALK. Structural elements: separator (S), commercials (C), musical performance (M) and undefined (N).



(a) MSA grammar (b) UNR grammar

Fig. 10: Structural models of GAME. Structural elements are: separator (S), anchor (A), dialog (D), full text (E) and undefined (N).

6.3 Experiments on grammar inference

Having fixed the size of the inference set and the threshold of the density function, we now analyze the resulting structural models as obtained with multiple

(a) MSA grammar (b) UNR grammar 1 (c) UNR grammar 2

Fig. 11: Structural models of MAGZ. Structural elements: separator (S), anchor (A), commercials (C), dialog (D), full text (E), musical performance (M) and undefined (N).

(a) MSA grammar (b) UNR grammar 1 (c) UNR grammar 2

Fig. 12: Structural models of NEWS. Structural elements are: separator (S), outline (T), dialog (D) and undefined (N).

sequence alignment (MSA) and with uniform resampling (UNR) to compare their effectiveness on program structure understanding and on segmentation of new episodes.

Table 2 reports the generic rules that were used to map repeated events to structural elements before grammatical inference. Specifically, if an event detector contributes to identifying a specific structural element, it is marked as "yes" or with its corresponding detected results. For example, a structural element corresponding to a sequence of monochrome images is a separator for NEWS, while a long duration shot containing the dominant person is deemed as an anchor's monologue.

6.3.1 Qualitative analysis

We qualitatively analyze the models obtained by the two grammatical inference techniques. In particular, we focus on the interpretation of the grammars and thoroughly discuss the differences induced by the two techniques.

Examples of the structural models that were obtained are represented in Figures 9, 10, 11 and 12 for TALK, GAME, MAGZ and NEWS respectively. As cross-validation is used in practice, we randomly chose one of the structural grammars obtained through the different folds. However, we did not observe significant differences between different folds. For TALK, three structural models are illustrated in Figure 9, where three semantically interpretable structural elements are identified, i.e., separator (S), commercials (C) and musical performance (M). Figure 9(a) shows a structural model obtained with multiple sequence alignment, while Figure 9(b) and 9(c) are the two models obtained with uniform resampling. All three structural grammars describe a program with three main chapters bounded by separators (S) and commercials (C). The clustering stage in the uniform resampling strategy enables to identify two distinct structural models, depending on whether the episode ends with a musical performance (M) or not. For the two structural grammars obtained

with uniform resampling (Figures 9(b) and 9(c)), the evident difference is the presence of musical performance segment, based on which the episodes are clustered in two different groups, hence resulting in two different structural models. In other words, all episodes belonging to the structural model in Figure 9(b) are supposed to have the musical performance segment. Note that the musical performance is not totally lost in the structural model obtained with multiple sequence alignment but appears with a frequency less than that of the other elements in the model. In fact, the MSA structural model can be seen as a superposition of the two models identified with uniform resampling. Concretely, multiple sequence alignment relies on a dynamic alignment of the symbolic sequences, which leads to a relative high presence probability for each element. On the contrary, uniform resampling does not allow warping between episodes (apart from the duration normalization) and compute the probability of each structural element in each time interval. In other words, one element could be found in more than one time intervals, which reduces its presence probability in each time interval and somehow expands the element duration in structural grammars. The element duration in the MSA grammar is an average over the structural elements aligned at a position. Hence, the elements in the UNR grammars usually have longer duration and lower presence probability than the ones inferred by MSA.

The same phenomena could also be observed on the three other types of programs. Figure 10 shows the structural models for GAME respectively inferred by MSA and UNR. GAME has just one structure, hence, for multiple sequence alignment as for uniform resampling, there is just one structural model inferred. Reading the two models in Figure 10(a) and 10(b), the program structure for GAME is: Starting with anchorperson's opening (A), the program features an alternation of interviews (D) and of full text scenes (E). Comparing the two structural models, the elements in the UNR structural model generally have longer duration and lower presence probability than the ones inferred by MSA, which can be evidently observed from all the full text segments, separators and the first segment of dialog.

Following the same general observations, MAGZ interestingly shows a more complex structure in comparison to other programs. The structural grammars are in Figure 11, where anchorperson's opening (A), music (M), dialog (D), full screen text (D) and commercials (C) are determined as the elements structural of the grammar's vocabulary. MAGZ is divided into many chapters by separators, and content of each chapter varies a lot. The MAGZ models tell us that the proposed grammar inference methods work for complex structures, UNR being better at providing more details.

Finally, for NEWS, the three structural models in Figure 12 correspond to the coarse-grain structure of a classical news program: The program is introduced by a separator (S) and starts with the headlines (T); The following non interpreted element (N) correspond to the alternation of anchor's announcements and reports, possibly including an interview (D). The clustering stage in the uniform resampling strategy enables to identify two distinct grammars, depending on whether an interview is included at the end of the program

or not. However, the inference techniques used did not allow identifying the structure of the program at a finer grain, e.g., to discover the alternation of anchor's announcements and reports. This limitation, that didn't appear in GAME programs, is mostly due to the fact that the number of reports varies across episodes. While grammatical inference techniques that can cope with such features exist, we leave their use to future work, the main point in this paper being to show that structural element discovery and grammatical inference can be combined to infer structural models for further processing of new episodes. Lacking of the alternation of anchor's announcements and reports in NEWS shows a limitation of the proposed grammar inference approach, which makes the structural grammar of NEWS not practically usable for segmentation tasks.

6.3.2 Use-case: segmentation of new episodes

Quantitatively evaluating the quality of the structural models can hardly be done in a direct manner as no reference model exists, nor do we have a distance between models. We thus rely on a use-case scenario to verify the effectiveness of the models to segment new episodes. The scenario is the following: Given a number of episodes, we want to infer a structural model from a small number of episodes and segment the remaining episodes according to the model. By segmenting, we mean finding the structural elements that are present in an episode and determining their respective start and end times by mapping the structural model to the data, thus effectively providing a dense structure or, equivalently, a dense segmentation, for all episodes within the collection. The design of the use-case scenario considers the practical use of structural models, in particular in the workflow of broadcast archivists. For instance, at INA, the segmentation task mainly depends on manual operations that are errorprone, time taking and of little interest for librarians. The inferred structural model is thus welcome to provide a structural reference of the program for librarians to improve manual indexing tasks and focus on added-value operations. In this paper, the segmentation task solely relies on the time information of the structural elements provided by structural models, and no content-based interpretation of segmented episodes is involved. As mentioned previously, designing content-aware models is highly challenging because of the diversity across episodes: While the structure remains stable, the visual and audio content do not. The segmentation use-case also calls for models only aware of time as the goal is to predict which structural elements are present in new episodes and, most importantly, what their boundaries are. For the boundary issue, which is by far the most important for librarians to describe and index segments, time-based models suffice. For structural element prediction, content would certainly help but, again, designing a content-based model is far from trivial and remains out of the scope of this paper.

For comparison purposes, we constructed a (fairly naive) baseline model, whose segmentation results are compared with the structural models constructed via grammatical inference. The construction of the baseline model

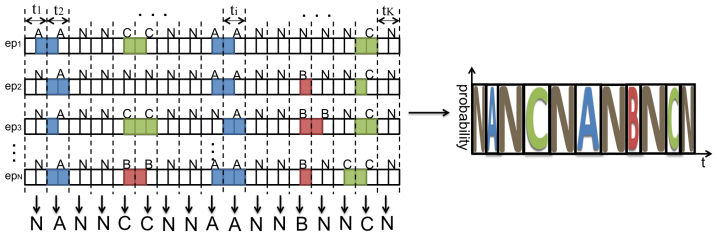


Fig. 13: Baseline model construction

is illustrated in Figure 13. After turning a collection of episodes into a set of fixed-length sequences based on uniform resampling, for each time interval, the element (including symbol 'N') appearing most frequently is voted as the element in that position. The boundary of the element is computed by counting the number of the successive time intervals having the same structural elements, and the presence probability is always deemed as 1. This model can be seen as a majority voting model where alignment is somewhat disregarded to keep only the most frequent structural element at each time interval.

In our experiments, we compare the segmentation results using standard performance measures such as precision (P), recall (R) and F measure (F). The metrics are computed on a time basis. A structural element is considered as being correctly predicted if it overlaps with a structural element of the same type in the ground-truth. In the case of multiple elements at one position, if one of them is matched, it is deemed as correctly predicted. Recall is measured as follows:

$$R = \frac{\sum_{j=1}^{m} \mid S_j^o \mid}{\sum_{i=1}^{n} \mid S_i^g \mid} \tag{4}$$

It computes the overlapping time of the elements correctly predicted divided by the total duration of structural elements in the ground-truth, where n is the number of annotated structural elements S^g in the ground-truth (not counting structural elements of type N) and m is the number of overlapping segments S^o between predicted structure and ground-truth. Precision is defined in a similar way, dividing by the total duration of the structural elements in the predicted structure, i.e.,

$$P = \frac{\sum_{j=1}^{m} \mid S_j^o \mid}{\sum_{i=1}^{w} \mid S_i^p \mid} \tag{5}$$

where w refers to the number of predicted structural elements S^p in the predicted structure (not counting structural elements of type N). Recall and precision measures are averaged across episodes. In the case of multiple models, the structural model exhibiting the best F measure is chosen as the final results .

Results for the four recurrent programs are reported in Table 3. NEWS has the best score in terms of precision. However, this result owes to the very

stable structure of news programs and to the simplicity of the inferred structure. Comparing with NEWS, the other programs exhibit lower performance, mostly because of the higher number of determined structural elements in the structural grammar. Especially, the structural elements having very short duration are easier to be missed than longer ones, such as the elements in MAGZ. The duration of structural elements in GAME is on average longer than the one for TALK and MAGZ, which results in a slight better performance for GAME. Furthermore, in the case of TALK and MAGZ, for all three inference methods, recall is rather high whereas precision is relatively low. This reveals that the duration of predicted elements is generally longer than the duration of the structural elements in the reference annotations. The contrary holds in the case of GAME and NEWS.

Comparing the three inference methods, one can see that generally the two proposed methods perform better than the baseline method, i.e., the proposed methods is higher from 0.03 to 0.22 than the baseline model in term of F measure. These results can be explained by the nature of grammatical inference techniques. First, both multiple sequence alignment and uniform resampling methods consider at a given position all the elements across episodes, where multiple sequence alignment relies on a dynamic alignment and uniform resampling adopts all elements at the each time interval. While the baseline model just counts the most frequent element at a given position, which means that the elements from some episodes are abandoned, thus providing incomplete boundary information for the model. Second, for the two proposed inference methods, multiple types of elements may be found at one position with their presence probabilities, which augment the chance of a structural element being correctly predicted. However, some cases may have similar Recall or Precision between proposed methods and the baseline method, e.g., for MAGZ the F measure of UNR model is just greater 0.03 than the baseline model. We can still say that UNR model preforms better than the baseline model, because the Recall of the UNR model is far greater (i.e., 0.15) than the baseline model. Recall computes the overlapping time of the elements correctly predicted divided by the total duration of structural elements in the ground-truth, which is more important for the constructed model in practical use, e.g., segmentation tasks. Comparing the results given by the two proposed inference methods, one can see that in case of simple structures, i.e., NEWS and TALK, uniform resampling is more precise. These results can be explained by the fact that the UNR method is capable of identifying multiple structures for a program, which highly raises the prediction precision, as the tested episodes are supposed to be more targeted by a certain model. However, in the case of more complex structures, i.e., GAME and MAGZ, multiple sequence alignment performs slightly better than uniform resampling. These results mainly owes to the different strategies used for the determination of the structural element boundaries in the two inference methods. The duration of structural elements in the UNR models is somehow extended by counting the length of time intervals, while the boundary of structural elements in the MSA models is the average value

	MSA			UNR			Baseline		
Dataset	P	R	F	P	R	F	P	R	F
NEWS	0.69	0.54	0.61	0.82	0.55	0.66	0.55	0.37	0.44
TALK	0.53	0.54	0.53	0.48	0.86	0.61	0.46	0.64	0.53
GAME	0.69	0.50	0.58	0.56	0.53	0.54	0.53	0.49	0.51
MAGZ	0.42	0.64	0.50	0.33	0.73	0.45	0.34	0.58	0.42

Table 3: Comparison of segmentation performance

of the aligned elements across episodes. The extended boundary in the UNR models leads to a lower prediction performance than for MSA models.

Although evident differences can be noticed between the structural grammars obtained by multiple sequence alignment and uniform resampling, they factually reflect the structure that one would expect demonstrated by the qualitative analysis. Furthermore, the quantitative evaluation conducted in the way of segmentation use-case shows that the feasibility of the proposed grammatical inference methods for segmentation tasks in practical use, e.g., to assist librarians for facilitating their indexing tasks of TV programs.

7 Discussion

The paper has demonstrated that grammatical inference is feasible for recurrent program structure modeling with minimal prior knowledge through symbolic representation, at the same time providing maximal semantic interpretation of program structures. This key result opens the door to a number of variations along the very same idea with the goal of improving the quality of the approach and the level of details that can be accounted for. We discuss hereunder several research directions that might be considered: improving the detection of general events; using more expressive grammatical inference techniques; automatically building a content-based model of the program.

7.1 Small object mining

The general event detectors adopted in the paper can be successfully used to discover the structural elements for various types of programs. However, there may exist a better choice of event detectors, e.g., having less detectors but more structural elements that are discovered, or adding various types of detectors to enrich the set of structural elements considered. In particular, we focused on generic structural elements and thus limited ourselves to global-scaled audiovisual events—i.e., the general events that are usually extracted from a whole video frame (e.g., monochrome image) or the most important part of a video frame (e.g., person clustering). However, as pointed out in the results discussed above, almost half of the program's structure (measured in time) remains undetermined. For example, in the talk show, the flash question sections conducted by different people with various questions can hardly

Fig. 14: Instances for small object mining

Program	Raw structure	Fine structure
GAME	61.5%	70.3%
MAGZ	46.3%	59.6%
TALK	23.2%	42.7%

Table 4: Time occupied of determined segments for GAME, MAGZ, TALK

be detected. The lack of such important structural elements usually leads to an incomplete program structure, and limited understanding of the program. Therefore, it is crucial to improve the completeness of a program structure, i.e., recovering the structure with all potential structural elements.

To improve completeness, we propose to consider small visual objects present in a recurrent TV program in addition to global-scaled elements. Some small logos (usually less than 20% of the image) often appear at specific parts of a program. For example, in the talk show program, some small objects such as the same screen usually appear in flash question parts across episodes. We conducted preliminary experiments where small object recurrence is considered to improve an initial structural grammar obtained from global-scale elements. Given the initial structural grammar, we mined the presence of small objects using the approach in [18] and analyzed their temporal distribution for the video segments that are labeled as unknown structural elements in the initial grammar. The small objects discovered are grouped into clusters, where frames in one cluster are supposed to correspond to the same object/event. As for global-scaled events, we analyze the temporal distribution of clusters, enabling the definition of new structural elements from recurrent small objects and thus a more complete grammar.

As an illustration, Figure 14 shows five instances of small objects extracted from the programs, where each row represents the images from a cluster containing the same small objects. The corresponding datasets are indicated on the left of each row. Obviously, we can tell that the screen and the lighted circles are the same objects shared by the images in the first row. Practically, these frames correspond to a flash question scene at the end of different episodes. Similar observation can also be found in others examples: the frames

in the second row, containing television logos and black regions, refer to a flash news scene; the third row with the same weather cliparts and the striped backdrops obviously corresponds to the weather forecast. The clusters in the last two rows seem less obvious to interpret. However, very similar shapes can be found for each of them: in the forth row, the flowers and leaves are the clue for the frames belonging to a plant representation scene; while the yellow panes in the last row are deemed as the clue for a scene of game winner for the game show. These scenes can hardly be detected using the global-scaled audiovisual detectors. One may argue that global-scaled detectors can find such elements, such as the examples of the first and the third rows. This is indeed the case but to the price of massive prior knowledge and detailed characteristics on the frames. By leveraging small object mining, no prior knowledge is required for detection. However, identifying structural elements from small-object events in a lightly supervised way remains a challenge.

We also evaluated the effectiveness of element discovery using small object mining by computing an error rate on the new elements. The element detection error rate measures the fraction of small object clusters that are not correctly attributed to structural elements, where the correspondence between small object clusters and structural elements is manually evaluated. Taking an average over the three programs, 18.1 % of the identified clusters are not recognized as valid structural elements. This low error rate highly raises the confidence of the newly improved structural model. In addition, we compare the completeness of the previously obtained models (i.e., raw models) and the models improved by small object mining (i.e., fine models). We compute the percentage of the total duration with determined elements in a program. For each program, the result is the average value of all the existing models. Specifically, we observe that, in Table 4 compared to the raw model, the completeness of the fine model is augmented by 8.8 %, 13.3 %, and 19.5 %, for GAME, MAGZ and TALK, respectively. Obviously, the percentage of determined structural elements for fine models is highly improved. These results show that small object mining is indeed an effective way to augment the structural model completeness of recurrent programs.

7.2 Regular expression

With the goal of demonstrating that the construction of structural models for recurrent TV programs can be cast as a grammatical inference with no supervision and very limited domain knowledge, this work is limited to simple grammatical inference techniques, which does not allow for complex structures. This is evidenced in the case of NEWS, where the alternation of anchor's announcement and news reports that constitute the bulk of a news program cannot be expressed because of the varying number of such alternations across episodes. Obviously, more expressive inference techniques, like identifying regular pattern during the processing of grammatical inference [26], or enhancing the grammar generalization after an initial grammars using regular expres-

sions [12]. In particular, regular expressions [3] appear as an appealing choice for a straightforward extension of this work. A regular expression is used to specify certain regular patterns by leveraging logical operators. Making use of such regular expressions can enhance the structure expression capacity of grammars, hence improving the structure granularity of certain programs. For instance, in NEWS, the alternation of anchor's announcement and news reports can be easily expressed by regular expressions as $(AB)^*$, where A refers to anchor's announcement and B refers to news reports. While the initial results reported in this paper hints at the feasibility of using such inference techniques, combinatorial issues are to be expected.

7.3 Content-based segmentation

The quality of structural models refers not only to the descriptiveness of the models obtained, but also to their ability for segmenting new episodes, which is the most important application of having a structural model for programs. The model derived from the structural grammar was still limited to time considerations to find the boundaries of the structural elements for the new episodes. Not using a content-based model for segmentation task is justified by the fact that the structural elements might have little, if any, commonalities at the content level. Yet, simple rules could be used to design a content-based model of some structural elements, e.g., separators or dialogues, based on the set of broad scope detectors. In particular, after directly scaling the structural model to the episodes to be segmented, as stated in Section 6, time-based boundaries could be predicted. The proposed segmentation strategy did not account for the content of new episodes. We now suggest to apply the set of broad scope events detectors on the predicted episodes to simply discover their contents, estimating content-based boundaries based on the event types as well as domain knowledge. Specifically, after the time-based segmentation, we could adjust the boundaries of structural elements by considering the content around the boundaries. On the one hand, by jointly considering the time-based boundaries and the content-based boundaries, the more accurate boundaries of structural elements could be found. On the other hand, the content discovery of the episodes also provides a verification of the prediction. This perspective however raises scientific challenges in content-based modeling, for instance to mix structural elements having a content-based model with structural elements having no content model, which turns the problem to new research directions, e.g., a mixture model generalized by a hidden Markov model.

8 Conclusion

We have proposed an unsupervised approach addressing the problem of structure modeling for recurrent TV programs. Leveraging grammatical inference techniques and symbolic representation, we have shown that relevant structures can be discovered with only minimal domain knowledge and that a model

can be constructed to segment new episodes in practical use. As the proposed approach can be applied on a large variety of programs, it might be practically used to assist librarians to segment episodes of recurrent programs, for the purpose of saving time of manual operations. Apart from structure modeling, the discovery of structural elements and the corresponding symbolic representation of episodes that we propose can be used as an input to a number of symbolic data mining algorithms to extract knowledge from a collection of episodes. Furthermore, the proposed grammatical inference approach can be utilized in other domains to build structural/behavior models, e.g., to predict upcoming events or to detect anomalies, where provided domain knowledge is given to turn repeated events into structural elements. For instance, video surveillance appears like a potential application domain, with a regular structure throughout different days that could be identified with grammatical inference. The resulting structural model could then be used to segment new days of footage so as to facilitate browsing or detection of abnormal events. Note that this is also a typical case where hierarchical approaches to grammatical inference and the possibility of identifying multiple structures (e.g., week days vs. Sat. vs. Sun.) are required.

References

1. Abduraman, A.E., Berrani, S.A., Merialdo, B.: An unsupervised approach for recurrent TV program structuring. In: EuroITV'11, pp. 123–126 (2011)
2. Abduraman, A.E., Berrani, S.A., Merialdo, B.: Audio/visual recurrences and decision trees for unsupervised TV program structuring. In: VISAPP'13, pp. 701–708 (2013)
3. Alfred, V.: Algorithms for finding patterns in strings. Algorithms and Complexity **1**, 255 (2014)
4. Ancona, N., Cicirelli, C., Branca, A., Distante, A.: Goal detection in football by using support vector machines for classification. In: IJCNN'01, vol. 1, pp. 611–616 (2001)
5. Ben, M., Gravier, G.: Unsupervised mining of audiovisually consistent segments in videos with application to structure analysis. In: ICME'11, pp. 1–6 (2011)
6. Botev, Z., Grotowski, J., Kroese, D.: Kernel density estimation via diffusion. Ann. Stat. **38**(5), 2916–2957 (2010)
7. Chang, Y.F., Lin, P., Cheng, S.H., Chan, K.H., Zeng, Y.C., Liao, C.W., Chang, W.T., Wang, Y.C., Tsao, Y.: Robust anchorperson detection based on audio streams using a hybrid i-vector and DNN system. In: APSIPA ASC'14, pp. 1–4. IEEE (2014)
8. Chua, T.S., Chang, S.F., Chaisorn, L., Hsu, W.: Story boundary detection in large broadcast news video archives: techniques, experience and trends. In: MM'04, pp. 656–659 (2004)
9. Crooks, G.E., Hon, G., Chandonia, J.M., Brenner, S.E.: Weblogo: A sequence logo generator. Genome Res. **14**(6), 1188–1190 (2004)
10. Dumont, E., Quénot, G.: Automatic story segmentation for tv news video using multiple modalities. International Journal of Digital Multimedia Broadcasting (2012)
11. Gupta, V., Kenny, P., Ouellet, P., Stafylakis, T.: I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pp. 6334–6338. IEEE (2014)
12. Hopcroft, J.E.: Introduction to automata theory, languages, and computation. Pearson Education India (1979)
13. Jacobs, A.: Using self-similarity matrices for structure mining on news video. In: Lect. Notes Artif Int., pp. 87–94. Springer (2006)

14. Jayagopi, D.B., Ba, S., Odobez, J.M., Gatica-Perez, D.: Predicting two facets of social verticality in meetings from five-minute time slices and nonverbal cues. In: ICMI'08, pp. 45–52 (2008)
15. Ji, P., Cao, L., Zhang, X., Zhang, L., Wu, W.: News videos anchor person detection by shot clustering. Neurocomputing **123**, 86–99 (2014)
16. Kijak, E., Gravier, G., Oisel, L., Gros, P.: Audiovisual integration for tennis broadcast structuring. Multimed. Tools Appl. **30**(3), 289–311 (2006)
17. Lee, H., Yu, J., Im, Y., Gil, J.M., Park, D.: A unified scheme of shot boundary detection and anchor shot detection in news video story parsing. Multimedia Tools and Applications **51**(3), 1127–1145 (2011)
18. Letessier, P., Buisson, O., Joly, A.: Scalable mining of small visual objects. In: MM'12, pp. 599–608 (2012)
19. Li, H., Tang, J., Wu, S., Zhang, Y., Lin, S.: Automatic detection and analysis of player action in moving background sports video sequences. IEEE Trans. Circuits Syst. Video Technol. **20**(3), 351–364 (2010)
20. Mocanu, B., Tapu, R., Zaharia, T.: Automatic segmentation of tv news into stories using visual and temporal information. In: International Conference on Advanced Concepts for Intelligent Vision Systems, pp. 648–660. Springer (2016)
21. Qu, B., Vallet, F., Carrive, J., Gravier, G.: Content-based inference of hierarchical structural grammar for recurrent TV programs using multiple sequence alignment. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2014)
22. Qu, B., Vallet, F., Carrive, J., Gravier, G.: Content-based discovery of multiple structures from episodes of recurrent TV programs based on grammatical inference. In: International Conference on Multimedia Modelling, pp. 140–154 (2015)
23. Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., Trancoso, I.: Temporal video segmentation to scenes using high-level audiovisual features. IEEE Transactions on Circuits and Systems for Video Technology **21**(8), 1163–1177 (2011)
24. Stuhlsatz, A., Meyer, C., Eyben, F., Zielke, T., Meier, G., Schuller, B.: Deep neural networks for acoustic emotion recognition: raising the benchmarks. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 5688–5691. IEEE (2011)
25. Thompson, J.D., Higgins, D.G., Gibson, T.J.: Clustal W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids research **22**(22), 4673–4680 (1994)
26. Thompson, K.: Programming techniques: Regular expression search algorithm. Communications of the ACM **11**(6), 419–422 (1968)
27. Xie, L., Xu, P., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with domain knowledge and hidden Markov models. Pattern Recogn Lett. **25**(7), 767–775 (2004)
28. Yang, X.F., Tian, Q., Xue, P.: Efficient short video repeat identification with application to news video structure analysis. IEEE Trans. on Multimedia **9**(3), 600–609 (2007)
29. Zhang, D.Q., Lin, C.Y., Chang, S.F., Smith, J.R.: Semantic video clustering across sources using bipartite spectral clustering. In: ICME'04, vol. 1, pp. 117–120 (2004)
30. Zhang, J., Qiu, J., Wang, X., Wu, L.: Representation of the player action in sport videos. In: APSIPA ASC'13, pp. 1–4. IEEE (2013)
31. Zhu, S., Liu, Y.: Video scene segmentation and semantic representation using a novel scheme. Multimedia Tools and Applications **42**(2), 183–205 (2009)