Video Summarization using Causality Graphs

Shay Sheinfeld[1], **Yotam Gingold**[2], **Ariel Shamir**[1]
[1]The Interdisciplinary Center, Israel, [2]George Mason University

Abstract

Video summarization is useful for many applications such as content skimming and searching. Automatic video summarization is extremely challenging as it often depends on semantic tasks such as determining meaning, causal relationships, and importance of the displayed video events. We present a reliable, crowdsourced solution to video summarization based on human computation that addresses one of the main semantic challenges in story understanding: recognizing cause and effect. Our approach first automatically divides the video into simple shots as atomic elements. Using these elements we construct a context-tree (Verroios and Bernstein 2014) to gain global understanding, and introduce a crowdsource algorithm that explicitly builds a *causality graph* representing the causality relations between events in the video. We use both importance and causality to create a summary of the original video. Our evaluation shows that information from the causality graph creates better summarizations of the original video.

Introduction

Video summarization, in which a short version of a long video is created, is useful for many applications such as content skimming and searching in video libraries. Although fully automatic summarization algorithms are highly desired, they require solutions to many difficult tasks: detecting scene changes, determining who or what appears in each scene, recognizing words and actions, and above all assigning meaning, causal relationships, and importance to the displayed events. Computer vision algorithms can provide some solutions to the former tasks, but the latter task is semantic in nature and is therefore, the most challenging for a fully automatic algorithm (machine computation).

We present a hybrid automatic (machine) and human computation approach to the problem. Our crowdsourced part addresses one of the main semantic challenges in story understanding: recognizing cause and effect. Our approach first automatically divides the video into simple shots as building blocks. These shots are converted to textual descriptions using crowdsourcing while maintaining consistency in naming characters. We utilize the recent context trees approach of Verroios and Bernstein (2014), to build a context-tree of the shots. This approach was designed to create global understanding in tasks where each contributor only has access to local views. However, the context-tree does not provide knowledge about cause and effect. Using importance ranking from the context tree alone for summarization can exclude events crucial to the video understanding. Using simple crowd sourcing tasks we hierarchically build an explicit *causality graph* between story units of the context tree (see Figure 1). This causality graph is then used for better summarizations of the original video. Our evaluation shows that compared to a baseline of using only the context-tree, the information from the causality graph creates better summarizations of the original video.

Previous work

Extensive work has been done in the field of video abstraction and summarization. Truong and Venkatesh (2007) conducted an extensive survey of video abstraction techniques which produce either a sequence of representative still images or a shorter video.

Without humans in the loop, automatic approaches are limited to heuristics for understanding the video contents. For instance, in a restricted domain such as classroom lectures the pitch of the speaker (He et al. 1999) or transcript text (Shin et al. 2015) can be used. Crowdsourcing had been used to create text summaries of video segments (Pavel, Hartmann, and Agrawala 2014) as well as static representations of how-to videos (Kim et al. 2014; Weir et al. 2015). Bernstein et al. (Bernstein et al. 2011) introduced an approach to find high quality still images in short videos with parallel crowdsourcing. In contrast, our aim is to use crowdsouring to shorten a fiction videos. Wu et al (2011) described a technique which summarizes a movie by asking each worker to watch the entire movie. Our approach relies on parallel crowdsourcing as well as introduces the novel causality graph.

Our causality graphs are closely related to plot graphs used in the story generation literature. Plot graphs were first described by Kelso et al. (1993) as a structure for modeling events cause in interactive storytelling (Weyhrauch 1997; Young 1999).

Algorithm

In all parts of our algorithm each worker only sees a local part of the input video. The algorithm distributes and combines all these local views into a global understanding of the video's

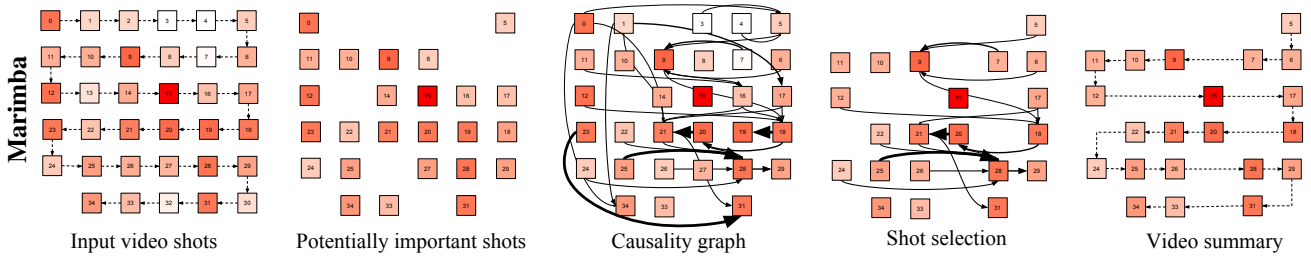

Figure 1: The entire Marimba video is automatically decomposed into shots. The importance of each shot is computed by the crowd via the context tree algorithm (Verroios and Bernstein 2014) (more important shots are darker). Potentially important shots are used in our crowd-based causality graph construction algorithm that computes causal relationships between the shots (thicker edges in the causality graph mean higher weight). Finally, important shots *and their causal dependencies* are selected to create a video summary.

plot, including the cause and effect graph of events in the video, crucial for creating high quality summarization.

Our algorithm is composed of the following steps. Some of these steps (namely, 1, 5, 7 and 8) are automatic in nature, and others (steps 2, 3, 4, 6) use *human computation*:

1. Shot detection
2. *Character naming*
3. *Creation of textual descriptions for shots*
4. *Importance scoring using the context tree*
5. Filtering of shots
6. *Construction of the causality graph*
7. Shot selection
8. Video summary creation

Examples of all human computation tasks are provided in the supplemental materials.

Shot-detection. The first step of our method creates the basic building blocks that cannot be separated during the various stages of the algorithm. We assume that the input video is built from multiple shots that tell a story. Therefore, we chose to use single shots as the building blocks of our algorithm. We further constrain the shots to have a minimum length of 4 seconds. We use a region-base histogram comparison to detect shot boundaries. Every two successive frames are split into four equal regions, and the HSV histogram of every two corresponding regions are compared using correlation. If the average correlation is more than a threshold and the length of the current segment is more than 4 seconds, a shot boundary is marked and the video is split.

Description & Naming. Each extracted shot is translated into a textual representation. This text will be used in the context tree algorithm as well as the causality graph creation. We use human computation to translate the shots into text. As this translation will be performed by different workers, the names of the characters must be unified such that each character will be addressed in the same manner. We either use well know characters (e.g. Popeye) or label each character so that important characters' pictures and names are provided to the worker in the description of the text translation task.

Context-Tree. The basic building blocks of the context tree algorithm are small parts of the original input. These parts are mutually exclusive, their union is the original input, and they constitute the leaves of the tree. For example, Verroios and Bernstein (2014) used their context tree algorithm to summarize a whole book by splitting the book into paragraphs. Our approach divides the video into smaller building blocks, each containing a single shot.

We use the context tree algorithm to obtain the importance of each shot. Our context tree construction algorithm follows the original two phase approach:

1. In the up-phase (from the leaves to the root), workers are shown b successive parts (e.g. shots) of the input video and asked to write a text summary of these shots. At the end of this phase, the root contains a context-less text summary of the whole input video.
2. In the down-phase (from the root back to the leaves), workers provide context to the text summary. They are asked to rate the importance of each inner node for understanding the parent node summary. At the end of this phase, all leaves have a normalized importance score.

We use a branching factor of $b = 3$, and a replication factor of $r = 4$, which means that every corresponding task is replicated four times, to be answered by four different workers.

Scoring & Filtering. The output of the previous context tree step is a numeric score for every node, including leaf nodes, which represent individual shots. Hence, the importance score of each leaf node provides a way to rank the shots. However, we have found that the relative value of the important score, i.e. the value of a shot relative to its neighboring shots, can also be indicative of the shot's importance.

Our filtering scheme is based on two importance threshold parameters. T_1 is the threshold for the top rated shots. Any shot whose importance score is greater than T_1 will be considered as potentially important. $T_2 = \frac{1}{2}T_1$ is the threshold for local maxima shots. Any local maxima shot whose importance score is greater than T_2 will also be marked as potentially important. In the next stage of creating the causality graph use only potentially important shots (see Figure 1).

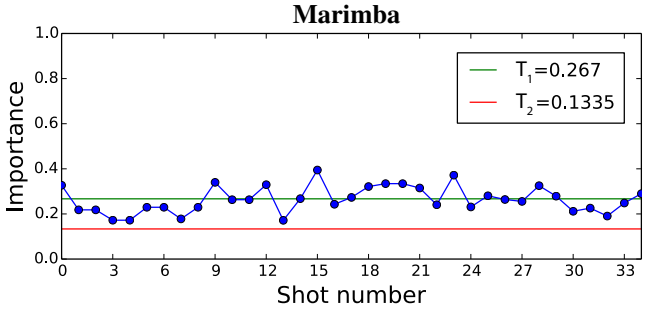

Figure 2: The importance score of every shot in the Marimba movie as computed by the context tree algorithm. We use two thresholds: every shot above threshold T_1 is marked as important, as well as local maxima shots above threshold T_2 (for example, shot number 4). All other shots are pruned before the construction of the causality graph.

Causality Graph. A causality graph is a weighted directed graph. Its nodes are events (shots) from the video. The directed edges represent a causal relationship between two events in the video: the source event is a direct cause of the target event. The weight of each edge represents the strength of the relationship, measured by the evidence found for this relationship. Although the previous stages provides a way to determine important events in the movie, there are no connections between these events and the plot structure is missing. Our causality graph provides a way to connect these events in a meaningful manner.

A naive crowdsourcing algorithm for constructing the causality graph will ask, for every pair of shots, if one causes the other. This will yield $O(n^2)$ queries, which is resource consuming and may also contain incorrect edges, since a worker cannot correctly identify a causal relationship between two events without context. To solve this, we build a sequence of causality graphs whose source nodes are nodes from level i of the context tree and target nodes are the leaves (i.e. individual shots). The construction of the causality graph for level i depends on the causality graph of the previous (higher) level. The final causality graph will include only leaves. This reduces computation time to $O(n \log n)$.

We define a *query node* as a text leaf node describing one shot in the movie. Note that the set of query nodes is fixed, and does not change when passing from one level of the context tree to the next. We define a *causal node* as any node in the context tree that could potentially be a cause of a query node.

The causality graph for level i, denoted CG_i, is a causality graph whose nodes consists of all the query nodes and the causal nodes from level i of the pruned context tree. The final causality graph will consist of only query nodes (i.e. shots) and edges between them, but along the way we recursively construct several causality graphs.

The base case of the algorithm constructs CG_1, the causality graph for the level below the root of the tree. Using the crowd, we find, for each query node, all the possible causal nodes in the first level of the tree. For a given query node, we consider only causal nodes whose subtree contains shots before the query node.

Now, the causality graph CG_i of level i is a graph that consists of edges from causal nodes at level i to query nodes. To construct graph CG_{i+1}, the potential causes (causal nodes) for query nodes are calculated as follows: Let v be a node from level $i+1$ of the pruned context tree, $p(v)$ its parent, and q a query node. We say that v is a potential cause for q if $(p(v), q)$ is an edge in CG_i. While constructing CG_{i+1}, we use the crowd to determine whether each potential causal node v is actually a direct cause of q or not.

We ask each query $d = 5$ times. The count $c(v, q)$ of an edge (v, q) is the number of workers who indicate that v is a direct cause of q. If the count is 0, we do not create an edge at all. The weights of the edges in the final causality graph CG are defined recursively by adding the counts of their parents (see Figure 1).

Shot Selection. Given a target video summary duration (or percentage of the input movie duration), our algorithm first sorts all the un-pruned context tree nodes according to their importance. We then loop until the target duration is exceeded, and choose the most important unselected node q for inclusion in the summary. We also include all shots v with causality edges that point to q if the causality edge weight is above a threshold $w(v, q) > T_3$. In our experiments, we set T_3 to be half of the largest edge weight in the graph. We continue this process until we exceed the desired movie duration.

Results

We have implemented our algorithm and created video summaries for several movies. We used Amazon Mechanical Turk paid crowdsourcing platform. Examples for all tasks (HITs) can be seen in the supplemental materials, including payment per task. For all videos, we set the target length at 50%. Our video summaries can be seen in the supplemental materials, including more drastic summarization of 25%.

Marimba[1] is a 7.1 minute long live-action movie with a moral message. This movie is challenging because seemingly unimportant events near the beginning of the movie become important near the end, and the moral message requires global context. **Cupido - Love is blind**[2] is a 7.4 minute long animated movie about Cupid and his quest to make two people fall in love. This movie offers several challenges as it has a long-term causality relationship, contains shots which are difficult to translate into text with only local information, and includes various artistic rendering styles. **Innocence**[3] is a 5 minute long live-action movie about a theft. This movie is challenging because the cause-and-effect of the theft involves characters in different locations, it includes a change of heart that requires global context to understand, and the return of the theft is implied but not actually shown.

The goal of our algorithm is to create a summary video that shortens an input video while still capturing the plot of the

[1] https://www.youtube.com/watch?v=PrXBiZD7a28
[2] https://www.youtube.com/watch?v=Pe0jFDPHkzo
[3] https://www.youtube.com/watch?v=B2V8lKgjc9I

input. To evaluate this, we first compared text summaries created by viewing our video summaries to summaries created by viewing the entire movie (ground truth). We showed the original movie and the two text summaries to raters and asked them to choose the better summary. Participants preferred the summaries written by the *entire movie* group 49% of the time (23, 31, and 19 votes for Marimba, Innocence, and Cupido, respectively) and summaries written by the *causality graph* group 51% of the time (27, 19, and 31 votes for Marimba, Innocence, and Cupido, respectively). These differences are not statistically significant, and we conclude that text summaries of the unabridged video and our video summaries are of similar quality.

We also compared our video summaries to video summaries created using only the context tree (baseline). Raters first watched the entire movie and were asked to write a text summary of the movie (this was intended to induce comprehension and, therefore, a thoughtful choice). Finally, raters watched the two video summaries and indicated which one they thought was better. Raters preferred the *causality graph* summaries 82% of the time (18, 16, and 15 votes for Marimba, Innocence, and Cupido, respectively) andthe *context tree* summaries 18% of the time (2, 4, and 5 votes for Marimba, Innocence, and Cupido, respectively). These differences are statistically significant ($p < 0.05$), showing a clear advantage for using causality relations in video summaries.

Conclusion and Future Work

We have presented a human computation algorithm for video summarization that takes semantics into account and preserves causality relations of events in the movie. Our algorithm takes advantage of the crowd by splitting the input into mutually exclusive shots and running tasks in parallel, so that each task is completed by a different worker with only local information. Our approach uses a context tree to provide global context for isolated and distributed crowd workers and builds a causality graph to capture causal relationships important for understanding. Video summaries created with our algorithm led viewers to similar understanding of the plot as viewers of the entire, unabridged video. Moreover, our video summaries were judged to be better than video summaries created with information from the context tree.

There are limitations to our method. For one, we work at the granularity of shots. If one shot is very long and is chosen for the summary, we do not cut it and it may take up too large a portion of the summary. In the future, we plan to develop methods for rating and possibly shortening individual shots. There are also still possibilities for confusion in the causality graph for very complicated plots or ones that are repetitive.

Other directions for the future are automate character identification, with either human or machine computation. We would also like to scale our algorithm to feature length films and adapt it to lecture videos. We also plan to explore additional applications of the semantic information generated by our human computation causality graph creation. This information could be used to create different types of summaries, such as static arrangements (tapestries, storyboards or comics). This information could also be used to create video remixes.

References

[2011] Bernstein, M. S.; Brandt, J.; Miller, R. C.; and Karger, D. R. 2011. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *User interface software and technology (UIST)*, 33–42. ACM.

[1999] He, L.; Sanocki, E.; Gupta, A.; and Grudin, J. 1999. Auto-summarization of audio-video presentations. In *Multimedia*, 489–498. ACM.

[1993] Kelso, M. T.; Weyhrauch, P.; and Bates, J. 1993. Dramatic presence. *PRESENCE: Teleoperators & Virtual Environments* 2(1):1–15.

[2014] Kim, J.; Nguyen, P. T.; Weir, S.; Guo, P. J.; Miller, R. C.; and Gajos, K. Z. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *SIGCHI Conference on Human Factors in Computing Systems*, 4017–4026. ACM.

[2014] Pavel, A.; Hartmann, B.; and Agrawala, M. 2014. Video digests: A browsable, skimmable format for informational lecture videos. In *User interface software and technology (UIST)*, 573–582. ACM.

[2015] Shin, H. V.; Berthouzoz, F.; Li, W.; and Durand, F. 2015. Visual transcripts: Lecture notes from blackboard-style lecture videos. *ACM Trans. Graph.* 34(6):240:1–240:10.

[2007] Truong, B. T., and Venkatesh, S. 2007. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.* 3(1).

[2014] Verroios, V., and Bernstein, M. S. 2014. Context trees: Crowdsourcing global understanding from local views. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

[2015] Weir, S.; Kim, J.; Gajos, K. Z.; and Miller, R. C. 2015. Learnersourcing subgoal labels for how-to videos. In *Computer Supported Cooperative Work & Social Computing (CSCW)*, 405–416. ACM.

[1997] Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, Carnegie Mellon University.

[2011] Wu, S.-Y.; Thawonmas, R.; and Chen, K.-T. 2011. Video summarization via crowdsourcing. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, 1531–1536. ACM.

[1999] Young, R. M. 1999. Notes on the use of plan structures in the creation of interactive plot. In *AAAI Fall Symposium on Narrative Intelligence*.