Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Event Video Mashup: From Hundreds of Videos to Minutes of Skeleton

Lianli Gao,[1] Peng Wang,[2] Jingkuan Song,[3] Zi Huang,[2] Jie Shao,[1] Heng Tao Shen[1]
[1]University of Electronic Science and Technology of China, Chengdu 611731, China.
[2]The University of Queensland, QLD 4072, Australia.
[3]Columbia University, NY 10027, USA. {lianli.gao, shaojie}@uestc.edu.cn, {p.wang6, huang}@itee.uq.edu.au,
jingkuan.song@gmail.com, shenhengtao@hotmail.com

Abstract

The explosive growth of video content on the Web has been revolutionizing the way people share, exchange and perceive information, such as events. While an individual video usually concerns a specific aspect of an event, the videos that are uploaded by different users at different locations and times can embody different emphasis and compensate each other in describing the event. Combining these videos from different sources together can unveil a more complete picture of the event. Simply concatenating videos together is an intuitive solution, but it may degrade user experience since it is time-consuming and tedious to view those highly redundant, noisy and disorganized content. Therefore, we develop a novel approach, termed *event video mashup* (EVM), to automatically generate a unified short video from a collection of Web videos to describe the storyline of an event. We propose a submodular based content selection model that embodies both importance and diversity to depict the event from comprehensive aspects in an efficient way. Importantly, the video content is organized temporally and semantically conforming to the event evolution. We evaluate our approach on a real-world YouTube event dataset collected by ourselves. The extensive experimental results demonstrate the effectiveness of the proposed framework.

Introduction

In recent years, we have witnessed the explosive growth of Web video content on the Internet due to the tremendous advance of video-sharing platforms, digital cameras, fast network and massive storage. Users are often overwhelmed by the unstructured videos and in danger of getting lost in the video data world. Therefore, it is increasingly important to automatically summarize a large collection of Web videos in an efficient yet comprehensive way.

Inspired by text and image summarization (Mason et al. 2016; Liu, Yu, and Deng 2015; Singla, Tschiatschek, and Krause 2016; Lin and Bilmes 2011), video summarization (Truong and Venkatesh 2007) has be proposed as an early attempt to solve this problem. Based on some designed criteria, video summarization typically selects a subset of frames and/or subshots from a long-running video to form a short summary. Traditional summarization methods usually rely on low-level cues to determine the importance of segments of a video (Ma et al. 2005; Ngo, Ma, and Zhang 2005), and formulate summarization as a reconstruction problem (Zhao and Xing 2014) where sparse coding is employed to remove the redundant video content from a single video. More recently, based on the assumption that Web images tend to capture subjects from canonical viewpoint, (Khosla et al. 2013; Kim, Sigal, and Xing 2014; Song et al. 2015) utilize Web image prior to help extract meaningful subsets of user videos. Meng *et al* (Meng et al. 2016) selects representative objects to form video summarization. Apart from the aforementioned unsupervised methods, the works in (Potapov et al. 2014; Gong et al. 2014; Zhang et al. 2016) formulate video summarization as a supervised content selection problem and utilize priori to train a system to determine the relevance of video content. However, the above methods are focusing on a single video summarization.

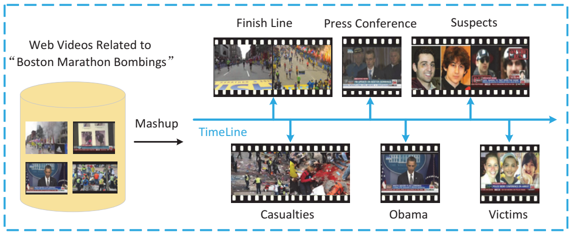

Figure 1: Overview of Event Video Mashup

There are some recent multiple-video summarization works (Chu, Song, and Jaimes 2015; Yeo, Han, and Han 2016; Hong et al. 2011; Tan, Tan, and Ngo 2010; Wang et al. 2012). However, video co-summarization methods (Chu, Song, and Jaimes 2015; Yeo, Han, and Han 2016) only focus on extracting visual co-occurrence across multiple videos. This summarization criterion may fail to capture some important information which are not shared by multiple videos. On the other hand, (Hong et al. 2011; Tan, Tan, and Ngo 2010; Wang et al. 2012) extract typical frames/shots and then assign them with semantic annotations, which is very similar to our method. But during the temporal alignment step, (Hong et al. 2011; Wang et al. 2012) simply assume that each tag is given a time stamp, and

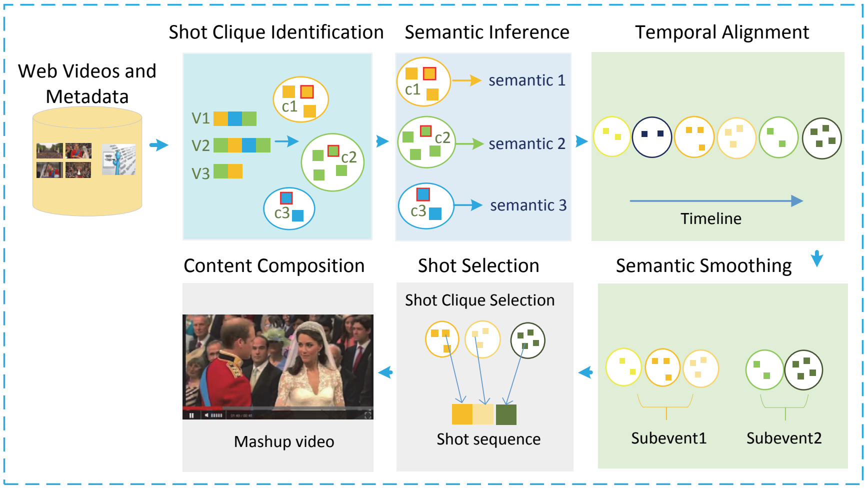

Figure 2: The flowchart of the proposed event video mashup approach

(Tan, Tan, and Ngo 2010) heavily relies on *Google Trend*. Therefore, these are not standalone multiple-videos summarization approach for practical applications.

To handle the aforementioned problems, in this paper we propose a novel framework named *event video mashup (EVM)* to automatically generate a short video from a collection of related videos to describe an event. Different from video summarization techniques which extract important frames/shots from a single video, video mashup identifies important frames/shots and temporally aligns them by simultaneously considering multiple videos from different sources with varying qualities, resolutions, lengths, etc. Figure 1 depicts the general idea of event video mashup, where a short video is constructed by the most important content selected from relevant videos along the time line.

Our contributions are summarized as follows: 1) as far as we know, this is the first standalone framework to automatically generate a temporal-aligned summarization from a collection of Web videos which are unstructured, unprofessional and may be edited arbitrarily. 2) To generate EVM, we first temporally align the shot cliques by utilizing the temporal information and a semantic smoothing strategy, and then devise an effective submodular based shot clique selection strategy to ensure the significance and coverage of the selected content; 3) To enable progress on this problem, we construct a new dataset with 1,600 videos describing 8 events covering different topics. Extensive empirical results on this dataset show that our proposed method significantly outperforms the state-of-the-art approaches.

Event Video Mashup

Event video mashup is a technique that mashes together many videos related to an event from different sources and then aggregates the most important content into a single new video, to allow users to understand the event efficiently and comprehensively. There are two main problems involved in event video mashup, including content selection and content composition. Specifically, content selection concerns how to select a subset of video content from different raw videos in order to cover representative and comprehensive information of the event in its evolving order. The objective of content composition is to use the selected content to compose a single mashup video in a way that the final video can be smoothly viewed by users.

To address the aforementioned problems, we propose a new framework that can automatically generate a short mashup video from a set of Web videos related to an event. Figure 2 gives an overview of our proposed mashup system. Given a collection of videos, firstly the shots of videos are grouped into shot cliques using a graph-based near-duplicate detection approach. Then semantics of those identified shot cliques are inferred to unveil their relevance to key semantic facets of the event. Next, the shot cliques are temporally aligned, followed by a semantic smoothing step to refine the alignment and discover key subevents of the event. Taking into consideration both importance and diversity factors, a shot clique selection method is performed. From each top ranked shot clique, a shot is chosen to represent the clique. All the above steps can be generalized as content selection. Finally, a sequence of selected shots in their previously determined order compose the final mashup video with transition smoothness being considered. The mashup video length can be adaptively adjusted by varying the number of top ranked shot cliques. Next, we describe these components in details.

Shot Clique Identification

Despite severe diversities among the Web videos related to the same event, they often share some partial overlaps in terms of shots and the repetitively occurred shots often carry important information about the event. For instance, in videos related to “Academy Awards”, shots describing ”Red Carpet”, ”Best Actor” will appear many times in different videos. In light of this observation, it is necessary to group the near-duplicate shots together into shot cliques. There are many works proposed for visual content near-duplicate detection (Shen et al. 2007; Zhao and Ngo 2009; Zhu et al. 2008; Huang et al. 2010; Song et al. 2013).

In this subsection, we apply the concept of maximally cohesive subgraph (MCS) (Huang et al. 2010) to detect and group near-duplicate shots. We first construct an undirected graph where each vertex represents the feature vector of a keyframe of a shot. An undirected edge is added between two vertices if their distance is below a threshold [1]. A shot clique is defined as a maximally cohesive subgraph representing a cluster of near-duplicate shots. A cohesive subgraph is an induced subgraph that satisfies a cohesion constraint which ensures that a graph has cohesive topological structures. Only the largest cohesive subgraph is meaningful, as it corresponds to a maximal group of near-duplicate shots. Therefore, the fast maximally cohesive subgraph mining algorithm (Huang et al. 2010) with linear time complexity is then used for real-time near-duplicate clique identification. Since the shots of poor quality can degrade user experience, we filter out some bad shot cliques similar to (Saini et al. 2012).

Semantic Inference

Utilization of semantic information can compensate visual content in describing and organizing an event. On one hand, combing semantic meaningfulness and occurrence frequency of visual content together can make representative content selection more robust. On the other hand, semantic cues play significant role in unveiling complex semantic structures of an event. In light of this, we propose to infer the semantics of the shot cliques. We first mine a set of candidate keywords from the associated textual metadata, and then estimate the relevance of shot cliques and the keywords resorting to the affluent labeled Web images.

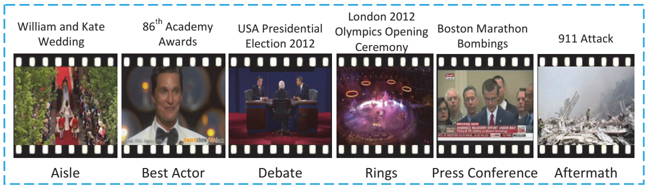

Figure 3: Examples of semantic inference results. Upper row: events; middle row: representative shots of shot cliques; bottom row: inferred semantic meaning via our approach

Keywords Mining: When uploading videos to the Web, users also provide some textual metadata (e.g., titles, descriptions, tags) to describe the content of the videos. However, such metadata is in most cases incomplete and noisy (Wang et al. 2012; Zhu et al. 2013). We propose to extract useful textual information from the metadata of all the videos related to an event to form a relatively comprehensive description for the representative content and further the event. After studying on a large amount of data, we find that titles of Web videos related to an event can provide sufficient keywords to cover key semantic facets of the event. To mine the keywords from titles, we first parse the sentences into words and phrases and eliminate the common stop words. Then we calculate the term frequency of all the remaining words and remove the words that appear with low frequency which may not cover useful information. Furthermore, words will be further filtered out based on the relevance scores between the words and shot cliques evaluated in next subsection (words that are not highly related to video content of an event will be removed). Note that phrase can also be selected as semantics using *n-gram*. For efficiency, we adopt *2-gram*.

Clique-to-Keywords Relevance and Clique Annotation: After obtaining a set of candidate keywords w.r.t. the event, we focus on evaluating a shot clique's relevance to these keywords. Instead of using labor-intensive manual labels to learn a classifier (Gan et al. 2016; Song et al. 2016), we devise a weak-supervised strategy which estimates the relevance between words and shot cliques by resorting to the affluent labeled images automatically collected from image search engines, such as Google and Bing. The rationality of exploiting the resources from the search engine is that in most cases they can provide sufficient relevant information to the query. Also, since images are inclined to be shot to record canonical scenes of an event, they can benefit the video content selection. Specifically, we use each individual keyword together with the query event as a composite query to gather a set of relevant images from the Web. For instance, if we have a candidate keyword "suspect" in the event "Boston Marathon Bombings", we use "Boston Marathon Bombings Suspect" as the query. Then, we calculate the relevance value between a shot clique and a keyword as follows:

$$rel(x, w) = |\{z \,|z \in \Omega(w); d(z, x) \leq \tau\}|, \qquad (1)$$

where x denotes keyframe of a shot clique, z denotes a Web image, w indicates a keyword, $|\cdot|$ is the cardinality of a set, $\Omega(w)$ is the image corpus of w, $d(z, x)$ measures the Euclidean distance between z and x, and τ is a predefined threshold. The relevance values are utilized to further remove candidate keywords so that at most one keyword or key phase is left in each title.

Finally, semantics of a shot clique is inferred as the keyword with largest relevance value to the clique. Figure 3 displays some examples of our semantic inference results.

Temporal Alignment

An event evolves in a certain chronological order with each subevent and/or subtopic lasting for a specific period. Only when the video content is organized according to the evolution of the event can it assist users to better understand the event. In this subsection, we aim at aligning the shot cliques in a temporal order.

First, we build a matrix $L \in \mathbb{R}^{n \times n}$ based on the pair-wise temporal orders of shot cliques obtained from the original videos as:

$$L_{i,j} = \begin{cases} 1 & \text{if } s_i \text{ is before } s_j, \\ -1 & \text{if } s_i \text{ is after } s_j, \\ 0 & \text{if not determined} \end{cases} \qquad (2)$$

[1] We use the library provided by Zhu et al. (Zhu et al. 2008) to represent a keyframe as a feature vector. Euclidean distance is used to measure the distance between feature vectors.

where $L_{i,j}$ is element (i,j) of L, s_i denotes the ith shot clique and there are n shot cliques. The temporal orders of some shot clique pairs cannot be directly observed from the original videos, but they may be inferred via some bridging shot cliques. Based on this, we conduct matrix completion for L:

$$L_{i,j} = \begin{cases} 0 \rightarrow +1 & \text{if } L_{i,p} = 1,\ L_{p,j} = 1 \\ 0 \rightarrow -1 & \text{if } L_{i,p} = -1,\ L_{p,j} = -1 \end{cases} \tag{3}$$

Next, we can align the shot cliques using the local temporal information contained in L. Specifically, we assign a temporal position t_i ($t_i \in \{1, 2, \cdots, n\}, t_i \neq t_j\ if\ i \neq j$) to each shot clique s_i in order to maximize the match between the mined temporal orders L and the estimated temporal orders:

$$t = \underset{t_i,t_j \in \{1,2,\ldots,n\}}{\arg\max} \sum_{i=1}^{n} \sum_{j=1}^{n} sgn(t_j - t_i) L_{i,j} \tag{4}$$

where $sgn(x)$ is a sign function:

$$sgn(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{if } x < 0 \end{cases} \tag{5}$$

Eq.(4) is a NP-hard problem. Alternatively, we propose an algorithm to approximate the optimal solution. To be specific, we first initialize the temporal positions of the shot cliques randomly. And then swap the temporal positions t_i and t_j of s_i and s_j iteratively as long as: a) t_i and t_j contradict with $L_{i,j}$; b) the swap can increase the objective function.

We regard all the shot cliques as a graph, where the vertices are shot cliques and an edge is added between two vertices s_i and s_j, if $L_{i,j} \neq 0$. Then the graph can be divided into a set of connected subgraphs, which means the inter-temporal-orders of the subgraphs cannot be determined from the videos. In fact, these subgraphs often correspond to independent components of the events. Since users' interest and focus towards an event change with the evolution of the event, we can employ users' interest trend to estimate the temporal orders of the independent sets of shot cliques. Upload time of video is a good indicator of users' interest trend about an event. Therefore, we calculate the average upload time of the videos related to the subgraphs to determine the temporal orders.

Semantic Smoothing

Since the pair-wise temporal information mined from original videos is incomplete, some shot cliques may be mistakenly positioned. Also, there can be shot cliques inferred with inaccurate semantics. To overcome these problems, we utilize previously inferred semantic information to smooth the ordered shot cliques. For example, if shot clique s_i owns different semantics from the neighboring shot cliques before and after it which share same semantics, it is highly possible that it has been assigned to a wrong temporal position or inferred with inaccurate semantics. Thus it will be removed and not considered in content selection.

As a high level concept, events are composed of a series of key subevents in chronological order, with each subevent lasting for a specific period. Thus, another important objective of semantic smoothness is to correspond the ordered video content to a series of subevents with each of the subevent being described by a keyword mined in Section , such as "bride arrival", "red carpet", "vows" in an event "wedding". There are two problems involved. While one problem is to determine the boundary between subevents, the other lies in choosing a proper keyword to describe a specific subevent.

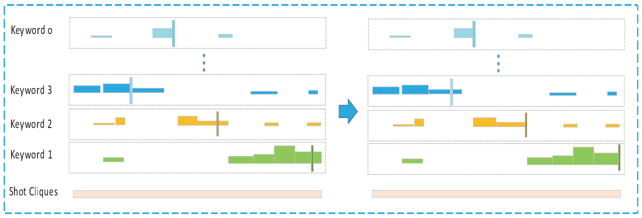

Figure 4: Illustration of subevent detection. Horizontal bars with different colors represent different keywords that are active in periods of shot cliques and the height of the bars represent the relevance between shot cliques and keywords (activeness). Vertical bars represent identified boundaries of subevents. Left: subevent boundary initialization; right: boundary refinement

We employ the relevance values w.r.t. the mined keywords as the semantic representation for the shot cliques. Specifically, suppose we have mined o keywords and we denote $r_i = \{r_{i1}, r_{i2}, \cdots, r_{io}\}$ as semantic feature for shot clique s_i, where r_{ij} is the relevance value between s_i and the jth keyword. Then the problem can be formulated as maximizing the following problem:

$$Smooth(S) = \sum_{i=1}^{k} \sum_{i_p} r_{i_p j} \tag{6}$$

where the shot cliques S are divided into k subevents, i_p denotes the pth shot clique of the ith subevent, j is the selected keyword for the ith subevent. As illustrated in Figure 4, to divide the shot cliques into subevents, we first roughly determine the boundaries of subevents via evaluating activeness of the keywords. To be specific, we initialize each keyword j with a position l that enables keyword j a maximum accumulated relevance value on a sequence of shot cliques of length m:

$$l = \underset{l \in \{m, m+2, \cdots, n\}}{\arg\max} \sum_{l'=l-m+1}^{l} r_{l'j} \tag{7}$$

This means that keyword j is most active during the period $[l - m + 1, l - m + 2, \cdots, l]$ and it is highly possible that the video content of this duration is related to subevent described by keyword j. The value m should be properly selected: if m is too large, noisy semantics would be counted when calculating the accumulated relevance value; if m is too small, more than one positions can be detected for a keyword with same accumulated relevance value. Here, we choose the average value $m = \frac{n}{o}$ (n is the number of shot cliques and o denotes the number of keywords). Since the subevents are not of equal length of m, we need to further refine the subevents boundaries. It is difficult to optimize the positions of boundaries jointly. However, since a boundary position of a subevent is only related to the neighboring two subevents, we can adjust the boundaries sequentially. For example, if a boundary I is located at l, we can adjust this position among the positions between boundary $I-1$ and $I+1$ in order that the selected position can maximize Eq.(6).

Shot Clique Selection

After the shot cliques are well organized, we focus on selecting video content to constitute the mashup video. The selected content should have high importance and encourage diversity as well.

While **importance** means that we prefer the content that is highly related to key facet of the event, **diversity** requires that the selected content should cover comprehensive subevents and the content describing same subevent should be diversified as well.

Let $\mathcal{S} = \{s_1, \cdots, s_n\}$ denotes a series of n shot cliques with accompanied keywords $\mathcal{W} = \{w_1, \cdots, w_k\}$, and $\mathcal{M} = \{s_{k_1}, \cdots, s_{k_m}\}$, $\mathcal{M} \in \mathcal{S}$ indicates an order-preserving subset of m shot cliques. Our goal is to select an optimal subset $\mathcal{M}^*$ to maximize information coverage $F(\mathcal{M})$:

$$\mathcal{M}^* = \underset{\mathcal{M} \in \mathcal{S}}{\arg\max}\, F(\mathcal{M}) \tag{8}$$

Before examining the coverage of the whole set of selected content, we consider the coverage of a single shot clique. Let $cover_{s_i}(w) : \mathcal{W} \rightarrow [0, 1]$ quantify the amount a shot clique s_i covers keyword w, which is defined as:

$$cover_{s_i}(w) = \mathcal{I}(s_i) \cdot \frac{rel(s_i, w)}{\sum_{s_j \in \mathcal{M}} rel(s_j, w)} \tag{9}$$

The first term $\mathcal{I}(s_i) = \frac{\mathcal{N}(s_i)}{\mathcal{N}(s_{max})} \in (0, 1]$ (termed content importance) is the normalized *shot frequency* of shot clique s_i, where $\mathcal{N}(s_i)$ is measured as the number of different videos where the shots in s_i appear and s_{max} indicates the shot clique that includes most frequent shots. Since the repetition of video content is not robust for important content selection as discussed, we propose the second term (termed semantic importance) to represent the relevance between s_i and a keyword w, with $rel(s_i, w)$ defined in Subsection . Note that this term is similar to *tf-idf* of w.

Based on the single shot clique coverage, it is natural to formulate the set-coverage of the selected shot cliques $cover_{\mathcal{M}}(w)$ as:

$$cover_{\mathcal{M}}(w) = \sum_{s_i \in \mathcal{M}} cover_{s_i}(w) \tag{10}$$

However, treating the set-coverage of shot cliques as an addition of single-clique coverage is not reasonable. On one hand, since shot cliques corresponding to same subevent may have information overlaps, after seeing one shot that describes some facet of the event, we may obtain less information from another shot describing the same facet. On the other hand, the additive process does not encourage diversity. Based on the intuition that the more shot cliques about same subevent are selected, the less new information can be gained, $cover_{\mathcal{M}}(w)$ can be formulated as a sampling procedure:

$$cover_{\mathcal{M}}(w) = 1 - \prod_{s_i \in \mathcal{M}} (1 - cover_{s_i}(w)) \tag{11}$$

where $cover_{s_i}(w)$ can be understood as the probability that s_i covers w and Eq.(11) the possibility that at least one shot clique covering w is selected. As can be seen, as $\mathcal{M}$ grows, adding a new shot clique can increase less and less coverage, which encourages selecting shot cliques belonging to different subevents to convey comprehensive semantic facets.

Then, we can formulate the coverage function $\mathcal{C}(\mathcal{M})$ as a weighted sum of $cover_{\mathcal{M}}(w)$:

$$\mathcal{C}(\mathcal{M}) = \sum_{w} \lambda_w cover_{\mathcal{M}}(w) \tag{12}$$

where λ_w is the *term frequency* of w in the organized shot cliques.

Apart from selecting important shot cliques spanning comprehensive facets or subevents of an event, we prefer the content within the subevents is diversified to deliver more information. This can be achieved via $\mathcal{D}(\mathcal{M})$, which is defined as:

$$\mathcal{D}(\mathcal{M}) = \sum_{i=1}^{k} \sum_{s_j \in \mathcal{S}_i} (1 - exp(-\frac{1}{\Omega}\chi^2(s_j, s_{j+1}))) \tag{13}$$

where k represents the number of discovered subevents, $\mathcal{S}_i$ denotes the shot cliques corresponding to w_i, Ω is the mean of χ^2-distances among all adjacent shot clique pairs, s_j and s_{j+1} denote feature vectors of two adjacent shot cliques within the ith subevent. It is obvious that Eq.(13) encourages the visual difference between adjacent shot cliques in order to cover comprehensive scenes of a subevent. Finally, we can define $F(\mathcal{M})$ as:

$$F(\mathcal{M}) = \lambda_1 \mathcal{C}(\mathcal{M}) + \lambda_2 \mathcal{D}(\mathcal{M}) \tag{14}$$

where λ_1 and λ_2 are the weights w.r.t the two terms.

Finding an optimal subset of shot cliques $\mathcal{M}^*$ from $\mathcal{S}$ to optimize Eq.(14) is a NP-complete problem, which means it is intractable to find the exact optimal solution. While the visual diversity term $\mathcal{D}(\mathcal{M})$ can be computed quickly, $\mathcal{C}(\mathcal{M})$ is more computationally expensive. However, inspired by (El-Arini et al. 2009), $\mathcal{C}(\mathcal{M})$ well conforms to submodularity (Nemhauser, Wolsey, and Fisher 1978), which allows a good approximation to $\mathcal{C}(\mathcal{M})$ efficiently. $\mathcal{C}_{\mathcal{M}}$ is a submodular problem since it processes the *diminishing return* property: seeing some video content s after seeing a small set A of video content can gain more information than seeing s after seeing a bigger set B containing A.

Following (Nemhauser, Wolsey, and Fisher 1978), using simple greedy algorithm can achieve a $(1 - \frac{1}{e})$ approximation of optimal value of Eq.(12). Specifically, we run m (the number of shot cliques to be selected) times of greedy algorithm. In each iteration, we select the shot clique s that can cause maximum coverage increase:

$$Increase(s|\mathcal{M}) = \mathcal{C}(s \cup \mathcal{M}) - \mathcal{C}(\mathcal{M}) \qquad (15)$$

Due to large computational cost of simple greedy algorithm, we employ the method proposed in (Leskovec et al. 2007) which is claimed to be 700 times faster than simple greedy algorithm but maintains nearly optimal solution. Other solutions can also be adopted (Lin and Bilmes 2011).

After selecting shot cliques, the next step is to choose a chain of shots from selected shot cliques for the final content composition.

Shot Selection

Shot selection is to select a sequence of shots from the selected shot cliques to finally generate a smooth mashup video, which can be defined as: given a set of shot cliques $\mathcal{M} = \{s_1, \ldots, s_m\}$, where each shot clique is composed of a set of shots, select one shot from each shot clique s_i to form a shot sequence $\mathcal{P} = \{p_{k_1}, \ldots, p_{k_m}\}$ in order that the transition smoothness of $\mathcal{P}$ is maximized.

One simple idea about maximizing the transition smoothness of $\mathcal{P}$ is to minimize the quality difference between each consecutive shot-pair. Then the problem can be formulated as:

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \sum_j dis(p_{k_j}, p_{k_{j+1}}) \qquad (16)$$

where we use normalized resolution difference to measure the quality difference.

The problem Eq.(16) can be solved using greedy algorithm. We run K iterations of greedy algorithm and in each iteration we initialize the first node of the sequence with a different shot and then select the other shots sequentially. Specifically, from each shot clique we select a shot $p_{k_{j+1}}$ that is most similar (in terms of resolution) to the shot p_{k_j} selected from last shot clique. The finally selected shot sequence form the desired mashup video.

Experiments

In this section, we evaluate the effectiveness of our approach via comparing to existing methods on a real-world event video dataset collected from YouTube. To be specific, objective experiments are conducted to verify the performance of the proposed approach in representative content selection and subjective evaluation is performed to justify the practicality of our approach.

Dataset. In order to evaluate our proposed framework, we collected 1,600 videos from the Youtube. The collected video dataset contains five topics including "social", "sports", "entertainment", "political" and "technology" and consists of eight events (see Table 1). For each event, we collect top 200 videos from Youtube. For each video, we employ the library provided by Zhu et al.(Zhu et al. 2008) to extract features for the video frames. The extracted feature is a concatenation of color histogram, color moments, edge histogram, Gabor wavelets transform, local binary pattern and gist.

Parameter Settings. The proposed framework consists of several parameters, which are tuned experimentally. For near-duplicate detection and semantic reference, we use 0.25 as the threshold to determine whether two normalized feature vectors are near-duplicates. Also, we set $\lambda_1 = 0.7$ and $\lambda_2 = 0.3$ to emphasize content importance and semantic diversity.

Baselines. To evaluate the performance of the EVM approach, we compare it with several state-of-the art methods: a) Sparse Dictionary Selection (DS) (Cong, Yuan, and Luo 2012), that focuses on selecting content in single video; and b) Near-Duplicate Detection (ND) (Wang et al. 2012), which summarizes multiple videos.

Results of Semantic Inference

In this part, we study the capability of semantic inference for shot cliques. We employ accuracy, i.e., the proportion of shot cliques that are inferred with correct semantic meaning, as metric. In the new benchmark dataset, shot cliques are annotated with proper keywords via crowd-sourcing, and the evaluation results are shown in Table.1. From Table.1, we have the following observations: 1) the number of shot cliques with semantic meaning is in general positive relevant to the complexity and duration of the event. For instance, "Boston Marathon Bombings" has more complex visual content and more semantic structure than the "Academy Awards", thus it has more shot cliques with semantic meaning. and 2) to our knowledge, the semantic inference is accurate enough for content selection for the reason that semantic smoothness is conducted when temporally aligning the shot cliques, thus inaccurately inferred content will be further filtered out.

Results of Shot Clique Selection

In this sub-experiment, we assess the effectiveness of shot clique selection process in terms of precision and recall by comparing with the state-of-the-art methods Sparse Dictionary Selection (DS) (Cong, Yuan, and Luo 2012) and Near-Duplicate Detection (ND) (Wang et al. 2012). While precision is the fraction of selected shot cliques that describe important scenes of the event, recall is the fraction of representative content that can be covered by the selected con-

Table 1: The performance of Semantic Inference. SMSC is the number of shot cliques with semantic meaning; Key. is the number of keywords extracted from the metadata related to an event; Acc. is the accuracy (%) that measures the proportion of shot cliques that are inferred with semantic meaning. LOPC is the abbreviation of the London Olympics Opening Ceremony

Event Id & Name	#SMSC	#Key.	Acc.
#1, September 11 Attacks	103	10	**0.60**
#2, Boston Marathon Bombings	204	9	**0.71**
#3, Iphone 5s Launch	110	11	**0.75**
#4, Michael Jackson Death	92	10	**0.56**
#5, 86th Academy Awards	81	10	**0.73**
#6, US Presidential Election 2012	103	12	**0.65**
#7, William and Kate's Wedding	142	14	**0.78**
#8, LOPC 2012	166	17	**0.54**

Table 2: Comparison to the state-of-the-art results in terms of recall (Rec.%) and precision (Pre.%)

Event Id	**DS**		**ND**		**ours (EVM)**	
	Pre.	Rec.	Pre.	Rec.	Pre.	Rec.
#1	0.28	0.31	0.62	0.50	**0.78**	**0.75**
#2	0.38	0.42	0.81	0.53	**0.91**	**0.82**
#3	0.42	0.35	0.76	0.74	**0.88**	**0.84**
#4	0.76	0.64	0.79	0.55	**0.90**	**0.76**
#5	0.31	0.44	0.64	0.53	**0.82**	**0.86**
#6	0.80	0.39	0.59	0.43	**0.84**	**0.82**
#7	0.44	0.41	0.82	0.70	**0.92**	**0.88**
#8	0.46	0.38	0.79	0.70	**0.86**	**0.81**
Average	**0.48**	**0.42**	**0.73**	**0.59**	**0.86**	**0.82**

tent. Also, the ground truth are manually labeled by crowdsourcing. The precision and recall for each event is calculated and show in Table.2. From results shown in Table. 2, we have several observations:

1) compared with other state-of-the-art methods, our method significantly outperforms all of them in both precision and recall. Also, Near-Duplicate detection approach (ND) performs better than the Sparse Dictionary Selection approach (DS). This is probably due to that ND only considers the redundancy of the videos, while our proposed method combines both semantic meaning and occurrence frequency to obtain rare but interesting contents.

2) Moreover, the Near-Duplicate detection approach (ND) measures the importance of content isolatedly and it does not encourage diversity, thus it is difficult for it to gain comprehensive information for an event.

3) In addition, the dictionary selection model (DS) encourages diversity but it does not explicitly consider the content importance.

Results of Video Mashup

In order to further assess usability of the generated video clips. We adopt questionnaires collected from ten users with different academic backgrounds. Given an event, users are asked to response the following question on a questionnaire:

1) Information Coverage: (Q: To what extent do you think the results cover representative content of the event?)

2) Conciseness: (Q: Is there obvious redundant content existing in the results?)

3) Effectiveness of Subevent Discovery (organizing the content into subevents): (Q: Is the content describing same subevent put together and assigned with accurate descriptions?)

4) We compare our mashup output with traditional visualization waystoryboard proposed in ND (Wang et al. 2012) to see whether our approach can enable users to better navigate and understand the event. (Q: Are the visualization results pleasing to watch and would you like to use them to learn about the event?)

To conduct this experiment, each user will firstly browse the Internet to learn the query event about 40 minutes. Then, we present each user with two videos generated by ND and EVM without telling he/she the corresponding methods. Next, they will give a score from 0-1 to express their ideal about those two videos for each question. Eventually, we calculate the average score for each event for each question and show them in Table 3. By analyzing the feedback, we find that:

Table 3: Summary of the user feedback on each questions. Here, Id is the event id, Avg. is the average score.

ID	**Q1**		**Q2**		**Q3**		**Q4**	
	ND	**EVM**	**ND**	**EVM**	**ND**	**EVM**	**ND**	**EVM**
#1	0.70	0.80	0.62	0.74	0.40	0.64	0.65	0.72
#2	0.72	0.83	0.68	0.80	0.42	0.68	0.70	0.80
#3	0.78	0.83	0.62	0.78	0.48	0.71	0.68	0.86
#4	0.76	0.81	0.66	0.85	0.42	0.55	0.69	0.82
#5	0.70	0.82	0.70	0.82	0.44	0.70	0.68	0.79
#6	0.70	0.84	0.67	0.84	0.41	0.69	0.66	0.78
#7	0.78	0.95	0.76	0.88	0.45	0.88	0.74	0.90
#8	0.74	0.88	0.72	0.75	0.36	0.73	0.70	0.82
Avg.	**0.74**	**0.85**	**0.68**	**0.81**	**0.42**	**0.70**	**0.69**	**0.81**

1) the questionnaire results are very positive on our approach. All of the users who were surveyed found the generated video clip is more useful and comprehensive than the video clip generated by the Near-Duplicate Detection (ND) approach. In general, we provide a unified video with more informative content (i.e, Q1), more conciseness information (i.e, Q2), more related subevents (i.e, Q3) and more satisfaction in terms of user experience (i.e, Q4).

2) The user feedback also points out some limitations of this approach. For instance, compared with the ND based method, the video summarization generated by our approach can cover more comprehensive content, but some abstract facets of events such as “hoax” in “Michael Jackson Death” and “truth” in “September 11 Attacks” are difficult to be represented by visual content. Therefore, the performance may degrade in some cases.

Conclusions

In this work, we proposed a novel approach, namely *event video mashup* (EVM), to combine multiple relevant Web videos together to describe the query event. With visual, semantic, temporal cues being fully explored and exploited, we selected representative video content from the disorganized videos and organized the content into subevents and temporal sequences. Specifically, since we comprehensively consider content importance, content diversity and transition smoothness, we can provide users pleasing mashup result to assist them better understand the event. In the future, we will work on the scalability issue of our approach.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2014J063, No. ZYGX2014Z007) and the National Natural Science Foundation of China (Grant No. 61502080, No. 61632007).

References

Chu, W.; Song, Y.; and Jaimes, A. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*, 3584–3592.

Cong, Y.; Yuan, J.; and Luo, J. 2012. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE TMM* 14(1):66–75.

El-Arini, K.; Veda, G.; Shahaf, D.; and Guestrin, C. 2009. Turning down the noise in the blogosphere. In *ACM SIGKDD*, 289–298.

Gan, C.; Sun, C.; Duan, L.; and Gong, B. 2016. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *ECCV*, 849–866.

Gong, B.; Chao, W.; Grauman, K.; and Sha, F. 2014. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2069–2077.

Hong, R.; Tang, J.; Tang, H.-K.; Ngo, C.-W.; Yan, S.; and Chua, T.-S. 2011. Beyond search: Event-driven summarization for web videos. *ACM TOMM* 7(4):35:1–35:18.

Huang, Z.; Hu, B.; Cheng, H.; Shen, H. T.; Liu, H.; and Zhou, X. 2010. Mining near-duplicate graph for cluster-based reranking of web video search results. *ACM Trans. Inf. Syst.* 28(4):22.

Khosla, A.; Hamid, R.; Lin, C.; and Sundaresan, N. 2013. Large-scale video summarization using web-image priors. In *CVPR*, 2698–2705.

Kim, G.; Sigal, L.; and Xing, E. P. 2014. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 4225–4232.

Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *ACM SIGKDD*, 420–429.

Lin, H., and Bilmes, J. A. 2011. A class of submodular functions for document summarization. In *ACL*, 510–520.

Liu, H.; Yu, H.; and Deng, Z. 2015. Multi-document summarization based on two-level sparse representation model. In *AAAI*, 196–202.

Ma, Y.-F.; Hua, X.-S.; Lu, L.; and Zhang, H.-J. 2005. A generic framework of user attention model and its application in video summarization. *IEEE TMM* 7(5):907–919.

Mason, R.; Gaska, B.; Durme, B. V.; Choudhury, P.; Hart, T.; Dolan, B.; Toutanova, K.; and Mitchell, M. 2016. Microsummarization of online reviews: An experimental study. In *AAAI*, 3015–3021.

Meng, J.; Wang, H.; Yuan, J.; and Tan, Y.-P. 2016. From keyframes to key objects: Video summarization by representative object proposal selection. In *CVPR*, 1039–1048.

Nemhauser, G.; Wolsey, L.; and Fisher, M. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming* 14(1):265–294.

Ngo, C.-W.; Ma, Y.-F.; and Zhang, H.-J. 2005. Video summarization and scene detection by graph modeling. *Circuits and Systems for Video Technology, IEEE Transactions on* 15(2):296–305.

Potapov, D.; Douze, M.; Harchaoui, Z.; and Schmid, C. 2014. Category-specific video summarization. In *ECCV*, 540–555.

Saini, M. K.; Gadde, R.; Yan, S.; and Ooi, W. T. 2012. Movimash: Online mobile video mashup. In *ACM Multimedia*, 139–148.

Shen, H. T.; Zhou, X.; Huang, Z.; Shao, J.; and Zhou, X. 2007. Uqlips: A real-time near-duplicate video clip detection system. In *VLDB*, 1374–1377.

Singla, A.; Tschiatschek, S.; and Krause, A. 2016. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. In *AAAI*, 2037–2043.

Song, J.; Yang, Y.; Huang, Z.; Shen, H. T.; and Luo, J. 2013. Effective multiple feature hashing for large-scale near-duplicate video retrieval. *IEEE Trans. Multimedia* 15(8):1997–2008.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *CVPR*, 5179–5187.

Song, J.; Gao, L.; Nie, F.; Shen, H. T.; Yan, Y.; and Sebe, N. 2016. Optimized graph learning using partial tags and multiple features for image and video annotation. *TIP* 25(11):4999–5011.

Tan, S.; Tan, H.-K.; and Ngo, C.-W. 2010. Topical summarization of web videos by visual-text time-dependent alignment. In *ACM Multimedia*, 1095–1098.

Truong, B. T., and Venkatesh, S. 2007. Video abstraction: A systematic review and classification. *ACM TOMM* 3(1).

Wang, M.; Hong, R.; Li, G.; Zha, Z.-J.; Yan, S.; and Chua, T.-S. 2012. Event driven web video summarization by tag localization and key-shot identification. *IEEE TMM* 14:975–985.

Yeo, D.; Han, B.; and Han, J. H. 2016. Unsupervised co-activity detection from multiple videos using absorbing markov chain. In *AAAI*, 3662–3668.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016. Video summarization with long short-term memory. *arXiv preprint arXiv:1605.08110*.

Zhao, W.-L., and Ngo, C.-W. 2009. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. *IEEE TIP* 18:412–423.

Zhao, B., and Xing, E. P. 2014. Quasi real-time summarization for consumer videos. In *CVPR*, 2513–2520.

Zhu, J.; Hoi, S. C.; Lyu, M. R.; and Yan, S. 2008. Near-duplicate keyframe retrieval by nonrigid image matching. In *ACM Multimedia*, 41–50.

Zhu, X.; Huang, Z.; Cui, J.; and Shen, H. T. 2013. Video-to-shot tag propagation by graph sparse group lasso. *IEEE TMM* 15(3):633–646.