The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Video Summarization via Semantic Attended Networks

Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, Xiaokang Yang
Shanghai Key Laboratory of Digital Media Processing and Transmission, Shanghai Jiao Tong University
weihuawei26@gmail.com,{nibingbing,yanyichao,yuhuanyu,xkyang}@sjtu.edu.cn

Abstract

The goal of video summarization is to distill a raw video into a more compact form without losing much semantic information. However, previous methods mainly consider the diversity and representation interestingness of the obtained summary, and they seldom pay sufficient attention to semantic information of resulting frame set, especially the long temporal range semantics. To explicitly address this issue, we propose a novel technique which is able to extract the most semantically relevant video segments (*i.e.*, valid for a long term temporal duration) and assemble them into an informative summary. To this end, we develop a *semantic attended video summarization network (SASUM)* which consists of a frame selector and video descriptor to select an appropriate number of video shots by minimizing the distance between the generated description sentence of the summarized video and the human annotated text of the original video. Extensive experiments show that our method achieves a superior performance gain over previous methods on two benchmark datasets.

Introduction

With the upgrade of storage hardware and the faster and faster internet speed, video recording is becoming cheaper and more convenient. However, there is a large amount of useless information in the stored content. Therefore, automatic video summarization (Elkhattabi, Tabii, and Benkaddour 2015) is an urgent problem to be solved, which can not only save the storage resources but also save time for people to browse videos.

The redundancy of videos includes content redundancy and semantic redundancy. The best summary must satisfy compactness and the relevance with the video's topic. Previous video summarization methods (Gong et al. 2014; Zhao and Xing 2014) mostly try to filter the repeated content of a video but do not remove parts irrelevant with the video theme. Recently, some methods have been used to alleviate this issue by using exemplar videos or web images as priors (Khosla et al. 2013; Zhang et al. 2016a). However, they are only suitable for summarizing category-specific videos because of the need to collect a large number of videos on the same topic. Moreover, summaries generated by these methods are not semantically rich due to their attention is paid on the relevance with the exemplars.

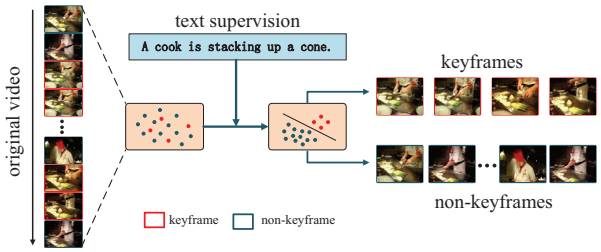
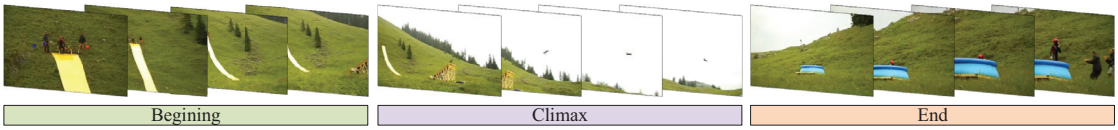


Figure 1: Our network can distinguish keyframes and non-keyframes determined by their relevance with the text description.

With the expanding of research on cross-domain, especially from vision to language, many applications, such as image retrieval (Ma et al. 2015) and video description (Venugopalan et al. 2015) have achieved great success. It's a natural idea to apply this technique on video summarization by adding language supervision. Lately, Plummer *et al.* (2017) and Choi *et al.* (2017) take advantage of text annotations to produce semantically rich video summaries assisted by submodular function or hidden Markov model. However, their proposed methods only exploit image-text embedding models (Ma et al. 2015) such that their methods cannot capture video's continuous context.

In order to obtain more story-telling summaries, the relationship between dynamic visual content and its corresponding high-level semantics should be taken into account. To this end, we develop a *semantic attended video summarization network (SASUM)*, which considers about not only the semantic richness but also the context continuity of the generated summary. Our inspiration comes from the fact that we humans tend to filter out frames that are independent of the meaning of the video when picking out the key parts of a video. As shown in Figure 1, according to the description "a cook is stacking up a cone", we certainly will not select the first frame whose content is a cook is mixing meat, which is semantically unrelated to the provided description. To mimic this selection manner, we train our *SASUM* su-



1.	*A man slides on a long sliding plate.*	*A man flies in the sky.*	*A man falls into the pool.*
2.	*A man slides on a yellow surface.*	*A man is in the air.*	*A man drops into the pool.*
3.	*A man slides down the slope.*	*A man flies through the air.*	*A man falls into the tank.*

Figure 2: A example of our annotated datasets. The video is split into 3-5 segments including the beginning, the climax and the end, then each segment is described by three sentences annotated by three different workers.

pervised by human annotated text descriptions to select the most semantically representative video shots. By minimizing the distance between the generated description of the summarized video and the ground-truth annotation, our network is capable of emphasizing importance on frames with relevant semantics to the video's theme.

The architecture of our approach is illustrated in Figure 3, which consists of a frame selector and video descriptor. Given a video, we first pass it through a Convolutional Neural Network (CNN) (Szegedy et al. 2016) to extract high-level semantic features. Then the features are fed to the selector which consists of a Long Short-Term Memory Network (LSTM) (Hochreiter and Schmidhuber 1997). In the training phase, under the supervision of text description, the selector will select the segments that best represent the video's topic. The descriptor is composed of an Encoder-Decoder structure (Venugopalan et al. 2015). For exploring the bidirectional temporal information of videos, we exploit a bidirectional LSTM network (BiLSTM) (Bin et al. 2016) as our Encoder whose input is the features weighted by the selector. Due to the fact that the output of the last node of LSTM tends to retain more information closer to it, we fuse the outputs of all nodes via average pooling as the Decoder's input to alleviate the negative effect caused by LSTM's time tendency. The Decoder is a simple LSTM network which can map visual information to a text representation. The frame selector and video descriptor are jointly trained so as to force the network to learn how to assign importance to frames guided by the text annotations. We also add two types of sparsity constraints to the frame selector for limiting the length of the generated summary or making full use of the human annotated keyframes to generate summaries with better human correspondence. In addition, we produce coarse-grained text annotations for two publicly datasets (Gygli et al. 2014; Song et al. 2015), unlike the settings of existing video description datasets (*e.g.*, YouTube2Text and M-VAD) (Chen and Dolan 2011; Torabi et al. 2015) which only provide short video clip and sentence pairs, each text description of our provided annotations is corresponding to a long term temporal video duration so that a modeling of long temporal range semantics can be supported.

Experiments on two benchmark datasets (*i.e.*, SumMe and TVSum) (Gygli et al. 2014; Song et al. 2015) incorporated by the text annotations we develop demonstrate our proposed model can really capture the mapping from vision to language and remove frames that are not related to the video's topic semantics. Our main contributions consist of:

1. A long temporal range semantics guided method for video summarization is first specified, which is capable of extracting the most semantically coherent subshots in the video.

2. Coarse-grained text annotations for two publicly video summarization datasets are provided so that the mapping from videos to integral semantics can be attained.

Related Work

Video summarization can generally be separated into two categories: content aware summarization and semantic aware summarization, the former typically emphasizes the diversity and representation interestingness in the generated summaries while the latter pays more attention to high-level context information to guarantee the summaries' semantic coherence. In this section, we first review recent work on this two types of video summarization manners, and then give a brief introduction to video description, the technique we will utilize in our proposed framework will be offered.

Content Aware Summarization

Early works mainly exploit low-level visual features, such as color histogram (De Avila et al. 2011), motion cues (Gygli et al. 2014; Ren et al. 2017) and spatiotemporal features (Laganière et al. 2008) to extract the most interesting frames. However, these low-level features based methods can rarely retain semantic concepts which very affects the viewing experience. Lately, some deep architectures (*e.g., VAE, LSTM*) are applied to attain high-level categories (Mahasseni, Lam, and Todorovic 2017; Zhang et al. 2016b), but they are almost modeled in terms of visual diversity by minimizing the reconstruction error between original video and the summarized one or utilizing determinantal point process (DPP), a probabilistic model for diverse subset selection. Generally speaking, these video summarization methods unilaterally focus on the diversity without considering context, so they can only generate content rich but not semantic deficient summaries.

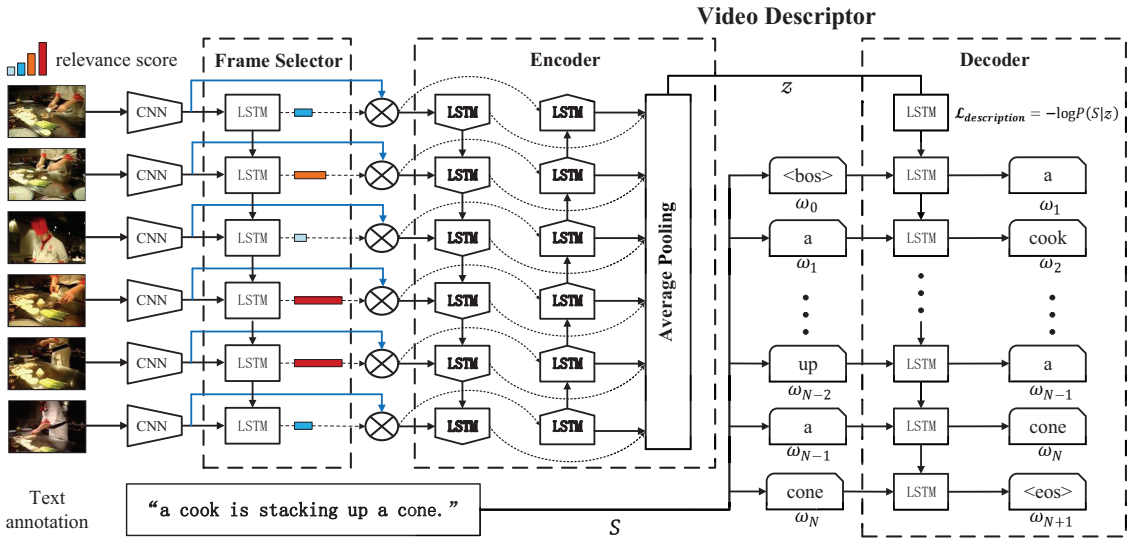


Figure 3: Overview of our *SASUM* architecture. The CNN features are first weighted by the relevance scores output from the frame selector. The relevance scores are indicated by the colored bars, where longer bars denote higher scores. Then the video descriptor, which is an Encoder-Decoder structure, will translate the visual content to a text description. By minimizing the distance between the generated description and the human written annotation, the network will select the most semantically representative video segments.

Semantic Aware Summarization

More recent works notice that adding semantic supervision is extremely conducive to produce more context coherent summaries. Sharghi *et al.* (2016) propose a noun-based video summarization approach, when given a long video sequence, the algorithm will return the keyshots which have high relevance to the predefined nouns. Another method of implicitly using semantics is developed by Chu *et al.* (2015), which points out that similar visual concepts tend to occur in videos with a similar theme, therefore, shots that co-occur most frequently among videos can be extracted to form a summary.

In order to explicitly make use of semantic information, Yeung *et al.* (2014) present a text-based video summary evaluation approach, which uses a NLP-based metric to measure the semantic distance between the generated text representation of the video summary and the ground-truth text summaries. Although this method is very innovative and easy to operate, it takes no consideration of visual quality as long as the content is shown. Choi *et al.* (2017) put forward another method on the basis of this work. Given a raw video, the semantically relevant frames can be extracted by measuring the similarity between the frames' vision-language embedding and the user-specific descriptions. However, this approach only takes into account the relationship of images and text, instead of a long temporal range of visual information and its integral semantics. Our work will explicitly address this issue by directly using a video descriptor to model the relationship of the entire video and its overall context information.

Video Description Models

The Encoder-Decoder structure (Venugopalan et al. 2015; Ji et al. 2017) is most commonly exploited to translate videos to language (*i.e.*, Video Description). This structure typically employs a deep LSTM network to map the visual sequence to a fixed-dimensionality vector, and then another deep LSTM network will decode the language sequence from the vector (Xu et al. 2016; Venugopalan et al. 2015). Video Description has a wide range of applications in video indexing and movie describing (Xu et al. 2016; Torabi, Tandon, and Sigal 2016). But to our knowledge, this is the first time that this t is applied on video summarization. We connect it to a frame selector so as to guide the selector to find the most semantically relevant video frames.

Semantic Attended Network

We cast video summarization as a semantic-guided subshots selection problem. The overview of our developed semantic attended network (*SASUM*) is illustrated in Figure 3, which consists of an Encoder-Decoder model of video description and a frame selector, the former aims at mapping the input visual information into text representation, and the latter utilizes the feedback from the former to find video keyframes that are the most relevant to the high-level context.

In this section, we first describe the video datasets for which we have provided text annotations that can be used in our approach, followed by a detailed introduction of the proposed semantic attended video summarization network. Then we introduce in detail our proposed network as well as the training procedure.

Datasets and Annotating Protocol

There are previous works using semantic information to perform video summarization (Plummer, Brown, and Lazebnik 2017; Choi, Oh, and Kweon 2017), but the datasets they use only provide short video segment and sentence pairs. Therefore, the proposed methods typically extract a frame from the video clip and train an image-text embedding model to select the most semantically representative frames using either submodular function or greedy algorithm. Obviously, such datasets cannot be exploited to model long term semantics. To this end, we provide text annotations for two currently popular datasets, *i.e.*, SumMe and TVSum (Gygli et al. 2014; Song et al. 2015). The details are as follows:

SumMe (Gygli et al. 2014) consists of 25 user videos which capture multiple events such as sports and cooking. Lengths of the videos vary from 1 to 6.5 minutes. TVSum (Song et al. 2015) contains 50 videos from YouTube in 10 categories such as animal grooming, parkour and dog show. The lengths vary from 2 to 10 minutes. Both of these two datasets contain contents with ego-centric and third-person camera. Besides,they also provide importance score for each video frame, which we will use as supervision information in our method.

In order to enable our network to capture long temporal range semantic information, we first apply the *Kernel Temporal Segmentation* (KTS) (Potapov et al. 2014) to split videos of SumMe and TVSum into short shots, and then combine these small pieces into 3 to 5 visually coherent segments with the time order preserved, the number of segments depends on the length of the video. Furthermore, each segment is annotated with three description sentences written by three different workers. The descriptions are further edited by additional workers to guarantee vocabulary and grammatical consistency. The reason why we follow this protocol to split the video is that we assume each video contains an event, which usually includes the beginning, the climax, and the end. In fact, these videos do satisfy this assumption. An example is depicted in Figure 2.

Attending to Video Semantics

Our proposed architecture consists of a frame selector and a video descriptor composed of an Encoder-Decoder structure, as illustrated in Figure 3.

Frame selector. Given a long video sequence $X = \{x_1, x_2, \cdots, x_T\}$, where T is the length of the sequence, the deep features extracted by CNN are denoted as $V = \{v_1, v_2, \cdots, v_T\}$. We then implement a variable length LSTM network, called frame selector, to select a set of keyframes. The output of frame selector are relevance scores, namely importance score of the video frames, denoted by $R = \{r_1, r_2, \cdots, r_T\}$. The scores are normalized to [0, 1]. The "relevance" here refers to the correlation between video content and high-level semantics. Note that if the scores are integerized into $\{0, 1\}$, they can be regarded as an indicator, when $r_t = 1$, the corresponding frame is selected and vice versa. After weighted processing, the frames' CNN feature can be represented as $\hat{V} = V \odot R$, where $\odot$ denotes element-wise multiplication.

Encoder-Decoder structure. We utilize an Encoder-Decoder structure to build our video descriptor, which is now the most appropriate framework for modelling serialized data.

When we humans summarize a video, we usually consider the context of the video and then extract the most semantically representative parts to form the final summary. To mimic this manner, we choose bidirectional LSTM network (BiLSTM) (Bin et al. 2016) as the encoder, which has a superior performance to capture context information in long term temporal video sequence. Feeding the weighted CNN features $\hat{V}$ to the Encoder, a fix-length vector $\boldsymbol{z} = E(\hat{v}_1, \hat{v}_2, \cdots, \hat{v}_T)$ will be output. In practice, this is performed by encoding $\hat{V}$ into a sequence of hidden state vectors h_e^t by BiLSTM, the evolution procedure can be illustrated as:

$$h_e^t = \left[h_{fw}^t, h_{bw}^t\right], \tag{1}$$

$$\theta_e = [\theta_{fw}, \theta_{bw}], \tag{2}$$

$$h_{fw}^t = \phi_{fw}\left(\hat{v}_t, h_{fw}^{t-1}; \theta_{fw}\right), \tag{3}$$

$$h_{bw}^t = \phi_{bw}\left(\hat{v}_t, h_{bw}^{t-1}; \theta_{bw}\right), \tag{4}$$

where $t = 1, 2, \cdots, T$, fw and bw represent *forward* and *backward* LSTM and their corresponding nonlinear mapping functions are ϕ_{fw} and ϕ_{bw}, whose parameters are θ_{fw} and θ_{bw}. And then $\boldsymbol{z}$ is obtained via a mean pooling of $\left\{h_e^1, h_e^1, \cdots, h_e^T\right\}$: $\boldsymbol{z} = \frac{1}{T}\sum_{t=1}^{T} h_e^t$. In standard Encoder-Decoder structure, the value of $\boldsymbol{z}$ is directly assigned by the last timestep's state, *i.e.*, $\boldsymbol{z} = h_e^T$. The fact why we do not choose this approach is that LSTM tends to emphasize more importance on current timestep, namely, the output of current LSTM timestep prefers to retain more information of current input. However, our video summarization policy is determined by frames' semantics, so we expect the network to allocate importance to each frame based on visual content. Therefore, we select mean pooling instead of the aforementioned manner to attain the encoded representation $\boldsymbol{z}$, which can alleviate the negative effect of LSTM's time tendency.

After obtaining the encoded representation $\boldsymbol{z}$, the decoder converts it to a words sequence $S = \{\omega_1, \omega_2, \cdots, \omega_N\}$, and each ω denotes a word. In addition, we use ω_0 and ω_{N+1} to indicate the beginning-of-sentence (<bos>) and the end-of-sentence (<eos>) tag. Given all the previous generated words, the conditional probability distribution of the current word is expressed as below:

$$P(\omega_n|\omega_0, \omega_1, \cdots, \omega_{n-1}, \boldsymbol{z}; \theta_d) = D\left(\omega_{n-1}, h_d^t, \boldsymbol{z}; \theta_d\right), \tag{5}$$

$$h_d^t = \psi\left(\omega_{n-1}, h_d^{t-1}, \boldsymbol{z}; \theta_d\right), \tag{6}$$

where h_d^t is the hidden state of decoding LSTM, ψ is the nonlinear mapping function and θ_d is the parameter of the decoder. Then the probability distribution of the generated sentence can be deduced by the joint probability of all words:

$$P(S|\boldsymbol{z}) = \prod_{n=1}^{N} P(\omega_n|\omega_0, \omega_1, \cdots, \omega_{n-1}, \boldsymbol{z}; \theta_d). \tag{7}$$

As mentioned, we directly feed a long video sequence to our semantic attended network, then the visual information will be translated to a text representation (*i.e.*, description sentence). In this manner, our framework can gradually learn the mapping between video content and its long temporal range semantics.

Network Training

To optimize the proposed networks, a loss function and types of sparsity constraints are defined. Then the parameters of our network can be gradually updated by reducing these loss terms using *Stochastic Gradient Descent* (SGD).

Caption loss. As mentioned, the probability distribution of the sentence generated by the video description model is denoted by $P(S|\boldsymbol{z})$. To make the generated text sentence approach the given ground truth statement, we can define a description loss $L_{description}$ depicted as:

$$\mathcal{L}_{description} = -\log P(S|\boldsymbol{z}). \tag{8}$$

Sparsity constraints. Following the work developed by Mahasseni *et al.* (2017), we use two variants of sparsity constraints $\mathcal{L}_{sparsity}$ to force the frame selector to generate high-quality video summaries.

The first variant is a penalty that restricts the length of the summaries:

$$\mathcal{L}^{length}_{sparsity} = \left| \frac{1}{T} \sum_{t=1}^{T} r_t - \delta \right|, \tag{9}$$

where δ is the percentage of the video's length that we expect to be preserved in the produced summary.

In order to make full use of the annotations provided in SumMe and TVSum, we develop another sparsity constraint approach for our architecture, where we provide the annotated keyframes $\hat{R} = \{\hat{r}_1, \hat{r}_2, \cdots, \hat{r}_T\}$ in training phase, then the sparsity loss can be denoted as the cross-entropy loss:

$$\mathcal{L}^{sup}_{sparsity} = \frac{1}{T} \sum_{t=1}^{T} \left(\hat{r}_t \log r_t + (1 - \hat{r}_t) \log(1 - r_t) \right). \tag{10}$$

Parameter update. The frame selector and the Encoder-Decoder video description model can be characterized by parameters θ_s and the aforementioned parameters $\{\theta_e, \theta_d\}$. Our goal is optimize these parameters by maximally reducing the above defined loss terms: $\{\mathcal{L}_{description}, \mathcal{L}_{sparsity}\}$. Algorithm 1 describes the training steps of our approach in detail.

Algorithm 1 Training semantic attended network

Input: : Segments of original videos
Output: : Optimized parameters $\{\theta_s, \theta_e, \theta_d\}$
Initialize network parameters $\{\theta_s, \theta_e, \theta_d\}$
for *iteration* = 1 to *max iterations* **do**
  $X \leftarrow$ mini-batch from video segments
  $V = CNN(X)$ % *extract CNN features*
  $R = Selector(V)$ % *compute relevance scores*
  $\boldsymbol{z} = Encoder(V, R)$ % *encoding*
  $S = Decoder(\boldsymbol{z})$ % *decoding sentences*
  % Update parameters using SGD
  $\{\theta_e, \theta_d\} \overset{+}{\leftarrow} -\nabla(\mathcal{L}_{description})$
  $\{\theta_s\} \overset{+}{\leftarrow} -\nabla(\mathcal{L}_{description} + \mathcal{L}_{sparsity})$
end for

Experiments

Experimental Setup

Datasets. We evaluate our approach on two video datasets, SumMe (Gygli et al. 2014) and TVSum (Song et al. 2015) annotated with text descriptions created by us. The details have been presented in Method.

Evaluation. Following the protocols in (Gygli et al. 2014; Zhang et al. 2016b), we assess the performance of the generated summary by measuring its agreement with the human annotated ground truth summary. Let A denotes the generated summary and B the ground truth summary, the *precision* P and *recall* R can be obtained as below:

$$P = \frac{overlapped\ duration\ of\ A\ and\ B}{duration\ of\ A}, \tag{11}$$

$$R = \frac{overlapped\ duration\ of\ A\ and\ B}{duration\ of\ B}, \tag{12}$$

then the F-measure is computed as:

$$F = \frac{2 \times P \times R}{P + R} \times 100\%. \tag{13}$$

We follow the method proposed by Zhang *et al.* (2016b) to convert frame-level relevance scores into keyshots. We first split the whole video into short intervals using KTS (Potapov et al. 2014) and then compute interval-level score by averaging the scores of the frames within the interval. The intervals are ranked in the descending order based on their scores. Then we select the keyshots from the ranked intervals such that the total duration of the keyshots is less than 15% of the original video's duration.

For each benchmark, We randomly select 80% for training and the remaining 20% for testing. For fair comparison, we run this procedure for 10 times and report the average performance as the final result.

Implementation details. We use the output from the average pooling layer *pool_3* of InceptionV3 (Szegedy et al. 2016), which is pretrained on ImageNet (Deng et al. 2009), as our feature (2048-dimensions) of each frame. Both the frame selector and the decoder of the video description model are a single-layer LSTM network, and the encoder of the video description model is a bidirectional LSTM work; all these LSTM networks include 1024 hidden units. To obtain abundant high-level categories and concepts, we pretrain the video description model over a large-scale vision-language dataset (MSR-VTT) (Xu et al. 2016). We train our networks with Adam optimizer with initial learning rate 0.0001. All experiments are conducted on the GTX TITAN X GPU using Tensorflow (Abadi et al. 2016).

Method	SumMe	TVSum
Gygli et al.	39.7%	-
dppLSTM	38.6%	54.7%
Zhang et al.	40.9%	-
SUM-GAN$_{sup}$	41.7%	56.3%
SASUM	40.6±0.2%	53.9±0.3%
SASUM$_{length}$	41.0±0.1%	54.6±0.2%
SASUM$_{sup}$	**45.3±0.1%**	**58.2±0.2%**

Table 1: Performance comparison (F-Score) between our frameworks and four state-of-the-art methods on SumMe and TVSum benchmarks.

Method	YouTube
Gygli et al.	-
dppLSTM	-
Zhang et al.	61.8%
SUM-GAN$_{sup}$	**62.5%**
SASUM	57.6±0.1%
SASUM$_{length}$	58.5±0.3%
SASUM$_{sup}$	60.3±0.3%

Table 2: Performance comparisons on YouTube benchmark.

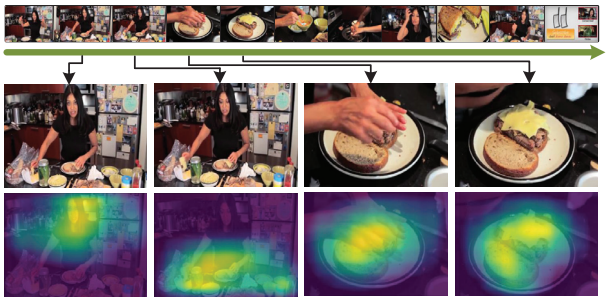
Compared methods. In our evaluation, we select four state-of-the-art video summarization approaches to be compared with our proposed framework: 1) a method proposed by Gygli *et al.* (2015) which formulates video summarization as a submodular maximization problem; 2) dppLSTM (Zhang et al. 2016b), a LSTM based method which utilizes determinantal point process (DPP) so as to select content diverse keyframes; 3) an approach developed by Zhang *et al.* (2016a) which leverages supervision in the form of human-annotated summaries to perform video summarization; 4) SUM-GAN$_{sup}$ (Mahasseni, Lam, and Todorovic 2017), a video summarization framework based on VAE and GANs. The above-mentioned video summarization approaches are all performed in a supervised manner.

In addition, three variants of our proposed method are also included for comparison:

- *SASUM*: our proposed semantic attended video summarization network without any sparsity constraints.
- *SASUM*$_{length}$: our proposed semantic attended video summarization net work with length constraint $\mathcal{L}_{sparsity}^{length}$.
- *SASUM*$_{sup}$: our proposed semantic attended video summarization network supervised by human annotated summaries $\mathcal{L}_{sparsity}^{sup}$.

Results

We present both quantitative and qualitative results of our proposed semantic attended video summarization framework. Meanwhile, a deep analysis of our method's advantages compared with other state-of-the-art approaches is provided. In addition, a visualization technique (Ramanishka et al. 2016) is used to show our method's ability to translate vision to language.



A woman is making a sandwich.

Figure 4: Visualization of the mapping from vision to language. Our network can capture the correspondence between visual content and high-level semantics.

Quantitative results. Table 1 shows the compared results of the three variants of our framework and four state-of-the-art methods on SumMe and TVSum benchmarks.

We first compare the results of three variants of our approach. As can be observed, *SASUM*$_{sup}$ outperforms the other two variants by 4% to 5%. The reasons are as follows: *SASUM*$_{sup}$ exploits human annotated labels as supervision, as a result, the generated summaries have a better human correspondence than *SASUM* and *SASUM*$_{length}$. As for the better performance of *SASUM*$_{length}$ (δ = 0.3) than *SASUM*, that's because when adding length sparsity constraint, the network is forced to reconstruct the original video's high-level context from its subset. Then attention will be paid to the most semantically representative frames and little attention on semantically irrelevant frames, such that keyframes and non-keyframes are easy to be distinguished. While in *SASUM*, there is no such constraint to guarantee keyframes selection with the above mentioned strong differentiation.

Compared with the state-of-the-art methods, our framework attains comparable or even better performance. Notice that *SASUM* and *SASUM*$_{length}$ only use weak supervision, that is, our provided annotated descriptions, but their results are almost on a par with the completely supervised methods which leverage the human annotated video summaries.

In particular, *SASUM*$_{sup}$ outperforms all referred approaches in all datasets. An interesting observation is that our approach outperforms much on SumMe than TVSum by 4% and 2%. It's mainly due to the fact that videos of SumMe have slowly changing content and few objects in the scene, which is very beneficial for our semantic attended network to capture the scene's context. While in TVSum, the scene is changeable so that the mapping from vision to language is difficult to perform.

Moreover, to verify the generalization performance of our method, we perform experiments on another dataset with no text annotations, YouTube, which includes 50 videos whose content contains cartoons, news, *etc.*. The result is shown in Table 2. We can notice that even no knowledge about this dataset is utilized, our method still achieves very competitive performance compared with other state-of-the-art methods supervised by human annotated labels.

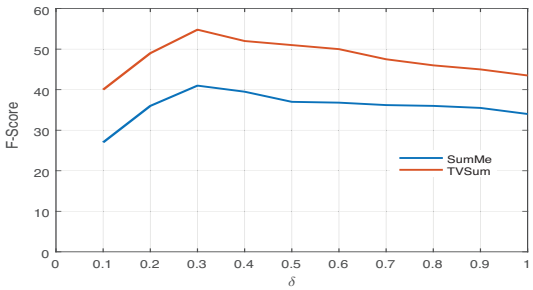


Figure 5: Performance of $SASUM_{length}$ on SumMe and TVSum benchmarks is affected by different values of δ.

Visualization of mapping from vision to language. In order to verify the fact that our network does capture the correspondence between visual content and high-level semantics, we apply the visualization technique developed by Ramanishka *et al.* (2016) to demonstrate it. As depicted in Figure 4, given a sentence "A woman is making a sandwich.", our network can generate spatiotemporal heatmaps of the corresponding visual objects.

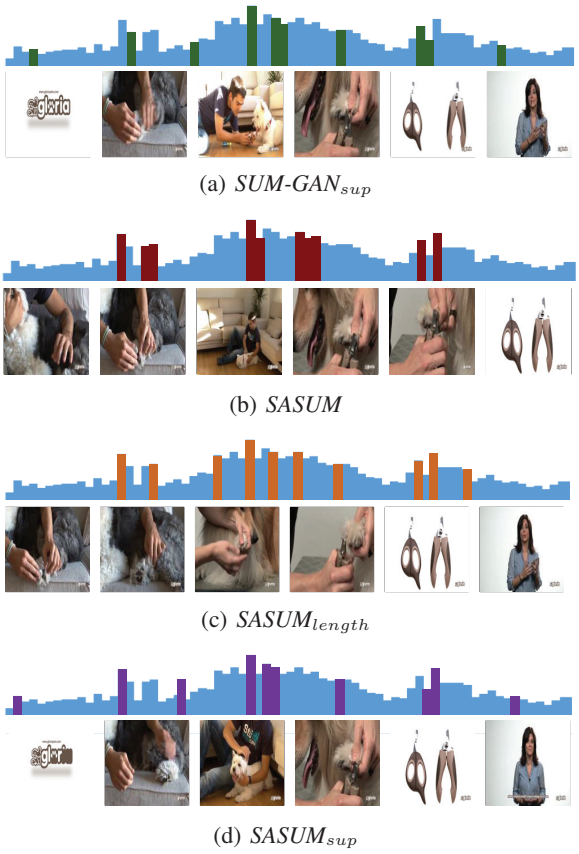
Sensitivity analysis of hyper parameters. We evaluate how the values of the hyper parameter δ affect the performance of $SASUM_{length}$. The result is illustrated in Figure 5. As depicted, the most suitable value of δ is 0.3, and the farther the value is away from 0.3, the worse the performance will be. It can be interpreted by the reason that nearly 70% content of the videos of these two datasets is semantically redundant.

Qualitative results. To explicitly illustrate the difference between our method and the previous approaches as well as the generality and individuality of our three variants, we demonstrate example summaries generated by *SUM-GAN*$_{sup}$ and our approaches in Figure 6, where the averaged human annotated importance score are indicated by the blue bars and the colored bars represent the extracted summaries.

We first compare the differences between our approaches. As can be observed, the summary generated by $SASUM_{length}$ is more sparse in time than the other two, that's because the length constraint will force the network to place attention on the most semantically representative video intervals. However, adjacent intervals are usually semantically similar in content so that they will not be selected simultaneously. On the other hand, the summary generated by $SASUM_{sup}$ has a better uniformity in time, the reasons are as follows: when supervised by human annotated labels, the network will try to cater different annotators' preferences, while different annotators pay attention to different parts of the video, so the network will generate summaries with a long temporal range.

As for the generality, we can observe that all three summaries cover the intervals with high importance scores. This can be explained that intervals with high scores usually represent the semantically rich snippets in the video, and all our variants can generate summaries with rich semantics.

Moreover, compared with our approaches, the summary



(a) *SUM-GAN*$_{sup}$

(b) *SASUM*

(c) $SASUM_{length}$

(d) $SASUM_{sup}$

Figure 6: An example of the generated summaries of a sample video in TVSum.

produced by *SUM-GAN*$_{sup}$ has less coverage than ours in high-score regions, namely semantically rich segments. That's because there is no semantic supervision in *SUM-GAN*$_{sup}$, such that this method can rarely extract semantically rich shots.

Conclusion

In this paper, we propose a novel *semantic attended video summarization network (SASUM)* which is able to extract semantically relevant video segments and assemble them into an informative summary. To assist our network to model the relationship of the long temporal video content and its integral context information, we provide text annotations for two publicly datasets. Extensive experimental results demonstrate that our network does capture the correspondence between the visual content and the text description.

Acknowledegment

This work was supported by NSFC (61502301, 61671298, U1611461), Chinas Thousand Youth Talents Plan, STCSM17511105401, the Ministry of Public Security project C17384 and the Opening Project of the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Bin, Y.; Yang, Y.; Shen, F.; Xu, X.; and Shen, H. T. 2016. Bidirectional long-short term memory for video description. In *ACMMM*, 436–440. ACM.

Chen, D. L., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 190–200. Association for Computational Linguistics.

Choi, J.; Oh, T.-H.; and Kweon, I. S. 2017. Textually customized video summaries. *arXiv preprint arXiv:1702.01528*.

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

De Avila, S. E. F.; Lopes, A. P. B.; da Luz, A.; and de Albuquerque Araújo, A. 2011. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters* 32(1):56–68.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 248–255. IEEE.

Elkhattabi, Z.; Tabii, Y.; and Benkaddour, A. 2015. Video summarization: Techniques and applications. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* 9(4):928–933.

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

Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating summaries from user videos. In *ECCV*, 505–520. Springer.

Gygli, M.; Grabner, H.; and Van Gool, L. 2015. Video summarization by learning submodular mixtures of objectives. In *CVPR*, 3090–3098.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Ji, Z.; Xiong, K.; Pang, Y.; and Li, X. 2017. Video summarization with attention-based encoder-decoder networks. *arXiv preprint arXiv:1708.09545*.

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

Laganière, R.; Bacco, R.; Hocevar, A.; Lambert, P.; Païs, G.; and Ionescu, B. E. 2008. Video summarization from spatio-temporal features. In *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, 144–148. ACM.

Ma, L.; Lu, Z.; Shang, L.; and Li, H. 2015. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, 2623–2631.

Mahasseni, B.; Lam, M.; and Todorovic, S. 2017. Unsupervised video summarization with adversarial lstm networks. In *CVPR*.

Plummer, B. A.; Brown, M.; and Lazebnik, S. 2017. Enhancing video summarization via vision-language embedding. In *CVPR*.

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

Ramanishka, V.; Das, A.; Zhang, J.; and Saenko, K. 2016. Top-down visual saliency guided by captions. *arXiv preprint arXiv:1612.07360*.

Ren, Z.; Yan, J.; Ni, B.; Liu, B.; Yang, X.; and Zha, H. 2017. Unsupervised deep learning for optical flow estimation.

Sharghi, A.; Gong, B.; and Shah, M. 2016. Query-focused extractive video summarization. In *ECCV*, 3–19. Springer.

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

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, 2818–2826. IEEE Computer Society.

Torabi, A.; Pal, C.; Larochelle, H.; and Courville, A. 2015. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*.

Torabi, A.; Tandon, N.; and Sigal, L. 2016. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *ICCV*, 4534–4542.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 5288–5296.

Yeung, S.; Fathi, A.; and Fei-Fei, L. 2014. Videoset: Video summary evaluation through text. *arXiv preprint arXiv:1406.5824*.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016a. Summary transfer: Exemplar-based subset selection for video summarization. In *CVPR*, 1059–1067.

Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016b. Video summarization with long short-term memory. In *ECCV*, 766–782. Springer.

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