Video to Text Summary: Joint Video Summarization and Captioning with Recurrent Neural Networks

Bor-Chun Chen[1]
sirius@umd.edu
Yan-Ying Chen[2]
yanying@fxpal.com
Francine Chen[2]
chen@fxpal.com

[1] University of Maryland
College Park, Maryland, USA

[2] FX Palo Alto Laboratory, Inc.
Palo Alto, California, USA

Abstract

Video summarization and video captioning are considered two separate tasks in existing studies. For longer videos, automatically identifying the important parts of video content and annotating them with captions will enable a richer and more concise condensation of the video. We propose a general neural network configuration that jointly considers two supervisory signals (i.e., an image-based video summary and text-based video captions) in the training phase and generates both a video summary and corresponding captions for a given video in the test phase. Our main idea is that the summary signals can help a video captioning model learn to focus on important frames. On the other hand, caption signals can help a video summarization model to learn better semantic representations. Jointly modeling both the video summarization and the video captioning tasks offers a novel end-to-end solution that generates a captioned video summary enabling users to index and navigate through the highlights in a video. Moreover, our experiments show the joint model can achieve better performance than state-of-the-art approaches in both individual tasks.

1 Introduction

The prevalence of recording devices encourages more people to capture their daily life with video data. However, the sheer amount of video data makes it hard to review and navigate, particularly long videos such as surveillance videos and life-logging videos. The major problems may be attributed to the scarcity of compactness and semantics in the video content. To this end, automatic video summarization has been proposed to extract a compact representation of video data. Although automatic video summarization can effectively reduce the video content while maintaining the most important information, it is still time-consuming for users to navigate and/or to search through a summarized video.

Moreover, automatic video captioning can generate semantic descriptions to a video segment, which can then be used for indexing and facilitating review. However, current state-of-the-art methods for video captioning may not be suitable for dealing with long video sequences because they focus on topic-coherent videos. With this approach, a caption could be very brief with poor coverage as the topics might vary throughout a long video. Simply

© 2017. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

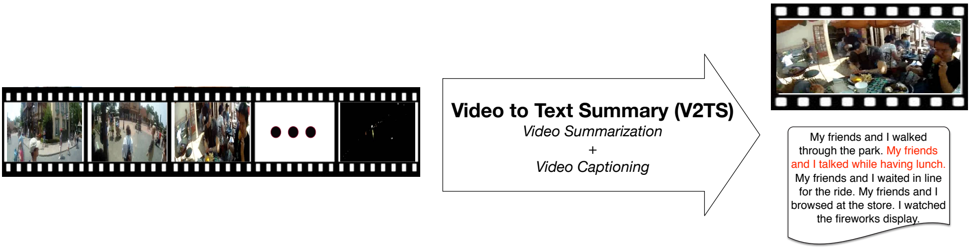
Video to Text Summary (V2TS)
Video Summarization
+
Video Captioning
My friends and I walked through the park. My friends and I talked while having lunch. My friends and I waited in line for the ride. My friends and I browsed at the store. I watched the fireworks display.

Figure 1: User-generated video data (e.g. life-logging) can be very lengthy and it is often hard to review and navigate through video. We propose a general neural network configuration called Video to Text Summary (V2TS), which combines video captioning and video summarization models into a single end-to-end recurrent network. The proposed model can generate summary videos with text descriptions at different levels of detail, which allows efficient navigation and retrieval.

applying segmentation and then captioning to long videos may result in several redundant or uninformative video captions.

To facilitate navigation of long videos, we propose a system, namely, Video to Text Summary (V2TS), to generate captions that summarize long video content as in the example shown in Figure 1. The proposed system offers a brief semantic understanding of a long video through a text summary. The details of the text summary can be adjusted by deciding how many shots are selected in a video summary. Moreover, each caption in the text summary can serve as an index to a shot in the video.

There are many works that address video summarization and captioning separately, but to the best of our knowledge, none of them consider modeling these two tasks together. However, video summarization and video captioning are complementary tasks, and a joint model could be beneficial for both. The context of the video summarization task can improve captioning because it helps the captioning model focus on the features in important frames. The context of the video captioning task can also enhance the video representation of the summarization model because the understanding of semantic video content such as objects, persons, and scenes, can help video summarization [25].

Convolution Neural Networks (CNN) combined with Recurrent Neural Networks (RNN) have shown promising performance on both video summarization [50] and video captioning [31] tasks. Taking advantage of this recent advancement, we propose a general neural network configuration to generate both video summarization and video captioning outputs. The proposed system is built upon an encoder-decoder framework that was previously used for a video captioning task [40], and is combined with an extra fully connected network for video summarization. During the training phase, the combined networks take video summary and video caption annotations as supervisory signals and update the weights in the corresponding sub-networks, respectively. By using multi-task learning, the shared RNN-based encoder is optimized over both the video summarization and video captioning tasks.

We conducted experiments on both video summarization and video captioning benchmarks and found that the proposed network indeed has better performance in both tasks compared to the models that only focus on either single task, which confirms our idea that the two tasks are highly complementary. We also conduct experiments on the VideoSET dataset [46], a dataset that contains both video summary and caption annotations; this allows us to demonstrate a novel use case of V2TS: generating a joint text and video summary of an input video, which can help users to easily navigate large amounts of video data.

Our contributions include: (1) proposing a general neural network configuration to learn a joint model that allows the context of video summarization to influence video captioning and vice versa; (2) demonstrating performance improvement of the joint model on both tasks in comparison to using the models that only focus on either aspect; (3) offering an end-to-end system to generate a compact text summary of a large amount of video data for efficient navigation. The proposed network can be potentially adapted to different types of supervisory signals to generate a video-to-text summary with selected focus. For instance, creating a text summary that features a certain person (e.g., a birthday boy) in a video.

2 Related Work

Video Summarization. Traditionally, researchers use unsupervised methods for automatic video summarization [8, 23, 24, 25, 26, 28, 49]. In these methods, researchers design criteria such as relevance, diversity, and representativeness to select important frames or shots from the video. Some researchers utilize the web media and metadata [19, 37] as prior knowledge to generate better summarization results. Visual attention [4, 10] was also used to select the important frames. However, video summarization requires a semantic understanding of the video content and is hard to model with a heuristic design. Recently, some researchers have begun to exploit supervised learning from summaries annotated by humans for video summarization [13, 16]. Zhang et al. [50] applied a Long-Short Term Memory (LSTM) model, a variant of an RNN, to model the long-term dependencies in a video, which performed well in several video summarization benchmarks. Our system also uses a supervised method with RNN for video summarization. Different from previous work, we incorporate the context of a video captioning model to learn a more semantic-driven video representation for the summarization task.

Video Captioning. Video understanding has long been an important subject in the field of computer vision. Researchers have worked on many different topics of video content analysis such as video classification, action recognition, video tagging and video captioning. For video captioning, earlier works used a template-based language model to generate natural language descriptions [14, 22, 35, 44]. Recently, with the success of image captioning using CNN and RNN [9, 11, 18, 21, 27, 41] models, many researchers have adopted similar approaches to describe video content with natural language. Donahue et al. [9] proposed a two-step approach, using Conditional Random Fields and an LSTM for generating video descriptions. Venugopalan et al. [40] proposed an end-to-end framework which they later extended [39] extend by adopting an encoder-decoder framework based on two LSTM modules, one for encoding the video into a compact representation, another for decoding the video representation into a video caption. Yao et al. [45] further extended the model by incorporating a temporal attention mechanism that learns to focus on some of the video frames in the context of text being decoded. Most recently, Pan et al. [31] and Yu el al. [47] add a hierarchical structure to either the encoder or decoder in the framework to further improve the performance. Xu et al. [43] provide a large-scale video captioning benchmark with 10,000 videos and 200,000 sentences. A comprehensive literature review on deep learning-based video captioning can be found in [42]. Our work is closely related to the encoder-decoder framework for video captioning in the above works. Note that our network differs from these works in that our model learns a video representation shared between the encoder and the video captioning components. We use multi-task learning, enabling the summarizer and encoder to jointly influence each other. Furthermore, the output importance scores are used

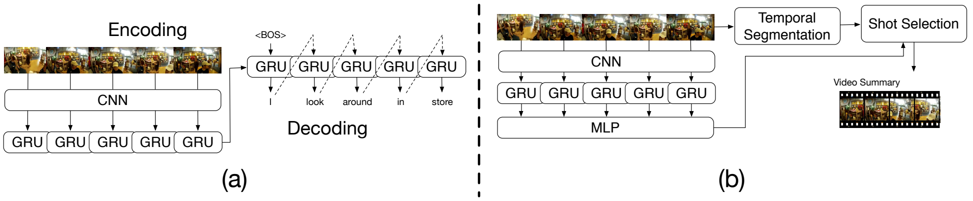

Figure 2: (a) The encoder-decoder RNN model for video captioning. (b) The RNN model for video summarization. Please see Section 3.2 and Section 3.3 for more details.

for both shot selection and to influence the caption generated for each shot. Moreover, the decoder is designed to focus on the important frames in the video, and our system differs from the temporal attention model used in [45] as our summarization weights are learned from supervised signals according to the labelers' understanding of highlights in the video content and are independent from the context of text generated during the decoding process. Recently, Ballas et al. [2] and Zanfir et al. [48] also propose different spatial-temporal video representation for video captioning by either using different layers of intermediate visual representation or attention models. Instead of learning from the context within a captioning task, our idea addresses a different point that the supervisory signals from another complementary task, summarization, can be helpful. We use multitask learning to translate the information for improving both tasks rather than either task only. The different points (context within a task and from another task) do not conflict and may be combined for further improvement.

3 Video to Text Summary (V2TS)

In this section, we first introduce an encoder-decoder framework for video captioning and video summarization separately. Then, we describe how to create the proposed joint model in our Video to Text Summary (V2TS) system. Here we adopt the CNN structure proposed in [36] (VGGNet) as one of our basic building blocks.

3.1 Recurrent Neural Network (RNN)

Here we adopt a variant of RNNs called Gated Recurrent Unit (GRU) [7], which uses following equations to generate output h_t for each time-step given input x_t and the output of the previous step h_{t-1}:

$$\begin{aligned} r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) & \hat{h}_t &= \phi(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \\ z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) & h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t, \end{aligned} \tag{1}$$

where $\sigma(\cdot)$ is sigmoid function, $\phi(\cdot)$ is a nonlinear activation function (Rectified Linear Unit) [29]; $\odot$ denotes element-wise multiplication and $W_r, W_z, W_h, U_r, U_z, U_h, b_r, b_z, b_h$ are the model parameters. Researchers have shown that GRU neural network models can model long-range dependencies within sequential data and performs well in many different tasks such as machine translation and video captioning.

3.2 RNN for Video Captioning

Figure 2 (a) shows the encoder-decoder framework for video captioning, where each video is annotated with a caption in the training phase. Each frame in a video is input to the CNN which produces frame-level feature representations. The frame-level CNN features $X = (x_1, x_2, ..., x_n)$ are fed into an encoder RNN described in section 3.1 to generate a video representation. The video representation is then used as the initial hidden state in the RNN decoder. During the decoding process, each generated word is represented as a one-hot vector $S = (s_1, s_2, ..., s_n)$, which first goes through an embedding layer to generate a word embedding, and then is fed into the decoder RNN. The outputs of the decoder RNN are then fed into an output projection and a softmax function to generate the probability distribution of the word at each time-step. During the training phase, we minimize the cross-entropy loss function respect to all model parameters below using Adam optimizer [20]:

$$L_c(X,S) = -\sum_{t=1}^{n} \log p_t(s_t), \tag{2}$$

where $p_t(s_t)$ is the probability of the correct word being output at time-step t. During the test phase, the RNN decoder initializes the first word with a special symbol <BOS> and uses word s_{t-1} as input to generate word s_t. For each time-step, the word with the maximum probability is selected.

3.3 RNN for Video Summarization

Figure 2 (b) shows the RNN-based system for video summarization, where each video frame is labeled with an importance score at the training stage. Similar to a video captioning system, each frame is input to a CNN that generates a frame-level feature representation. Frame-level CNN features are then fed into the RNN (cf. section 3.1) to sequentially generate an embedding at each time-step, followed by a multi-layer perceptron (MLP) to generate importance scores. During the training phase, the mean square loss between the generated score and the human annotation $K = (k_1, k_2, ..., k_n)$ is minimized using Adam optimizer:

$$L_s(X,K) = \sum_{t=1}^{n} ||k_t - f(x_t)||^2, \tag{3}$$

where $f(x_t)$ is the importance score of time-step t. To get a shot-level video summary at the test stage, a given test video is first segmented into small shots using the Kernel Temporal Segmentation (KTS) algorithm [34]. Shots are selected as a video summary according to the knapsack problem as in [37]:

$$maximize_{u_1,u_2,...u_n} \sum_{i=1}^{n} u_i v_i \quad s.t \sum_{i=1}^{n} u_i w_i \leq l, u_i \in \{0,1\}, \tag{4}$$

where v_i is a summation of frame-level importance scores of the i_{th} shot, w_i is the length of the i_{th} shot and l is the target length of the summarized video, u_i are the optimized binary variables that indicate whether i_{th} shot should be selected or not. Note that the above problem can be efficiently solved with dynamic programming, and the summary is a combination of selected shots (i.e. $u_i = 1$) in temporal order.

3.4 Joint Video Summarization and Captioning (V2TS)

The proposed V2TS system is shown in Figure 3. To bring the context of video summarization to captioning and vice versa, the two tasks share the same video representation network that is optimized by multi-task learning. Each frame in a video again undergoes CNN and RNN encoding to generate a video representation. However, instead of directly feeding the last output of the RNN encoder into a RNN decoder, V2TS is integrated with an MLP described in section 3.3 to obtain frame-wise importance scores that are then normalized by

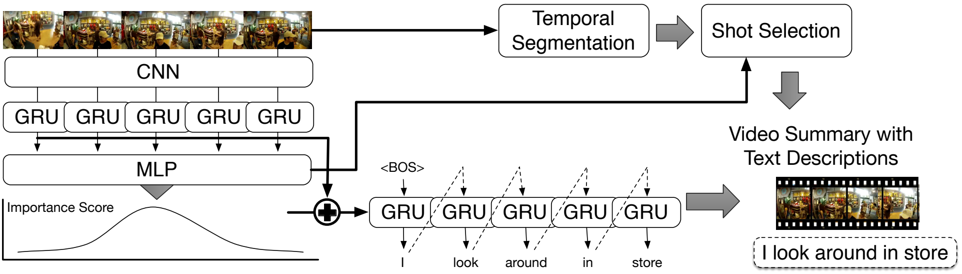

Figure 3: The proposed Video to Text Summary (V2TS) joint model. Please see Section 3.4 for more details.

a softmax function. The output features of the encoder RNN are weighted by normalized importance scores before entering the decoder RNN:

$$h_0^{(decoder)} = \sum_{i=1}^{n} \left(\frac{\exp(f(x_i))}{\sum_{j=1}^{n} \exp(f(x_j))} \right) h_i^{(encoder)}. \tag{5}$$

During the training phase, the following loss function is used to optimize both importance score generation and caption generation:

$$L(X,S,K) = \mathbf{1}[S \neq \emptyset] L_c(X,S) + \lambda \mathbf{1}[K \neq \emptyset] L_s(X,K), \tag{6}$$

where λ is a hyper-parameter that adjusts the weight between caption loss and summary loss. In addition to updating the shared CNN and RNN encoder, the indicator function $\mathbf{1}[\cdot]$ decides either the MLP for importance score generation or the RNN decoder for caption generation needs to be updated based on the input label type. During the test phase, the trained model generates both importance scores for video summarization (cf. shot selection in Section 3.4) and captions related to the summarized video. Note that the encoder outputs are weighted by the importance scores when generating a caption; therefore, the caption is likely to better reflect the summarized video. In the meantime, the model can learn a more semantic video representation through back propagation of caption labels and the shared representation also benefits the computation of importance scores for summarization.

4 Experimental Results

4.1 Datasets

To evaluate video captioning performance, we conduct experiments on two different datasets: Microsoft Video Description Corpus (MSVD) [5] and MSR - Video to Text Dataset (MSR-VTT) [43]. For evaluating video summarization performance, we use the TVSum Dataset [37]. Finally, we use the VideoSET dataset [46] to evaluate the proposed method in the context of video to text summary. Following are the detail descriptions of the datasets:

Microsoft Video Description Corpus (MSVD) The MSVD dataset contains 1,970 video clips downloaded from YouTube, and the average duration of the video clips is about 9 seconds. Each video clip is annotated with multiple short descriptions in multiple languages by human annotators. This dataset is widely used for evaluation on Automatic Video Captioning [31, 39, 40, 45, 47]. Following previous works, we only use the English descriptions (around 80,000) and split the dataset into 1,197, 100 and 670 clips for training, validation and testing, respectively.

MSR - Video to Text Dataset (MSR-VTT) The MSR-VTT dataset contains 10,000 video clips downloaded from YouTube across 20 different categories, and the average duration of the video clips is about 14 seconds. The dataset also contains 200,000 short descriptions along with the video clips, and each clip is paired with around 20 descriptions. We follow the protocol of the dataset and use 6,513, 497, and 2,990 clips for training, validation, and testing, respectively. Note that in both the MSVD and MSR-VTT datasets, multiple descriptions of the same video clips are annotated independently by different annotators.

TVSum 50 Dataset (TVSum50) TVSum contains 50 videos downloaded from YouTube across 10 categories, and the average duration of the video clips is around 4 minutes. The dataset contains frame-level importance scores on a scale from one to five labeled by human annotators. Each video is labeled by 20 different people so there are a total of 1,000 annotations. We follow [50] and use 40 videos with 800 annotations as training data and the rest for testing.

VideoSET Dataset (VideoSET) VideoSET provides text summaries and shot-based text descriptions for 11 videos. The duration of the videos range from 45 minutes up to 8 hours and the total length is around 40 hours. The videos are from three different sources: (1) Daily life egocentric dataset [23], (2) Disneyworld egocentric dataset [12], and (3) TV episodes from "Castle", "The Mentalist", and "Numb3rs".

4.2 Evaluation Metrics

For video captioning, we employ two different metrics: BLEU [33] and METEOR [3]. These two metrics are widely used in the machine translation as well as video captioning works and have been shown to have good correlation with human judgments. Following previous works on video captioning, we used the scripts provided by the authors of [6]. The detail computation of BLEU and METEOR can be found in [33] and [3] respectively.

For video summarization, we follow the protocol in [15, 16, 37, 50] and constrain the length of the generated video summary to be less than 15% of the original video (i.e. $l = 0.15$ in Equation 4). We then compute the F-Score against the annotated summary for evaluation. For video to text summarization, we evaluate against the ground-truth text descriptions using BLUE and METEOR.

4.3 Baselines

For video captioning, we compare our model with the following previous works: **Mean Pooling + LSTM (MP-LSTM)** [40]: Mean pooling is applied to the CNN features and then the features are fed into an LSTM language model to generate video descriptions. **LSTM + Temporal Attention (LSTM-SA)** [45]: A temporal attention mechanism is applied to the LSTM language model to generate video descriptions that focus on different temporal locations along the video. **Sequence to Sequence LSTM (S2VT)** [39]: The LSTM encoder-decoder framework. The CNN features are first fed into an encoder LSTM to generate a compact video representation. The representation is then fed into a decoder LSTM language model to generate video descriptions. **LSTM embedding (LSTM-E)** [32]: It projects the visual features and text features into a common space and uses an LSTM language model to generate video descriptions. **Paragraph RNN decoder (P-RNN)** [47]: The encoder-decoder framework with a hierarchical decoder module that generates paragraphs with multiple video sequences. **Hierarchical Recurrent Neural Encoder (HRNE)** [31]: The encoder-decoder framework with a hierarchical encoder module. **Hierarchical Recurrent Neural Encoder**

Method	BLEU@4	METEOR
MP-LSTM (Alex)	31.2	26.9
LSTM-SA (Google)	40.3	29.0
S2VT (VGG)	40.6	29.2
LSTM-E (VGG)	40.2	29.5
P-RNN (VGG)	44.3	31.1
HRNE (Google)	43.6	32.1
HRNE+A (Google)	43.8	**33.1**
V2TS (VGG) (Ours)	**45.1**	31.6

Method	BLEU@4	METEOR
MP-LSTM (VGG)	34.8	24.8
MP-LSTM (VGG + C3D)	35.8	25.3
LSTM-SA (VGG)	35.6	25.4
LSTM-SA (VGG + C3D)	36.6	25.9
V2TS (VGG) (Ours)	**37.5**	**25.9**

Table 1: (Left) Experimental results on MSVD dataset. (Right) Experimental results on MSR-VTT dataset.

with Attention (HRNE+A)[31]: HRNE with extra attention modules on the encoder inputs and decoder inputs.

For video summarization, we compared our model with the following works: **Title-based Video Summarization (TVSum)** [37]: Unsupervised video summarization framework that utilizes web images and video titles to find visually important shots. **Video Summarization with LSTM (vsLSTM)** [50]: Supervised learning method that uses a bi-directional LSTM to learn important shots. **Determinantal Point Process with LSTM (dppLSTM)** [50]: Supervised learning method that combines an LSTM with the determinantal point process to select important shots with high diversity. Trained with augmented training data from three other datasets.

For video to text summary, we show qualitative results of our model applied on VideoSET data and generating a text summary for a life-logging video.

4.4 Implementation Details

We use TensorFlow [1] to build our network. For every input video, we down-sample the frame to 3 fps for efficient training. For video captioning, we truncate each input sequence at 32 frames (which represents around 11 seconds) and pad sequences of less than 32 frames with zeros. For video summarization, we use KTS to detect shots in the video and limited the maximum length of each shot to 32 frames. For each frame, we extract a 4096-dimensional feature vector from the fc7 layer of the VGGNet (The weights of CNN parameters are all fixed). Following previous work, the embedding dimension, hidden dimension, and output dimension are all empirically set to 512. We use the 20,000 most frequent words in the training set as our vocabulary. The λ of the joint model (cf. Equation 6) is empirically set to 1. During the training process, the Adam optimizer with default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$) are used for optimize the network parameters and the initial learning rate is set to 0.0002 as in [31]. The batch size is set to 50 for all experiments. To prevent overfitting, the dropout rate is set to 0.5 for the CNN feature and word embeddings. We also add L2 regularization to further prevent over-fitting during training of the MSVD captioning model.

4.5 Results and Discussion

Experiments on Video Captioning. Table 1 (left) shows video captioning results on the MSVD dataset[1]. Our work outperforms all other methods in terms of BLEU and is compet-

[1] For fair comparison, we only compare results that uses static frame-level features. Note that our goal is to demonstrate the effectiveness of the summary signal in helping with captioning model and we can further improve the performance by extending our model with a more complicated encoder/decoder structure or with multiple feature streams (e.g. C3D temporal feature [38])

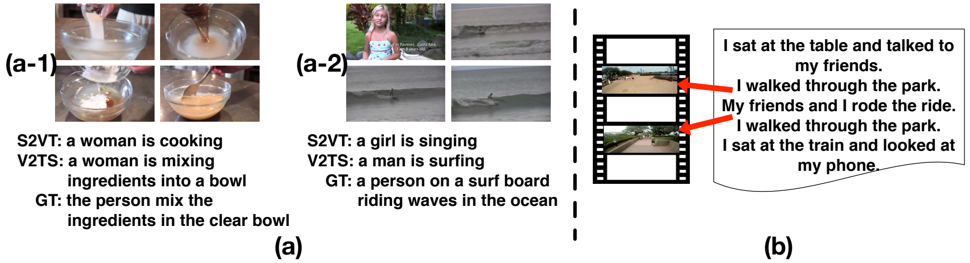
(a-1)
S2VT: a woman is cooking
V2TS: a woman is mixing ingredients into a bowl
GT: the person mix the ingredients in the clear bowl
(a-2)
S2VT: a girl is singing
V2TS: a man is surfing
GT: a person on a surf board riding waves in the ocean
(a)
I sat at the table and talked to my friends.
I walked through the park.
My friends and I rode the ride.
I walked through the park.
I sat at the train and looked at my phone.
(b)

Figure 4: (a) Examples of video captioning on MSR-VTT. (b) The text summary generated by our V2TS model.

itive with all other methods in terms of METEOR. MP-LSTM has the worst performance, mainly because it ignores the temporal information in the video, and its features (AlexNet) have less discriminative power compared to other more complicated networks. Our work is most similar to S2VT since we both utilize the same encoder-decoder structure. However, our works outperform S2VT because we utilize the video summarization signals to train our model to focus on the important frames in a video. P-RNN and HRNE employ a more complicated hierarchical recurrent network structure, but our model still has a competitive performance against theirs.

Table 1 (right) shows video captioning results on a different dataset (MSR-VTT)[2]. We compare our results with the ones reported in [43] and again show our model has competitive performance compared to state-of-the-art methods. Our model has better BLEU even when compared to the ones using multiple feature streams (i.e. VGG+C3D), which indicates our model can generalize well to different datasets. Note that our approach outperformed the LSTM model with attention mechanism (LSTM-SA) in both the MSVD and MSR-VTT datasets, which indicates that summarization weights learned from supervised signals can provide better guidance for the decoding process with our network configuration.

Figure 4 (a) shows example video captioning results on the MSR-VTT dataset. S2VT is from a simple encoder-decoder framework; V2TS is our Video to Text Summary Model, and GT denotes the ground truth caption. Figure 4 (a-1) shows our model can capture the highlight of the video while S2VT tends to focus on the whole video content. Figure 4 (a-2) is a video with someone surfing. However, the temporal segmentation of the video includes a short segment of a girl at the beginning of the video and it causes the S2VT model generate a caption of “a girl is singing.” Our V2TS model learns to focus on the later part of the video and correctly caption it as “a man is surfing.”

Note that the MSVD and MSR-VTT datasets do not provide summarization labels, therefore, we first pre-trained our network using the TVSum50 dataset to initialize the weights of our joint model so that the model can learn to capture important frames in the video.

Experiments on Video Summarization. Table 2 shows experimental results of video summarization on the TVSum50 dataset. Our model can achieve better performance in comparison to the RNN models only supervised by summary signals. This again shows that the tasks are complementary and video captions can indeed help the model to learn a more semantic video representation for better video summarization. This is aligned with the findings in the unsupervised video summarization work [30] that also utilize captions to learn semantic

[2] Recently, better results have been achieved in ACM Multimedia Grand Challenge as shown in paper [17]. However, they are using multi-modality features including video, audio, speech, and metadata. Therefore, the result is not directly comparable with our method.

Method	TVSum	vsLSTM	dppLSTM	V2TS (Ours)
F-score	50.0	54.2	59.6	**62.1**

Table 2: Experimental results on TVSum50 dataset.

features. Resembling the case of video captioning, the TVSum50 dataset does not provide captions, therefore, we pre-trained our joint model with the MSVD dataset.

Note that to the best of our knowledge, no existing works generate both video summaries and captions using a joint model. That is why we compare our method with video summarization and short-video captioning works separately. Our method does not focus on a more powerful network for either individual task, but aims to identify gains from combining the two supervisory signals and to handle long videos, which has not been addressed in existing works. The experiments on long videos are presented in the next section.

Experiments on Video to Text Summary. Figure 4 (b) shows an example text summary generated by our model with the VideoSET dataset. We use the first day and the second day of the Disneyworld egocentric video in the VideoSET dataset as the training data and evaluate the trained model on the third day of the Disneyworld egocentric video. The test data is about 8 hours long. We follow the protocol in [46] to generate a caption for every 5 seconds and generate a two-minute video summary using Equation 4. The captions of the selected shots are composed into a text summary and compared with the human annotations. The BLEU and METEOR scores of the generated text summary are 33.7 and 22.7 respectively.

Different from topic-coherent captions in short videos (cf. Figure 4 (a)), the generated text summary comprises multiple events across a long video (e.g., “talked to my friends”, “rode the ride”), which are important events appearing in the life-logging video (cf. Figure 4 (b)). our model is able to generate a summary that contains similar events (e.g., walked through the park) that happen at different times of the day, which is really important as human annotations also show similar behavior. Reporting similar events might be necessary in some applications; for example, monitoring the abnormal frequency of events. For applications that would avoid repeated events, the redundancy may be reduced by considering diversity in the process of shot selection. A baseline system composed of a video summarization model followed by a video captioning model would be expected to perform worse than our joint model since the two individual models do not perform better than the corresponding parts of our joint model (Section 3.2 and 3.3).

5 Conclusion

We propose a general neural network configuration called V2TS that exploits the complementary tasks of automatic video captioning and automatic video summarization. The proposed joint model allows the context of video summarization to influence video captioning and vice versa by using multi-task learning. Our experiments show that our V2TS model performs at least as well as the state-of-the-art methods on several video summarization and video captioning benchmarks. The captioned video summary generated by our V2TS model can help users to easily review and navigate through long videos. Currently, our model only employs a simple encoder-decoder RNN; more complicated structures such as hierarchical RNN or bi-directional RNN can be applied to further improve the performance. As future work, we would also like to investigate the effect of incorporating different supervisory signals such as location or people to generate summaries with different focuses.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/.

[2] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *International Conference of Learning Representations*, 2016.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[4] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2013.

[5] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011.

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[8] Sandra Eliza Fontes De Avila, Ana Paula Brandão Lopes, Antonio da Luz, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 2011.

[9] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[10] Naveed Ejaz, Irfan Mehmood, and Sung Wook Baik. Efficient visual attention based framework for extracting key frames from videos. *Signal Processing: Image Communication*, 2013.

[11] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015.

[12] Aliircza Fathi, Jessica K Hodgins, and James M Rehg. Social interactions: A first-person perspective. In *CVPR*, 2012.

[13] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2014.

[14] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[15] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[16] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *CVPR*, 2015.

[17] Qin Jin, Jia Chen, Shizhe Chen, Yifan Xiong, and Alexander Hauptmann. Describing videos using multi-modal fusion. In *ACM Multimedia*, 2016.

[18] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[19] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013.

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[21] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[22] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *ICJV*, 2002.

[23] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.

[24] Tiecheng Liu and John R Kender. Optimization algorithms for the selection of key frame sequences of variable length. In *ECCV*, 2002.

[25] Zheng Lu and Kristen Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013.

[26] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. A user attention model for video summarization. In *ACM Multimedia*, 2002.

[27] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *ICLR*, 2015.

[28] Padmavathi Mundur, Yong Rao, and Yelena Yesha. Keyframe-based video summarization using delaunay clustering. *International Journal on Digital Libraries*, 2006.

[29] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[30] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. *arXiv preprint arXiv:1609.08758*, 2016.

[31] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, 2016.

[32] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[34] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *ECCV*, 2014.

[35] Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. Coherent multi-sentence video description with variable level of detail. In *GCPR*, 2014.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[37] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015.

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[39] Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015.

[40] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT*, 2015.

[41] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[42] Zuxuan Wu, Ting Yao, Yanwei Fu, and Yu-Gang Jiang. Deep learning for video classification and captioning. *arXiv preprint arXiv:1609.06782*, 2016.

[43] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[44] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015.

[45] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[46] Serena Yeung, Alireza Fathi, and Li Fei-Fei. Videoset: Video summary evaluation through text. *arXiv preprint arXiv:1406.5824*, 2014.

[47] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.

[48] Mihai Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Spatio-temporal attention models for grounded video captioning. In *Asian Conference on Computer Vision*, 2016.

[49] Hong Jiang Zhang, Jianhua Wu, Di Zhong, and Stephen W Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 1997.

[50] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.