Describing Video With Attention-Based Bidirectional LSTM

Yi Bin, Yang Yang, *Member, IEEE*, Fumin Shen, Ning Xie, Heng Tao Shen, *Senior Member, IEEE*, and Xuelong Li, *Fellow, IEEE*

***Abstract*—Video captioning has been attracting broad research attention in the multimedia community. However, most existing approaches heavily rely on static visual information or partially capture the local temporal knowledge (e.g., within 16 frames), thus hardly describing motions accurately from a global view. In this paper, we propose a novel video captioning framework, which integrates bidirectional long-short term memory (BiLSTM) and a soft attention mechanism to generate better global representations for videos as well as enhance the recognition of lasting motions in videos. To generate video captions, we exploit another long-short term memory as a decoder to fully explore global contextual information. The benefits of our proposed method are two fold: 1) the BiLSTM structure comprehensively preserves global temporal and visual information and 2) the soft attention mechanism enables a language decoder to recognize and focus on principle targets from the complex content. We verify the effectiveness of our proposed video captioning framework on two widely used benchmarks, that is, microsoft video description corpus and MSR-video to text, and the experimental results demonstrate the superiority of the proposed approach compared to several state-of-the-art methods.**

***Index Terms*—Bidirectional long-short term memory (BiLSTM), temporal attention, video captioning.**

I. Introduction

WITH the proliferation of massive visual content, translating video clips to natural language has been attracting more and more research attention in the vision and multimedia community [1]–[9]. However, the semantic gap between visual content and natural language remains an open problem that obstructs the development of visual content understanding. Similar to image captioning, existing video captioning work falls into two general paradigms: 1) bottom-up paradigm that extracts semantic facets from videos and converts them into corresponding words, and then combines words to generate sentences following several sentence rules [1], [3], [10] and 2) the top-down paradigm comes up with the global representation of video clips and learns to translate to the natural language sentence directly [7], [8], [11], [12].

In the early stage of visual content description, researchers extract subject-verb-object (SVO) triplets from training sentences and match objects from visual data, then generate sentences with a given template during the test phase [1], [3]. Such a bottom-up approach, however, not only suffers from the scarcity of diversity in generated sentences, but also highly depends on the syntactical structures of templates. Furthermore, the description may also be inappropriately generated due to limited grammatical collections. For example, *a man is performing a stage* should be corrected to *a man is performing on a stage*. More important, SVO triplets representations fail to express temporal information (e.g., consecutive actions) from videos. Inspired by the great success of the neural translation machine [13], the other one, top-down paradigm [7], [8], [14], [15] employs an end-to-end structure to learn comprehensive video representation encoder and sentence decoder simultaneously, which makes the video representation more compatible with a language generator than individual concepts derived from video clips. The encoder in top-down workflow, usually applies convolutional neural networks (CNNs) [16] to extract static image features frame-by-frame, and fuses the features of all frames to one global representation for the video with various kinds of operation [e.g., pooling, recurrent neural networks (RNNs), and 3-D CNNs]. RNNs are then employed to translate video representation to sentence word-by-word.

Fig. 1 visually summarizes a general video captioning process. For a given video clip, most existing captioning systems first extract visual features frame-by-frame, which are then jointly manipulated to obtain video representation (e.g., sequence modeling or pooling). A language generator takes over the obtained video representation and translates to a natural language sentence. Ideally, the generated sentences should comprehensively capture salient aspects of the video as well as exploit the relationship between visual entities. Moreover, different from image data that contains static visual information [17]–[20], videos are dynamic data that combine a sequence of frames along time. How to exploit

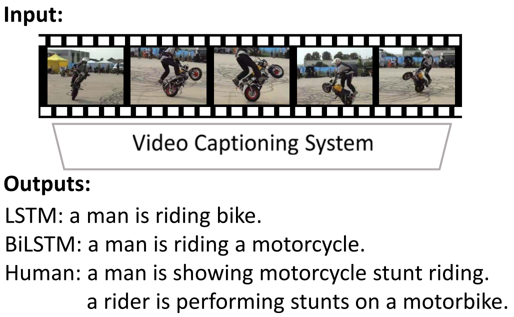
Manuscript received May 27, 2017; revised November 27, 2017 and February 27, 2018; accepted April 25, 2018. This work was supported in part by the National Natural Science Foundation of the China under Project 61572108, Project 61632007, Project 61502081, and 111 Project B17008. This paper was recommended by Associate Editor H. Yin. *(Corresponding author: Yang Yang.)*

Y. Bin, Y. Yang, F. Shen, N. Xie, and H. T. Shen are with the Center for Future Media and the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yi.bin@hotmail.com; dlyyang@gmail.com; fumin.shen@gmail.com; seanxiening@gmail.com; shenhengtao@hotmail.com).

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2831447



LSTM: a man is riding bike.
BiLSTM: a man is riding a motorcycle.
Human: a man is showing motorcycle stunt riding.
a rider is performing stunts on a motorbike.

Fig. 1. Typical process for video content description. Input a video clip and represent video by frames, then encode frames to video representation along time. We exhibit several descriptions from different approaches, where LSTM denotes a standard encoder–decoder framework, and "BiLSTM" and "Human" are descriptions generated by our bidirectional structure and human.

the spatial–temporal structure underlying in video and dynamically generate descriptive words along time is of great significance for video captioning.

To model the temporal dependencies of videos, recent work employed RNN for video content description and demonstrated the RNN with superb power for sequence exploration [7], [8], [15], [21]–[23]. RNN is a nonlinear dynamical module that maps an input sequence to an output sequence. Similar to classical neural networks, a vanilla RNN could be optimized by error back-propagation through time (BPTT). Previous research work on the video captioning pool video frames to obtain video-level representation and translate the representation to sentence using an RNN [15], [23] inevitably sacrifices the temporal structure underlying the video clip. Inspired by the superior performance of CNN in visual information analysis [16], 3-D CNN is proposed to capture the local temporal dependency in video segments with a fixed length [8]. To process video with a variable length of frames, RNNs have been applied to encode frames and integrate temporal information among locally neighboring frames [7], [14], [24]. However, existing methods cannot comprehensively exploit the long-term temporal structure, which may be helpful for exploring deeper semantics in video data.

Moreover, researchers found that the visual attention mechanism is also helpful to make the machine understand visual content in the past couple of years [25]–[30]. Specifically, the attention mechanism is first employed in image captioning by Xu *et al.* [31], which refers to the process that dynamically selects specific visual facets (e.g., main objects or regions of interest) in an image. Finally, these concepts are fused via weighted sum. This mechanism allows the model to appropriately assign the computational resource to the concepts that are more important to the visual content. Actually, the idea of an attention mechanism is analogous to the behavior of a human, which refers to the fact that one may merely focus on the most salient parts, and ignore the others.

In this paper, we propose a bidirectional long short-term memory (BiLSTM) structure, which fully explores both the forward and backward temporal information among the whole sequence of video frames. Specifically, we design a joint model by integrating a forward pass long-short term memory (LSTM), a backward pass LSTM, and CNN features to comprehensively exploit the bidirectional global temporal structure in a video clip. Furthermore, in order to make our BiLSTM focus more on semantic and relevant information, we reinforce the BiLSTM with an attention mechanism at the stage of decoding. Then, our BiLSTM could benefit from the bidirectional global representation and local focusing aspects of the video. In particular, the contributions of our BiLSTM can be summarized as follows.

1) To the best of our knowledge, our approach is one of the first to integrate bidirectional (forward and backward passes) RNNs for exploring a bidirectional global temporal structure for video captioning. Different from unidirectional approaches, our bidirectional network not only explores future fragments in videos, but also utilizes previous information.
2) Temporal soft attention in the encoder and decoder make our captioning framework focus on important video fragments. Integrated with BiLSTM, the entire system benefits from not only the comprehensive global video representation, but also local fragments that are relevant to words.
3) Extensive experiments on several real-world video captioning datasets illustrate the superiority of our proposal compared to unidirectional ones and other state-of-the-art approaches.

II. Related Work

A. Video Captioning

Video captioning integrates visual content understanding and natural language processing (NLP) techniques to generate descriptions for video clips. Promising research endeavors have been dedicated to proposing effective captioning approaches based on retrieval, grammar rules, and sequence generation. Existing video captioning approaches can be roughly divided into two categories. One category is, in the past couple of years, parsing an image to several semantic parts (i.e., subject, verb, and object) and generating sentences by predefining a sentence template with some grammar rules. Subsequently, how to align visual content with each semantic phrase and generate a comprehensive sentence became a principal problem. Krishnamoorthy *et al.* [1] applied an object detector and activity recognizer to extract SVO triplets of video clips and mined grammar dependency from an external text corpus (e.g., BNC, ukWaC, and GigaWord), and then generated a final sentence utilizing previous learned SVO triplets and sentence structure. Guadarrama *et al.* [32] constructed a hierarchical semantic tree to parse a sentence over subjects, verbs, and objects. Then, they chose the most appropriate level SVO as sentence fragments. To address the anti-commonsense problem of most SVO-based approaches, Thomason *et al.* [3] proposed a factor-graph model based on SVO methods, which integrated prior evidence in training text and posterior visual information to generate a sentence by performing a maximum *a posteriori* estimation. However, sentences generated by SVO-based models lack diversity and highly depend on the syntactical structure. This direction regards video captioning as a linguistic task more than visual content description.

Sequential modeling is the other principal direction, which learns parameters to map visual information and a textual sentence into a semantic space and explores distribution in the common space. In image captioning, researchers designed various RNN structures to model sequential dynamics of the natural language sentence [22]. According to these works, Venugopalan *et al.* [23] proposed stacking two layers of RNNs to decode video information from the video feature vector which is obtained by pooling across frames. Pan *et al.* [15] mapped a pooled video feature and sentence representation into a visual-semantic embedding space, and jointly learned the relevance and coherence, which are measured by visual-semantic similarity and cross entropy, respectively. Their joint model optimized the word generating and global relationship between sentence semantics and visual content simultaneously. Pooling operation, however, fails to explore the temporal structure underlying video snippets, and recurrently feeds video representation into RNNs, which makes the input sequence of language model with the same information at every time step, and may cause undesirable results. To address these problems, 3-D CNNs and LSTM [33], a variant of RNN, have been proposed to encode video frames to obtain video-level representation [7], [8], [15]. Inspired by the superior performance of sequence to sequence learning in machine translation [34], Venugopalan *et al.* [7] constructed an end-to-end system with two LSTMs, named S2VT, to model video clips and paired descriptions, respectively. Different from [34], the concatenated encoder and decoder, S2VT stacked two LSTM layers, where the first layer takes video frames as input sequentially and the another translates visual information to the sentence word-by-word after the whole video is read in. The authors argued that their S2VT could share parameters between the encoding and decoding phase. However, they ignored the negative influence by padding zeros to fix the length both in the encoding and decoding stage.

In recent research, the soft attention mechanism [13], [35] has achieved great success and received more research attention in the computer vision community. Xu *et al.* [31] implemented a spatial version of the soft attention model to describe image content with comprehensive sentences, which crops an image into several patches and produces words with corresponding patches. They also investigated the "stochastic hard attention" and concluded that the attention-based approach could learn comprehensive understanding of visual content as well as human intuition. Pan *et al.* [14] constructed a hierarchical recurrent neural encoder (HRNE) network for video encoding and then translated video representation to words. Their HRNE model implemented soft attention along time with a learnable sliding window filter and shared parameters among fragments as a convolutional filter of CNNs in an image. Yao *et al.* [8] devised a 3-D CNN to exploit local motion information and applied an attention mechanism to investigate global temporal dependencies among different snippets.

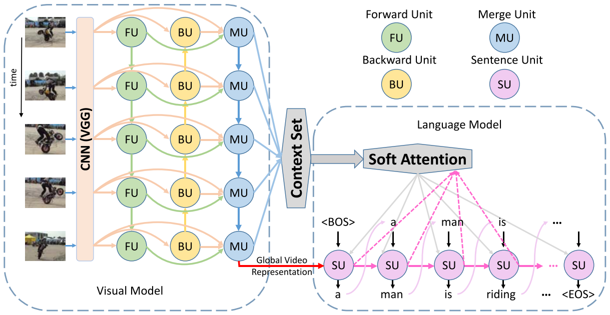


Fig. 2. Overall flowchart of the proposed video captioning framework. We first extract CNNs features of video frames and feed them into the forward pass networks (FU, green units) and the backward pass networks (BU, yellow units). We then combine the outputs of hidden states together with the original CNNs features with shortcut connection and temporal attention (details in Section III-C), and pass the integrated sequence to another LSTM (MU, blue units) to generate final video representation. We initialize the language model (SU, pink units) with global video representation and start to generate words sequentially with <BOS> token and local temporal exploration with attention, and terminate the process until the <EOS> token is emitted.

B. Bidirectional RNNs

RNNs are proposed to process sequence tasks, for example, machine translation, speech translation, and music composition [36], which stores information of previous activations by using feedback connections recurrently. Schuster and Paliwal [37] constructed a bidirectional RNN for speech recognition, which merges two separate expert networks with linear and logarithmic fusion, respectively, and comprehensively utilizes all of the input information. Later on, researchers designed a variety of bidirectional RNNs, including bidirectional LSTM, a type of improved RNN with long dependency, for sequential processing, especially in the field of NLP. Recently, Karpathy and Fei-Fei [22] also applied bidirectional RNN for image captioning. In particular, the authors employed a bidirectional RNN for word embedding and then learned the relationship between representations of images and natural-language sentences. A more similar work by Ullah *et al.* [38] devised a deep bidirectional LSTM to model a video sequence for action recognition with CNN features. From previous research, bidirectional RNNs have demonstrated better performance than the unidirectional one in sequence processing.

We will present a novel approach for translating video snippets to natural language, which employs a deep soft attention network and integrates information of both forward and backward passes. Our approach belongs to the sequential modeling of video captioning. The aforementioned, methods however, either failed to exploit the temporal structure in the video snip, or only investigated temporal information locally. Our attention-based bidirectional recurrent structure was for comprehensive and global analysis of time series, as well as local exploration. Moreover, the approach is also a sequence to a sequence model that could process the variational length sequence more flexibly.

III. Proposed Approach

In this section, we elaborate on the proposed video captioning framework, including an introduction to the overall flowchart (as illustrated in Fig. 2), a brief review of the LSTM-based sequential model and the soft attention mechanism, the joint visual modeling with bidirectional LSTM and CNNs, as well as a sentence generation process.

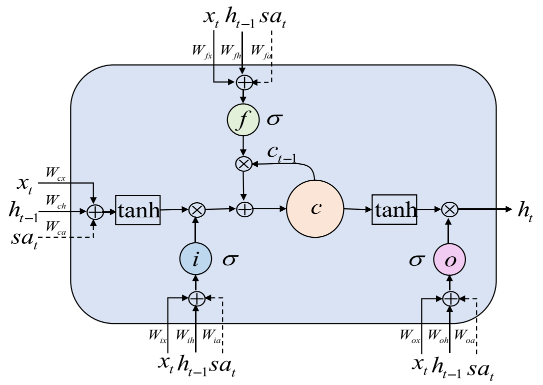


Fig. 3. Long short-term memory unit. The solid line makes up a standard unit and the dashed line with sa_t denotes integrating soft attention.

A. LSTM-Based Sequential Model

Suppose that given a video clip $V = (f_1, \ldots, f_N)$, f_t denotes the representation of the tth frame in the video. Our goal is to present the video by a feature vector V_{feature} compatibly and decode it with words step-by-step. In order to exploit the temporal dependencies of frames, it is natural to employ the recurrent structure for each frame.

RNNs have dominated the field of sequence processing since it allows us to execute variable computational steps. Vanilla RNNs map an input sequence $X = (x_1, \ldots, x_t)$ to an output sequence $Z = (z_1, \ldots, z_t)$, which can be formulated as

$$\begin{pmatrix} h_t = \Phi(W_{hx}x_t + W_{hh}h_{t-1}) \\ z_t = \Psi(W_o h_t) \end{pmatrix} \tag{1}$$

where h_{t-1} and h_t denote the hidden states of the RNN at time $t-1$ and t, respectively. Φ and Ψ are nonlinear functions, for example, the sigmoid, softmax, or hyperbolic tangent function, which are executed element-wisely. All W_{h*}-like matrices are weights that connect with previous hidden states or current input, and W_o transforms the hidden states from the hidden space to output space. All bias variables have been omitted.

However, standard RNNs severely suffer the gradient vanishing or explosion problem during BPTT. LSTM [33], an improved variant of RNN, designs a memory cell to store information and incorporates several control gates to read and write the memory cell, which has demonstrated superior performance for exploiting very long dependency in temporal sequences. Fig. 3 is a simple diagram of LSTM (without visual attention context sa_t). All of the gates are formulated as follows:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1}) \tag{2}$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1}\right) \tag{3}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1}) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \phi(W_{cx}x_t + W_{ch}h_{t-1}) \tag{5}$$

$$h_t = o_t \odot \phi(c_t) \tag{6}$$

where W_{*h}-like matrices are weight parameters that connect previous hidden states to each gate in an LSTM unit. Similarly, W_{*x} indicates connections between current input and each gate. σ and ϕ are nonlinear activation functions corresponding to the sigmoid and hyperbolic tangent function, respectively. $\odot$ represents the element-wise multiplication operation. LSTM has demonstrated superior ability for sequence processing and could capture long dependency [34]. We construct an LSTM-based network to investigate the video temporal structure for video representation, and generate video descriptions word-by-word by injecting the video-level feature to the language model as initialization.

B. Temporal Attention

Suppose we are observing a girl's face. We may just focus on her face and make details clear, but we can also notice other parts of her roughly, such as the dressing and the hair. In other words, we can "focus" our visual attention on a small area and do not lose all awareness of other visual areas. In particular, we just adjust the "weights" of our visual attention for each visual field, which assigns high weights for the focused areas. Xu *et al.* [31] divided an image into several patches and applied a spatial attentional mechanism to select patches which should be "focused" while every word is generated for image description.

The principle of the soft attention mechanism is to simulate the process of attention allocation for the field of view (FOV). It first constructs a context set $VC = (vc_1, \ldots, vc_m)$ as FOV, where vc_i denotes the ith element in visual data, for example, regions of an image or frames of a video clip. To obtain the attention vector, it dynamically generates weights α_i^t to combine each element in context set at each time step

$$sa_t = \sum_{i=1}^{m} \alpha_i^t vc_i \tag{7}$$

where α_i^t satisfies the constraint of $\sum_{i=1}^{m} \alpha_i^t = 1$ to ensure the conservation of activation. Concretely, for temporal attention in video captioning, the context set consists of all frames of a given video clip. To decide which segments are more relevant to the current step, we formulate a relevance score with the last hidden state for each element in the context set as

$$\gamma_i^t = W_{\text{rel}}\tanh(W_a vc_i + U_a h_{t-1} + b_a) \tag{8}$$

where W_a and U_a are parameters that learn to project each context element and hidden state into a latent space. W_{rel} denotes the relevance parameter. To constrain the conservation of aforementioned activation, obtained relevance scores will be normalized by

$$\alpha_i^t = \exp\left(\gamma_i^t\right) / \sum_{j=1}^{m} \exp\left(\gamma_j^t\right). \tag{9}$$

At every step, α_i^t are calculated recurrently, and assigned to each element in the context set for attention vectors by (7).

In our video captioning case, we employ temporal attention units in two stages: 1) fusion of forward and backward passes and 2) generating the sentence word by word. For each case, we construct the context set using the original CNNs feature of frames and all hidden states of the merging layer, respectively. In the language decoder, each word denotes a step in the principle derivation (7)–(9). When a word is generated,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

our temporal attention makes the language decoder focus on specific temporal locations with more semantic relevance. In particular, feeding the attention context with input simultaneously (as illustrated in Fig. 2) could be helpful when the input sequence is not aligned with the output sequence.

C. Bidirectional Video Modeling

Different from other video description approaches that represent videos by implementing pooling across frames [23] or 3-D CNNs with local temporal structure [15], we apply BiLSTM networks to exploit the bidirectional temporal structure of video clips. Specifically, we employ two LSTMs with forward and backward passes to encode frame representations separately (**FU** and **BU** layers in Fig. 2) and merge outputs of both pass step-by-step. CNNs have demonstrated promising power for facilitating image recognition; classification [16], [39]–[43]; and visual content analysis [5], [7], [44]. Therefore, we represent video frames using CNN features extracted from the *fc*7 layer, the second fully connected layer of VGG-16 layers model [45], and by the Caffe framework. Then, an N-by-4096 feature matrix is generated to denote the given video clip, where N indicates the number of frames in the video.

To obtain representation of a video snippet, Pan *et al.* [15] applied mean pooling across frames as

$$V_{\text{feature}} = \frac{1}{N}\sum_{t=1}^{N} f_t \tag{10}$$

which collapsed the temporal dependencies among frames. Venugopalan *et al.* [7] employed an LSTM to exploit the temporal structure underlying in video. At each step of encoding, the information could be expressed as

$$v_t = g(f_t \mid v_{t-1}) \tag{11}$$

where v_t denotes the encoded information at the tth time point and g indicates the encode function. However, their approaches only utilized the past information in the video.

In order to make the most use of information underlying in video clips, we propose a bidirectional recurrent structure integrating forward and backward order information for video encoding as

$$v_t = g(f_t \mid v_{t-1}, v_{t+1}). \tag{12}$$

What is interesting is that at each time point in bidirectional structure, we not only "see" the past frames, but also "peek" at the future frames. In other words, our bidirectional LSTM structure encodes video by scanning the entire video sequence several times (the same as the number of time steps at the encoding stage), and each scan is relevant to its adjacent scans.

We then apply another LSTM to generate the final feature vector of video, which combines the original CNN feature with bidirectional representation. Here, we attempt two different ways to inject the CNN feature: 1) shortcut connection described in [46]. As depicted on the left side of Fig. 4, a shortcut connection constructs a highway between layers that enables error back propaganda to be easier [46], [47] and 2) temporal attention. We regard CNNs feature of frames as a context set and dynamically select different frames depending on the current state (right side of Fig. 4).

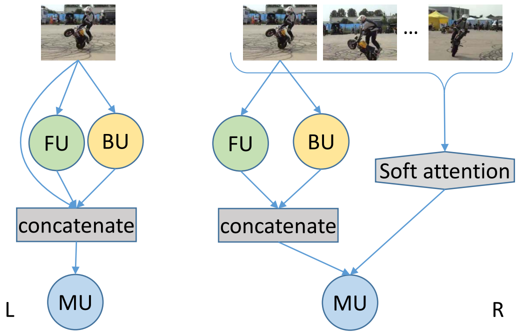


Fig. 4. Two ways for the combination of original CNN features and BiLSTM representation. The left side denotes a shortcut connection that concatenates both feature step-wisely, and the right one injects CNN features to merge units with temporal attention.

D. Generating Video Description With Attention

Most existing visual content description approaches share a common part of the visual model and language model as representation [7], [15], [22], which may fail to exploit the specific information implied in videos and sentences. Besides, Venugopalan *et al.* [23] took the same pooled visual vector of the whole video as input for every word-processing unit repeatedly. Due to the duplicate inputs at every time point, their system not only ignored the temporal structure but also might easily result in undesirable outputs. To address these issues, we first initialize the initial states of the language model with video representation, that is, the last state of video encoder, rather than default zero-initialization. From our previous research [11], such an operation connects the visual encoder and language decoder smoothly, which enables the encoder to provide more comprehensive and compatible global representation of the video for sentence generation. Then, we construct a context set using the outputs of the merging layer (**MU** in Fig. 2) that the language model pays attention to. The attention mechanism explores the local temporal relationship by weighing and summing every element which depends on the current neuron state. The language model only takes the sequence of word vectors as sole input.

Inspired by the great success of the probabilistic sequence generation machine in NLP, we generate each word by maximizing the probability at each time point. Suppose that we have a video presented by V, and the log probability of sentence S can be expressed as

$$\log p(S|V) = \sum_{t=1}^{N} \log p(w_t|V, w_1, \ldots, w_{t-1}; \theta) \tag{13}$$

where θ denotes all of the learned parameters (including weights and biases in the decoding LSTM, and the parameters in the word embedding process) in the sentence generation model. w_t and N indicate the tth word and the total number of words in the sentence, respectively. As mentioned previously, to make the video feature more compatible with the sentence generator, we jointly train the encoder and decoder by initializing decoder with video representation. Specifically, we employ an LSTM as **MU** to recurrently merge forward and backward passes together with the original CNNs feature, and inject the last hidden state and memory cell to the language decoder as a global representation of video. Considering the attention mechanism, we take all of the output states of the merging layer as the context set that our sentence generator will pay attention to. We identify the most likely sentence by maximizing the log likelihood in (13) as

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{N} \log p(S|\, V; \theta). \tag{14}$$

The optimizer updates θ using the stochastic gradient descent across the entire training process. During the training phase, the error is back propagated through time and each LSTM unit learns to derive an appropriate hidden representation h_t for the corresponding input word. We then implement the softmax function to obtain the probability distribution over all of the words in the entire vocabulary.

At the beginning of the sentence generating process, as depicted in Fig. 2, an explicit starting token (<BOS>) is needed and we terminate each sentence when the end-of-sentence token (<EOS>) is fed in. During the test phase, similar to [7], our language model takes the previous word w_{t-1} with the maximum likelihood as input at time t repeatedly until the <EOS> token is emitted.

IV. Experiments

In this section, to evaluate the performance of the proposed approach, we conduct experiments on several datasets. We first briefly introduce these datasets and detail the experimental setup, which is followed by comparing the results with several state-of-the-art methods.

A. Datasets

MSVD: The Microsoft research video description (MSVD) [48] corpus is the first open-domain video description dataset, which contains 1970 video snippets and more than 80 000 corresponding sentences. Each video clip depicts a single action or a simple event (e.g., “shooting,” “cutting,” “playing the piano,” and “cooking”) with the duration of 8–25 s. There are roughly 43 available sentences per video and seven words in each sentence on average. Following previous works [7], [8], [15], [23], we split the entire dataset into training, validation, and a test set with 1200, 100, and 670 snippets, respectively.

MSR-VTT 10K: Compared to the image description task on MSCOCO [49], video captioning lacks a very large-scale benchmark dataset. Recently, Xu *et al.* [9] released MSR-VTT 10K, a subset of the MSR-video to text project, with a collection of 10 000 video clips and 20 descriptions for each clip. The length of each video clip is between 10 and 30 s, and the average number of words in each sentence is around 10. More than 29 000 unique words are contained in the descriptions of the entire dataset. Furthermore, to help researchers devise novel approaches for video captioning, the dataset also provides comprehensive categories for videos, such as “music,” “TV shows,” and “travel.”

B. Experimental Setup

1) Preprocessing (Video Preprocessing): Following previous video captioning work [7], [8], [14], [15], [23], in order to reduce computational cost, we sample once every ten frames for video snippets in MSVD, and then represent videos with the corresponding frames. After downsampling, we have 28.5 frames for each clip on average. For MSR-VTT 10K, we set the sampling interval to 30 frames similar to [50], which takes the first place in MSR-VTT Challenge[1] of the human evaluation (containing coherence, relevance, and helpful for the blind). We extract Caffe *fc*7 layer features of video frames using the VGG-16 layers model for both datasets, and then feed the sequential features into our BiLSTM video caption generating system.

a) Description preprocessing: To clean the description corpus and prepare input for our video captioning system, we first employ *word_tokenize* operation in the NLTK toolbox[2] to obtain individual words, and then convert all of the words to lowercase. All of the punctuation and characters that cannot be converted to ASCII code are removed, and then we start each sentence with <BOS> and end with <EOS>. Finally, we obtain vocabularies with 12 708 and 29 110 unique words (both <BOS> and <EOS> are counted) for MSVD and MSR-VTT 10K, respectively. Each input word is represented by one-hot vector with the dimension of its corresponding vocabulary size.

2) Models: To comprehensively investigate the performance of the bidirectional structure, we not only implement our BiLSTM for video captioning, but also revise S2VT [7], another sequence to sequence model for video describing, to a bidirectional and reinforced version, and report the performance of different structures, respectively. For convenient comparison, we set all of the LSTMs in our experiments with 512 hidden units as [7] and [15]. During the training phase, we set the total max length of video-sentence pairs L_{total} as 80 for MSVD, and 55 for MSR-VTT 10K. From our statistics, more than 99% of sentences in MSVD contain less than 40 words, and in MSR-VTT 10K, 98.9% of sentences are less than 25 words. We also note that all of the videos in MSR-VTT 10K have a number of frames that are less than 30 (sample once every thirty frames), and Venugopalan *et al.* [7], pointed out that 94% of the YouTube training videos satisfy our maximum length limitation. To preserve sufficient visual information, we adopt two ways to truncate the videos and sentences adaptively when the sum of the number of frames and words exceeds the max length limit. We first set a conditional max length L_s for the sentence. If the number of words is smaller than L_s, we truncate the frames to satisfy the maximum length. When the length of the sentence is larger than L_s, we discard the words that exceed the length and use video frames with a maximum number of L_v, which satisfies the constraint of $L_v + L_s = L_{\text{total}}$.

a) Bidirectional S2VT: Similar to [7], we implement several S2VT-based models, that is, S2VT, bidirectional S2VT, and reinforced S2VT with the bidirectional LSTM video encoder. We conduct experiments on S2VT using our visual

[1] http://ms-multimedia-challenge.com/challenge

[2] http://www.nltk.org

TABLE I
COMPARISON RESULTS OF UNIDIRECTIONAL, BIDIRECTIONAL STRUCTURES, AND REINFORCED BiLSTM IN BOTH S2VT-BASED AND BiLSTM STRUCTURE WITH BLEU-N (N=1,2,3,4) AND METEOR (REPORTED IN PERCENTAGE, HIGHER IS BETTER). UNI, BiLSTM, AND RELSTM ARE OUR PROPOSED APPROACHES WITH UNIDIRECTIONAL, BIDIRECTIONAL, AND REINFORCED VIDEO ENCODER, RESPECTIVELY

Model	**MSVD**					**MSR-VTT 10K**				
	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	BLEU1	BLEU2	BLEU3	BLEU4	METEOR
S2VT-unidirectional	79.7	60.5	48.3	36.5	29.6	77.0	58.1	43.9	31.9	25.2
S2VT-bidirectional	79.5	60.2	48.4	36.6	29.7	77.3	58.2	44.1	32.2	25.6
S2VT-BiLSTM reinfored	**80.2**	60.5	47.7	35.3	29.9	78.5	60.0	45.7	33.6	25.9
Uni	79.5	59.5	47.3	35.6	29.6	78.2	59.6	45.3	33.2	25.7
BiLSTM	79.4	60.5	**48.6**	37.1	29.8	**79.0**	60.4	46.0	33.6	26.1
ReBiLSTM(shortcut)	79.0	**60.5**	48.4	**37.3**	**30.3**	78.9	**60.4**	**46.1**	**33.9**	**26.2**

features and LSTM structure instead of the end-to-end model in [7], which needs original RGB frames as input. For the bidirectional S2VT model, we first implement forward and backward passes for bidirectional video encoding and merge the hidden states with another LSTM layer step-by-step. Then, the language layer receives merged hidden representation with the null padded as words. In other words, we replace the video layer in vanilla S2VT with forward, backward, and merge operation for bidirectional encoding. We also pad the input of forward and backward LSTMs with zeros at the decoding stage, and integrate the hidden states as input for the merging layer rather than null padding. Finally, the merged hidden states are used to concatenate with embedded words and fed into the language layer at the decoding stage. In the last model, we regard merged bidirectional hidden states as a complementary enhancement and concatenate to the original $fc7$ features to obtain a reinforced representation of video, then derive sentences from new features using the language model. The loss is computed only at the decoding stage in all S2VT-based models.

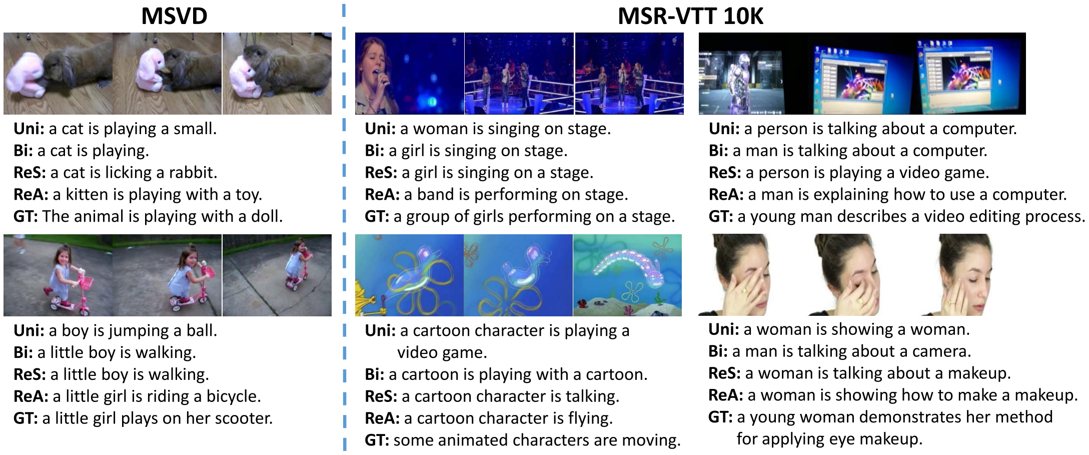
b) Our BiLSTM structure: Different from S2VT-based models, we design a bidirectional captioning model that learns to encode video frames and decode visual representation to sentences individually rather than sharing the common parameters between encoding and decoding stages. To validate the performance of the bidirectional-based structure, as in S2VT-based models, unidirectional LSTM, bidirectional LSTM, and reinforced BiLSTM are executed to investigate the performance of each structure. Specifically, our unidirectional one only applies to an LSTM layer with 512 hidden units as an encoder and an LSTM as a decoder. We use the memory cell and hidden state of the last time point of encoder to represent global video features and apply them to initialize the description generator. Then, we feed the <BOS> to invoke the decoder to caption video clip, and the previously generated word is fed into the current time step. The bidirectional structure and reinforced BiLSTM in the encoder are implemented similarly to the corresponding-type structure in S2VT-based models, respectively. Then, the video representations are fed into description generator as the aforementioned unidirectional model.

For attention-based models, we first add the temporal attention mechanism at the decoding stage, which makes the decoder generate words focusing on different video fragments with dynamical weights. To investigate different ways of merging CNN features and bidirectional representation, as depicted in Fig. 4, we inject CNN features into our bidirectional structure to reinforce our BiLSTM to ReBiLSTM by employing a shortcut connection and temporal attention. For shortcut connection, we do not apply any processing to CNN features, but concatenate them with hidden states of the bidirectional layer directly. This may alleviate the problem of the gradient vanishing problem in the deep model. However, the shortcut connection integrates multiple representations of video frames and encodes video along time, which indicates that each time step uses the same factor at the encoding stage. To enable our model to capture important time fragments for encoding video, we implement another attention unit to connect CNN and bidirectional representation as our final model. As shown on the right side of Fig. 4, the merging layer takes the concatenated hidden states of the forward and backward passes as input, and pays attention to the context set of CNN features. Then, the hidden states are treated as the decoding context set at the decoding stage. The memory cell and hidden state of the last step are used to represent the global feature of videos and initialize the language model.

TABLE II
COMPARISON OF PRACTICAL TIME CONSUMING. MEASURED BY THE AVERAGE TIME OF EACH ITERATION DURING TRAINING PHASE, WITH A MINI-BATCH OF 128 SAMPLES. THE EXPERIMENTS ARE CONDUCTED ON MSVD DATASET WITH AN NVIDIA GEFORCE GTX1080 GPU

	Unidirectional	Bidirectional	Reinforced
S2VT	13.04s	15.37s	17.97s
Ours	10.11s	12.83s	15.25s

3) Training and Test Details: We train our model with Theano [51], a Python library for fast scientific computation. One video-sentence pair is regarded as a training sample, and each sentence is fed into the decoder and the expected output is the word that delays one time step. We compute the mean error of cross entropy of each word and gradient applying back propagation through time during each iteration. We only compute error at the decoding stage. We employ adadelta [52] to optimize all of the parameters to maximize the log probability of a whole sentence in (13). The initial learning rate is set to 1.0. ρ and ϵ are 0.95 and 10^{-6}, respectively. We train our models using mini-batch with size 128. As shown in Table II, we evaluate the efficiency of our approaches in terms of practical time cost on MSVD. During the training phase, each iteration consumes 10.11 s, 12.83 s, and 15.25 s with respect to unidirectional, bidirectional, and reinforced models on an NVIDIA GeForce GTX1080 graphic card. The corresponding items of S2VT are 13.04 s, 15.37 s, and 17.97 s, which are slower than ours. Our training course stops early when the error has kept increasing over 20 iterations. At the decoding



Uni: a cat is playing a small.
Bi: a cat is playing.
ReS: a cat is licking a rabbit.
ReA: a kitten is playing with a toy.
GT: The animal is playing with a doll.

Uni: a woman is singing on stage.
Bi: a girl is singing on stage.
ReS: a girl is singing on a stage.
ReA: a band is performing on stage.
GT: a group of girls performing on a stage.

Uni: a person is talking about a computer.
Bi: a man is talking about a computer.
ReS: a person is playing a video game.
ReA: a man is explaining how to use a computer.
GT: a young man describes a video editing process.

Uni: a boy is jumping a ball.
Bi: a little boy is walking.
ReS: a little boy is walking.
ReA: a little girl is riding a bicycle.
GT: a little girl plays on her scooter.

Uni: a cartoon character is playing a video game.
Bi: a cartoon is playing with a cartoon.
ReS: a cartoon character is talking.
ReA: a cartoon character is flying.
GT: some animated characters are moving.

Uni: a woman is showing a woman.
Bi: a man is talking about a camera.
ReS: a woman is talking about a makeup.
ReA: a woman is showing how to make a makeup.
GT: a young woman demonstrates her method for applying eye makeup.

Fig. 5. Video captioning examples of our proposed bidirectional structures. MSVD and MSR-VTT 10K denote the exemplar videos from corresponding dataset. Uni and "Bi" are networks that encode videos with unidirectional and bidirectional LSTMs, respectively, and "ReS" and "ReA" indicate combining BiLSTM and CNN with shortcut connection and attention mechanism for reinforcement (All methods applied attention mechanism at decoding stage). GT is ground truth description that is randomly selected from reference set.

stage, we initialize the language model with global video representation, and then generate the sentence with <BOS> and take the output with the maximum log probability as the expected word and the input of the next step. The sentence-generating process is terminated when the <EOS> token is emitted.

C. Results and Analysis

BLEU [53], METEOR [54], ROUGE-L [55], and CIDEr [56] are the common evaluation metrics in image and video description. The first three are originally proposed to evaluate machine translation, and CIDEr is proposed to evaluate the image description with sufficient reference sentences. Most of the previous works adopt METEOR to evaluate their results because of its robust performance. In contrast to the other three metrics, METEOR can capture the semantic aspects since it identifies all possible matches by extracting the exact matcher, stem matcher, paraphrase matcher, and synonym matcher using the WordNet database, and computes sentence level similarity scores according to matcher weights. The authors of CIDEr also argue that METEOR outperforms CIDEr when the reference set is small [56]. BLEU-n matches words and computes n-gram precision between the reference sentence set and candidate sentence, which captures the lexical and textual consistency between sentences but cannot measure the semantic similarity. Therefore, to quantitatively evaluate the performance of our bidirectional approaches, we adopt METEOR as the main metric in the entire evaluation and BLEU as the auxiliary metric in the comparison of bidirectional and unidirectional structure. Because ROUGE-L and CIDEr are seldom used in literature, we simply do not consider them as our metrics.

We first compare unidirectional, bidirectional, and reinforced bidirectional structures by conducting experiments on S2VT and our BiLSTM. As shown in Table I, to comprehensively compare our BiLSTM structure with the S2VT sequential structure, we illustrate the experimental results of METEOR and BLEU-N. As we can see, in both S2VT-based and BiLSTM-based models, the bidirectional structure outperforms the unidirectional structure and shows worse performance than the reinforced ones with original CNN features. In other words, our bidirectional structure could provide more information for the language decoder since it integrates the past and future frames at each time step. We note that the BiLSTM reinforced structure gains more improvement than the unidirectional model in both S2VT-based and joint LSTMs structures, which means that combining bidirectional encoding of video representation is beneficial for exploiting the additional temporal structure in the video encoder (Fig. 2). On the structure level, Table I illustrates that our BiLSTMs-based models outperform almost all S2VT-based models correspondingly. The phenomenon demonstrates that our BiLSTMs structure benefits from encoding videos and decoding natural language separately, rather than sharing common parameters at both stages in S2VT.

Besides, Fig. 5 exhibits a few video captioning exemplar results with a comparison of our unidirectional, bidirectional, and reinforced BiLSTM structure (all methods with temporal attention at the decoding stage). From these examples, we can see that networks with a bidirectional structure are able to capture more details of video than unidirectional ones. This implies that the joint CNN feature with a bidirectional structure could enhance the encoding capacity of BiLSTM. The model that utilizes CNN feature as a context set demonstrates better performance because it dynamically allocates weights for each time fragment for each encoding step depending on the current state.

We also evaluate our BiLSTM structure by comparing it with several other state-of-the-art approaches, which exploit either local or global temporal structure. We first briefly review these methods as below.

1) *LSTM:* LSTM was introduced in [15], which extracts the video frame feature through Alexnet [16] and applies mean pooling to obtain video representation, then translates to natural language using a single LSTM

network. Here, we directly refer the results reported in [15].

2) *LSTM-E:* LSTM-E [15] is an improved LSTM network, which maps a mean pooled video feature and corresponding sentence representation into an embedding space. It formulates an objective function by integrating cross entropy of words and distance between video and sentence with a hyper-parameter, and minimizes error by optimizing model parameters and tuning the hyper-parameter.
3) *S2VT:* As mentioned before, S2VT is a sequence-to-sequence method that is proposed by Venugopalan *et al.* [7]. It stacks two LSTM layers for video encoding and sentence decoding, respectively, and pads null for the corresponding stage for video and sentence sequences. S2VT is the most likely approach with ours; we revise and implement several variants of S2VT in our experiments for evaluation.
4) *SA-LSTM (C3D):* SA-LSTM [8] integrates 3-D convolutional networks and LSTM with an attention mechanism for exploiting the temporal structure of video clips. Concretely, the authors first devise a 3-D CNN that captures spatial and local temporal information in each video segment and utilizes an attention mechanism to generate words by focusing on all of the C3D representation of this video.
5) *SA-LSTM (C3D+VGG-16):* Because there is not an experimental result of LSTM-E on MSR-VTT 10K, we utilize the best performance method reported in [9] for comparison. SA-LSTM (C3D+VGG-16) combines C3D and VGG-16 features for the sentence LSTM pays attention to while achieving 25.8 in METEOR on MSR-VTT 10K.
6) *MM-VDN:* MM-VDN [57] employs a multiscale CNN framework to capture multiple instances in the video clip, which represents objects with a different scale network.
7) *LK:* Venugopalan *et al.* [58] proposed improving a video description with linguistic knowledge. In particular, they trained the S2VT model and a language model on external textual data, then applied deep fusion to merge the hidden states of both models. Benefiting from external linguistic information, the sentences generated from LK are more semantic and fluent in grammar.

As shown in Table III, our joint BiLSTM reinforced model outperforms most baseline methods on both datasets. The result of “LSTM” in the first row is from [15]. “SA-LSTM(C3D)” denotes the best model, combining the local temporal structure using C3D and the global temporal structure utilizing temporal attention in [8]. From LSTM (line 1) and our “Uni” (line 13), our unidirectional joint LSTM shows rapid improvement in both datasets, which means that the pooling operation lost much information, especially temporal data, in obtaining video representation.

Although S2VT gains inferior performance by padding null for video and sentence at the decoding and encoding stage, respectively, the reinforced S2VT also outperforms most baselines. This indicates that combining the bidirectional representation and CNN feature of video enables the models to align all of the parts in video and generate better visual representation. We also observe that C3D with attention performs better than the standard LSTM encoder, which implies that C3D with a local temporal structure captures more motion characteristics in video clips.

TABLE III
COMPARISON OF OUR APPROACH AND SEVERAL STATE-OF-THE-ART MODELS (REPORTED IN PERCENTAGE, HIGHER IS BETTER)

Model	METEOR	
	MSVD	**MSR-VTT 10K**
LSTM	26.9	23.4
LSTM-E		
-VGG	29.5	-
-C3D	29.9	-
MM-VDN [57]	29.0	-
LK [58]	30.3	-
S2VT		
-unidirectional	29.6	25.2
-bidirectional	29.7	25.6
-reinforced	29.9	25.9
-VGG [7]	29.2	-
-VGG + Flow(Alexnet) [7]	29.8	-
SA-LSTM(C3D)	29.6	25.7
SA-LSTM(C3D+VGG)	-	25.8
Ours		
-Uni	29.6	25.7
-BiLSTM	29.8	26.1
-ReBiLSTM	30.3	26.2
with attention		
-Uni SA	30.2	25.9
-BiLSTM SA	30.5	26.2
-ReBiLSTM SA(shortcut)	30.7	26.4
-ReBiLSTM SA(attention)	**30.9**	**26.6**

In the last seven lines, we compare our standard joint BiLSTM structure with their corresponding attention models. Attention mechanism-based models outperform all standard networks, which means that models focus on the semantic aspects of video segment through the temporal attention when generating words. Specifically, we employ two methods to combine the CNN and BiLSTM structure for obtaining ReBiLSTM (see Section III-C). As shown in the last two rows in Table III, injecting the CNN with attention demonstrates much better performance than the shortcut connection one. This indicates that encoding video with BiLSTM and temporal attention captures more semantics because the BiLSTM can exploit the long dependency rather than the local relationship between nearby video fragments.

V. CONCLUSION

In this paper, we presented a novel approach to generate the natural-language description for video clips. Specifically, we applied two LSTM networks for the visual encoder and natural-language generator component. We encoded video sequences with a bidirectional LSTM network and paid attention to the original CNN features, which could effectively capture the bidirectional global temporal structure in video clips. Then, the language decoder was employed to translate video representation to sentences word by word. Experimental results on the MSVD and MSR-VTT 10K dataset demonstrated superior performance over several other state-of-the-art methods.

In the future work, we will focus on two main directions: 1) exploring object-level spatial–temporal dependency, which is analogous to human visual understanding and 2) reasoning relationships among different objects at the same time (spatial reasoning) or the same object at a different time (temporal reasoning).

Acknowledgment

The authors would like to thank the helpful suggestions from Y. Luo, Y. Xu, H. Wu, J. Sun, and H. Xu.

References

[1] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama, "Generating natural-language video descriptions using text-mined knowledge," in *Proc. AAAI Conf. Artif. Intell.*, 2013, pp. 541–547.

[2] F. Shen *et al.*, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2018.2789887.

[3] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney, "Integrating language and vision to generate natural language descriptions of videos in the wild," in *Proc. Int. Conf. Comput. Linguist.*, 2014, pp. 1218–1227.

[4] Y. Bin, Y. Yang, J. Zhou, Z. Huang, and H. T. Shen, "Adaptively attending to visual attributes and linguistic knowledge for captioning," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 1345–1353.

[5] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 2625–2634.

[6] F. Shen *et al.*, "Asymmetric binary coding for image search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2022–2032, Sep. 2017.

[7] S. Venugopalan *et al.*, "Sequence to sequence—Video to text," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4534–4542.

[8] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4507–4515.

[9] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 5288–5296.

[10] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2346–2352.

[11] Y. Bin, Y. Yang, F. Shen, X. Xu, and H. T. Shen, "Bidirectional long-short term memory for video description," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 436–440.

[12] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2045–2055, Sep. 2017.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[14] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang, "Hierarchical recurrent neural encoder for video representation with application to captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 1029–1038.

[15] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 4594–4602.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[17] Y. Yang, F. Shen, Z. Huang, H. T. Shen, and X. Li, "Discrete nonnegative spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1834–1845, Sep. 2017.

[18] X. Zhu, X. Li, S. Zhang, C. Ju, and X. Wu, "Robust joint graph sparse coding for unsupervised spectral feature selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1263–1275, Jun. 2017.

[19] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1083–1094, May 2015.

[20] Y. Yang, Z.-J. Zha, Y. Gao, X. Zhu, and T.-S. Chua, "Exploiting Web images for semantic video indexing via robust sample-specific loss," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1677–1689, Oct. 2014.

[21] J. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: Fully convolutional localization networks for dense captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 4565–4574.

[22] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 3128–3137.

[23] S. Venugopalan *et al.*, "Translating videos to natural language using deep recurrent neural networks," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2015, pp. 1494–1504.

[24] W. Zhang, B. Ma, K. Liu, and R. Huang, "Video-based pedestrian re-identification by adaptive spatio-temporal appearance model," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2042–2054, Apr. 2017.

[25] M. Jian, K.-M. Lam, J. Dong, and L. Shen, "Visual-patch-attention-aware saliency detection," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1575–1586, Aug. 2015.

[26] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 3156–3164.

[27] M. Zhang, Y. Yang, Y. Ji, N. Xie, and F. Shen, "Recurrent attention network using spatial–temporal relations for action recognition," *Signal Process.*, vol. 145, pp. 137–145, Apr. 2018.

[28] Z. Yücel *et al.*, "Joint attention by gaze interpolation and saliency," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 829–842, Jun. 2013.

[29] Y. Luo *et al.*, "Robust discrete code modeling for supervised hashing," *Pattern Recognit.*, vol. 75, pp. 128–135, Mar. 2018.

[30] M. Hu, Y. Yang, F. Shen, N. Xie, and H. T. Shen, "Hashing with angular reconstructive embeddings," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 545–555, Feb. 2018.

[31] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[32] S. Guadarrama *et al.*, "YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, 2013, pp. 2712–2719.

[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[35] Y. Xia *et al.*, "Perceptually guided photo retargeting," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 566–578, Mar. 2017.

[36] M. C. Mozer, "Induction of multiscale temporal structure," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 1992, pp. 275–282.

[37] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[38] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[39] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[40] J. Song *et al.*, "Optimized graph learning using partial tags and multiple features for image and video annotation," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 4999–5011, Nov. 2016.

[41] M. Hu *et al.*, "Robust Web image annotation via exploring multi-facet and structural knowledge," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4871–4884, Oct. 2017.

[42] Y. Yang, F. Shen, H. T. Shen, H. Li, and X. Li, "Robust discrete spectral hashing for large-scale image semantic indexing," *IEEE Trans. Big Data*, vol. 1, no. 4, pp. 162–171, Dec. 2015.

[43] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 154–162.

[44] R. Hong *et al.*, "Coherent semantic-visual indexing for large-scale image retrieval in the cloud," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4128–4138, Sep. 2017.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.

[47] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks,” in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2377–2385.

[48] D. L. Chen and W. B. Dolan, “Collecting highly parallel data for paraphrase evaluation,” in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2011, pp. 190–200.

[49] T.-Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[50] R. Shetty and J. Laaksonen, “Frame-and segment-level features and candidate pool evaluation for video caption generation,” in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 1073–1076.

[51] T. T. D. Team *et al.*, “Theano: A python framework for fast computation of mathematical expressions,” *arXiv preprint arXiv:1605.02688*, 2016.

[52] M. D. Zeiler, “Adadelta: An adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.

[53] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2002, pp. 311–318.

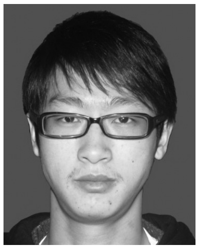
[54] M. Denkowski and A. Lavie, “Meteor universal: Language specific translation evaluation for any target language,” in *Proc. Workshop Stat. Mach. Transl.*, 2014, pp. 376–380.

[55] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Proc. Workshop Annu. Meeting Assoc. Comput. Linguist.*, vol. 8, 2004, pp. 74–81.

[56] R. Vedantam, C. L. Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 4566–4575.

[57] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, and K. Saenko, “A multi-scale multiple instance video description network,” in *Proc. Workshop IEEE Int. Conf. Comput. Vis.*, 2015.

[58] S. Venugopalan, L. A. Hendricks, R. Mooney, and K. Saenko, “Improving LSTM-based video description with linguistic knowledge mined from text,” in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, 2016, pp. 1961–1966.



Yi Bin received the B.Eng. in electronic engineering in 2013. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

His current research interests include multimedia analysis, vision understanding, and deep learning.

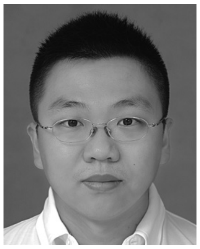
Yang Yang (M’16) received the bachelor’s degree from Jilin University, Changchun, China, in 2006, the master’s degree from Peking University, Beijing, China, in 2009, and the Ph.D. degree from the University of Queensland, Brisbane, QLD, Australia, in 2012, all in computer science, under the supervision of Profs. H. T. Shen and X. Zhou.

He is currently with the University of Electronic Science and Technology of China, Chengdu, China. He was a Research Fellow with the National University of Singapore, Singapore, from 2012 to 2014 under the supervision of Prof. T.-S. Chua. His current research interests include multimedia content analysis, computer vision, and social media analytics.

Fumin Shen received the bachelor’s degree from Shandong University, Jinan, China, in 2007, and the Ph.D. degree from the Nanjing University of Science and Technology, Nanjing, China, in 2014, all in computer science.

He is currently with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His current research interests include computer vision and machine learning.

Dr. Shen was a recipient of the Best Paper Award Honorable Mention at ACM SIGIR 2016 and ACM SIGIR 2017, and the world’s FIRST 10K Best Paper Award—Platinum Award at IEEE ICME 2017.



Ning Xie received the M.E. and Ph.D. degrees in computer science from the Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, in 2009 and 2012, respectively.

In 2012, he was appointed a Research Associate with the Tokyo Institute of Technology. In 2017, he is an Associate Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. From 2014 to 2016, he was an Assistant Professor with the School of Software Engineering, Tongji University, Shanghai, China. His current research interests include the theory and application of machine learning, computer graphics, and computer vision.

Heng Tao Shen (SM’10) received the B.Sc. (First Class Hons.) and Ph.D. degrees in computer science from the Department of Computer Science, National University of Singapore, Singapore, in 2000 and 2004, respectively.

He is currently a Professor of National “Thousand Talents Plan,” the Dean of School of Computer Science and Engineering, and the Director of Center for Future Media with the University of Electronic Science and Technology of China, Chengdu, China. He then joined the University of Queensland, Brisbane, QLD, USA, as a Lecturer, a Senior Lecturer, a Reader, and became a Professor in 2011, where he is also an Honorary Professor. He has published over 200 peer-reviewed papers, most of which appeared in top-ranked publication venues, such as *ACM Multimedia*, IEEE Conference on Computer Vision and Pattern Recognition, IEEE International Conference on Computer Vision, AAAI Conference on Artificial Intelligence, International Joint Conference on Artificial Intelligence, The ACM Special Interest Group on Management of Data, International Conference on Very Large Databases (VLDB), The annual IEEE International Conference on Data Engineering, *ACM Transactions on Information Systems*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and *VLDB Journal*. His current research interests include multimedia search, computer vision, artificial intelligence, and big data management.

Dr. Shen was a recipient of the best paper awards from international conferences, including the Best Paper Award from ACM Multimedia 2017 and Best Paper Award—Honorable Mention from ACM SIGIR 2017. He has served as a PC Co-Chair for *ACM Multimedia* 2015 and currently is an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Xuelong Li (M’02–SM’07–F’12) is a Full Professor with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi’an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi’an, China.