From Deterministic to Generative
Multimodal Stochastic RNNs for Video Captioning

Song, Jingkuan; Guo, Yuyu; Gao, Lianli; Li, Xuelong; Hanjalic, Alan; Shen, Heng Tao

DOI
10.1109/TNNLS.2018.2851077

Publication date
2018

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Neural Networks and Learning Systems

From Deterministic to Generative: Multimodal Stochastic RNNs for Video Captioning

Jingkuan Song, Yuyu Guo, Lianli Gao, Xuelong Li, *Fellow, IEEE*, Alan Hanjalic, *Fellow, IEEE*, and Heng Tao Shen

***Abstract*— Video captioning, in essential, is a complex natural process, which is affected by various uncertainties stemming from video content, subjective judgment, and so on. In this paper, we build on the recent progress in using encoder–decoder framework for video captioning and address what we find to be a critical deficiency of the existing methods that most of the decoders propagate deterministic hidden states. Such complex uncertainty cannot be modeled efficiently by the deterministic models. In this paper, we propose a generative approach, referred to as multimodal stochastic recurrent neural networks (MS-RNNs), which models the uncertainty observed in the data using latent stochastic variables. Therefore, MS-RNN can improve the performance of video captioning and generate multiple sentences to describe a video considering different random factors. Specifically, a multimodal long short-term memory (LSTM) is first proposed to interact with both visual and textual features to capture a high-level representation. Then, a backward stochastic LSTM is proposed to support uncertainty propagation by introducing latent variables. Experimental results on the challenging data sets, microsoft video description and microsoft research video-to-text, show that our proposed MS-RNN approach outperforms the state-of-the-art video captioning benchmarks.**

***Index Terms*— Recurrent neural network (RNN), uncertainty, video captioning.**

I. Introduction

IN RECENT years, various fields of computer vision have developed rapidly, including image recognition [1]–[3], facial recognition [4], [5], action recognition [6]–[8], and other tasks [9]–[11]. With the explosive growth of online videos over the past decade, video captioning has become a hot research topic. In a nutshell, video captioning is the problem of translating a video into meaningful textual sentences describing its visual content. As such, solving this problem has the potential to help various applications from video indexing and search [12]–[16] to human–robot interaction.

Building on the pioneering work of Kojima *et al.* [17], a series of studies has been conducted to come up with the first generation of video captioning systems [18]–[20]. Recently, however, the development of these systems has more and more relied on deep neural networks (DNNs) that have been proven effective in both computer vision (e.g., image classification and object detection) and natural language understanding (e.g., machine translation and language modeling), forming two technological pillars of video captioning solutions. In particular, deep convolutional neural networks (CNNs) (e.g., VggNet [21] and ResNet [22]) have been widely deployed to extract representative visual features, while recurrent neural networks (RNNs) (e.g., long short-term memory (LSTM) [23] and gate recurrent unit [24]) have been deployed to translate sequential term vectors to natural language sentences. Despite the significant conceptual and computational complexity of these DNN-based models, their effectiveness has given rise to the so-called *encoder–decoder* scheme as a popular modern approach for video captioning. In this scheme, typically a CNN is used as an encoder and an RNN as a decoder. This approach has shown better performance than the traditional video captioning methods with hand-crafted features.

Recent efforts toward developing and implementing an encoder–decoder scheme for video captioning have mainly focused on solving the following questions.

1) How to help an encode–decoder framework to more efficiently and effectively bridge the gap between video and language [25]?
2) How to facilitate video captioning using semantic information [26]?
3) How to deploy an attention mechanism to help decide what visual information to extract from video [27], [28]?
4) How to extract attributes/key concepts from sentences to enhance video captioning? [29]–[31].

Numerous approaches have been proposed to address these questions [26]–[28], [32], [33].

However, the above-mentioned approaches have been deterministic without incorporating uncertainties (i.e., both subjective judgment and model uncertainty) into the model calculations at all stages of the modeling. First, in essential,

Manuscript received July 25, 2017; revised January 16, 2018 and May 28, 2018; accepted May 31, 2018. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2014J063 and Grant ZYGX2014Z007, in part by the National Natural Science Foundation of China under Grant 61772116, Grant 61502080, Grant 61632007, Grant 61602049, and Grant 61761130079, in part by the National Key Research and Development Program of China under Grant 2018YFB1107400, and in part by the 111 Project under Grant B17008. *(Corresponding authors: Lianli Gao; Heng Tao Shen.)*

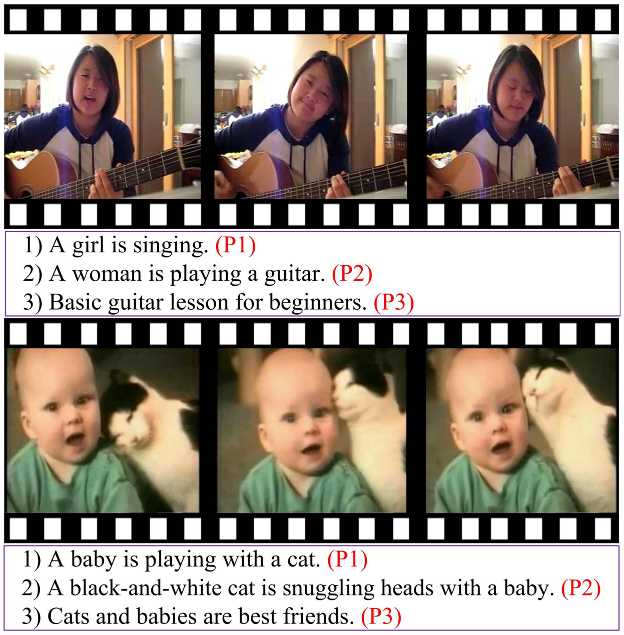
J. Song, Y. Guo, L. Gao, and H. T. Shen are with the Center of Future Media, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: lianli.gao@uestc.edu.cn; shenhengtao@hotmail.com).

X. Li is with the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China.

A. Hanjalic is with the Department of Intelligent Systems, Delft University of Technology, 2600 Delft, The Netherlands.

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/TNNLS.2018.2851077



1) A girl is singing. (P1)
2) A woman is playing a guitar. (P2)
3) Basic guitar lesson for beginners. (P3)

1) A baby is playing with a cat. (P1)
2) A black-and-white cat is snuggling heads with a baby. (P2)
3) Cats and babies are best friends. (P3)

Fig. 1. In real-life scenario, a video can be described by different sentences because the providers have different intents, experiences, and so on. However, if we use deterministic model for video captioning, only one sentence is predicted with the highest probability, which conflicts with the real scenario. By taking different hidden factors (e.g., intention and experience) into consideration, a trained model should be able to output different sentences. P1, P2, and P3 indicates three persons.

video captioning is a complex process and involves many factors, such as video itself, description intents, personal characteristics, and experiences. Except for the video content, other factors are inherently random and unpredictable. For example, in Fig. 1, we asked three people to describe two videos separately, and they provided different descriptions for each video. This indicates that video captioning is subjective and uncertain. Second, video captioning models are always abstractions of the natural video captioning processes by leaving out some less important components and keeping only relevant and prominent components, thus modeling uncertainty arises. However, both uncertainties are ignored in the previous work.

Therefore, in this paper, we are focusing on dealing with the above-mentioned uncertainties. All our attempts are to ascertain the true nature about video captioning. We propose a novel approach, namely multimodal stochastic RNN networks (MS-RNNs), which model the uncertainty observed in the data using latent stochastic variables. Our method is inspired by variational autoencoder (VAE) [34], which uses a set of latent variables to capture the latent information. This paper makes the following contributions. 1) We propose a novel end-to-end MS-RNN approach for video captioning. To the best of our knowledge, this is the first approach to video captioning that takes the uncertainty, both subjective judgment and model uncertainty, into consideration. Therefore, for each video, our model can generate multiple sentences to describe it from different aspects. 2) We propose a multimodal LSTM (M-LSTM) layer, which incorporates the features from different information sources (i.e., visual and word) into a set of higher level representation by adjusting the weights on each individual source for improving the video captioning performance. 3) We develop a novel backward stochastic LSTM (S-LSTM) mechanism to model uncertainty in a latent process through latent variables. With S-LSTM, the uncertainty is expressed in the form of probability distribution of latent variables. The uncertainty can be model into a prior distribution by making use of the consistency between prior distribution and posterior distribution. 4) The proposed model is evaluated on two challenging data sets, microsoft video description (MSVD) and Microsoft Research (MSR) video-to-text (MSR-VTT). The experimental results show that our method achieves superior performance in video captioning. Note that our model only utilizes the appearance features of videos, and no attention mechanism is incorporated.

II. Related Work

A. Recurrent Neural Networks

RNNs [35] form a directed cycle to connect units. This mechanism allows them to process arbitrary sequential data streams; thus, RNNs have been widely used in computational linguistics and achieved great success. Taking language model as an example, RNNs model a sequential data streams (e.g., a sentence) $\mathbf{s} = \{s_1, \ldots, s_T\}$ by decomposing the probability distribution over outputs

$$P(\mathbf{s}) = \prod_{t=2}^{T} P(s_t | s_{<t}) P(s_1). \tag{1}$$

At each time step, an RNN observes an element and updates its internal states, $h_t = f_\theta(h_{t-1}, s_t)$, where f is a deterministic nonlinear function and θ indicates a set of parameters. The probability distribution over s_t is parameterized as: $P(s_t|s_{<t}) = P_\theta(s_t|h_{t-1})$. The RNN language model (RNNLM) [36] parameterized the output distribution by applying a softmax function onto the previous hidden state h_{t-1}. To learn the model's parameters, RNNLM maximizes the log-likelihood by adopting the gradient descent. However, most existing RNNs models propagate deterministic hidden states.

B. Visual Captioning

The study of visual captioning problem has been going on for many years. In 2002, the video captioning system [17] was proposed for describing human behavior; the method first detects visual information (i.e., position of head, direction of head, and positions of hands) to find the position where the person is and the gesture what the person does and then selects appropriate predicate, object, and so on with domain knowledge. Finally, the method applies syntactic rules to generate a whole sentence. Following this work, a series of studies is conducted to utilize such a technique to enhance different multimedia applications [18]–[20]. And there are some works that tackle the problem with the probabilistic graphical model. Farhadi *et al.* [37] introduce the meaning space, which is represented as triplets of ⟨object; action; scene⟩ in the form of a Markov random field, and map the images and sentence to the meaning space to find the relationship between images and sentences. Rohrbach *et al.* [38] try to model the relationship between different components of the visual information with a conditional random field and then tackle the captioning problem as a machine translation problem to generate sentences.

Inspired by the recent advances in image classification using CNN networks (e.g., VggNet [21], GoogLeNet [39], and ResNet [22]), and in machine translation utilizing RNN, there have been a few attempts [26], [27], [32], [33], [40]–[42] to address video caption generation by first adopting an efficient CNN network to extract video appearance features and second utilizing an RNN to take video features and the previous predicted words to infer a new word with a softmax. In order to further improve the performance, more complex approaches [26], [27], [33] are proposed from different aspects. Specifically, Yao *et al.* [27] adopted a spatio-temporal CNN (3-D CNN) for capturing video motion information and a soft attention mechanism to select relevant frame-level features for video captioning. Pan *et al.* [26] incorporated the semantic relationship between sentence and visual content for video captioning, while Yu *et al.* [33] proposed a hierarchical framework consisting of a sentence generator to describe a specific short video internal and a paragraph generator to capture the intersentence dependence. However, all of them treat video captioning as a deterministic problem, which can only generate one output, which violate the nature of video captioning. By taking different hidden factors (e.g., intention and experience) into consideration, a trained model should be able to output different sentences. Note that the model introduced in [43] can also generate diverse sentences for image captioning, because it uses different LSTMs to generate different sentences (the number of LSTMs is equal to the number of different sentences), so their model has no uncertain factors and does not capture the uncertainty in captioning problem.

C. What Is Uncertainty

From the management point of view, uncertainty is the lack of exact knowledge, regardless of what is the cause of this deficiency [44]–[46]. Models provide us a solution to clarify our understanding of our knowledge gap, but in real life, understanding the average processes is often not sufficient and it is impossible to predict with certain results [47]. In general, besides language uncertainty, uncertainty can be classified into six major types [44], [47]: 1) measurement errors resulting from imperfections in measuring devices and observational techniques; 2) systematic error, which occurs as the results of bias in the measuring devices or the sampling process; 3) natural variation, which occurs in a system that changes, with respect to time, space, or other variations, in ways; 4) inherent randomness, which results from a system that is irreducible to a deterministic one; 5) model uncertainty, which mainly arises because the mathematical and computer models that are used for predicting future events or for answering question under specific scenarios; and 6) subjective judgment, which occurs as a result of interpretation of data. Without sufficient data, the experts' judgment will be based on observations and experience. All of these uncertainties are hidden factors affecting the results of video captioning, and we propose to model these uncertainties using latent stochastic variables.

D. Variational Autoencoder

As mentioned earlier, we know that we should find a method to capture the uncertainty in the video captioning problem. But how can we model the uncertainty? VAE [34] model gives us a good way to solve this problem. For capturing the variations in the observed variables $\mathbf{x}$, the VAE model introduces a set of latent random variables $\mathbf{z}$ and rewrites the objective function $\log P(\mathbf{x})$ as follows:

$$\log P(\mathbf{x}) \geq E_Q[\log P(\mathbf{x}|\mathbf{z})] - \mathrm{KL}[Q(\mathbf{z}|\mathbf{x})||P(\mathbf{z})] := \mathcal{L} \qquad (2)$$

where $\mathrm{KL}[Q||P]$ is the Kullback–Leibler divergence between two distributions Q and P, which measures the nonsymmetric difference between two probability distributions. And $Q(\mathbf{z}|\mathbf{x})$ is an approximate posterior distribution, which avoids to solve the intractable true posterior distribution. In [34], the VAE model was used to paint the digits, so it needs to decide not just which number is written but the angle, the stroke width, and also abstract stylistic properties, so the model uses a set of latent random variables to capture the latent information. Inspired by this, we also use latent variables with a stochastic layer to capture the uncertainty information in the video captioning. Different with painting digits, the video captioning task needs to generate different sentences based on the content of the video, so our objective function is a conditional probability and we use the loss function introduced in conditional VAE (CVAE) [48], which extend the VAE to dispose conditional probability distribution. And Krishnan *et al.* [49] compared the different variational models, and they guide us to choose an effective variational model. And there are some works that extend the VAE model to RNN [50]–[52] for generating speech or music signal. All these works inspire us to extend the captioning problem to an uncertainty problem.

III. Proposed Approach

In this section, we introduce our approach for video captioning, and we follow the conventional encoder–decoder framework. The encoder is based purely on neural networks to generate video descriptions, and the decoder, named MS-RNNs (see Fig. 2), is our major contribution. We first introduce the architecture of our proposed network and then devise the loss function and optimization.

A. Problem Formulation

Given a video $\mathbf{v}$ with N frames, we extract their frame-level features, and $\mathbf{v}$ can be represented as $\mathbf{v} = \{v_1, v_2, \ldots, v_i, \ldots, v_N\}$, where $v_i \in \mathbb{R}^{D_v \times 1}$ and D_v is the dimension of the frame-level features. For each $\mathbf{v}$, we also have a textual sentence $\mathbf{a}$ to describe it, and $\mathbf{a}$ includes T words, which can be represented as $\mathbf{a} = \{a_1, a_2, \ldots, a_t, \ldots, a_T\}$. Specifically, $a_t \in \mathbb{R}^{D_a \times 1}$ is the one-hot vector, where D_a is the

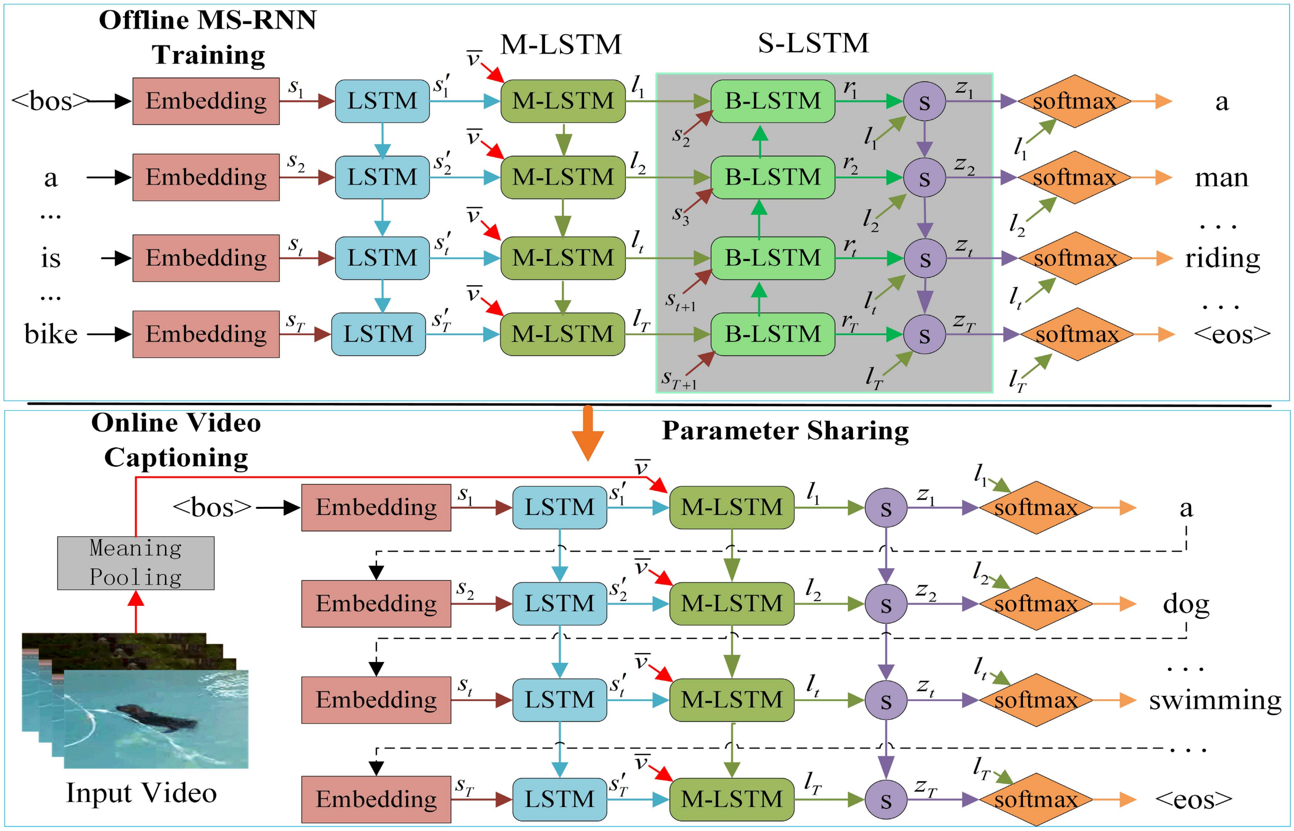


Fig. 2. End-to-end multimodal RNNs stochastic architecture for video captioning. The S-LSTM is proposed to introduce latent variables to propagate uncertainty. During the training phase, S-LSTM enables the consistency between prior distribution and posterior distribution. Therefore, during the test phase, we only need the learned prior distribution to support video caption generation. It is a common strategy in the VAE model. And we use the B-LSTM to infer the posterior distribution over latent variables, so the B-LSTM layer is removed during the test phase. Our MS-RNN model uses an embedding layer for mapping the one-hot word vectors to low-dimensional vectors (s_t) and then adds the LSTM layer to explore the temporal information between the low-dimensional vectors and generates sentence features (s_t'). We mix the visual feature ($\overline{v}$) and sentence features (s_t') with the M-LSTM layer. The latent variables (z_t) are generated by the S-LSTM layer, which includes a backward-LSTM and a stochastic layer. Finally, the word probabilities are generated by the softmax layer with latent variables (z_t) and mixed features (l_t). During the testing phase, the model generates words one by one with the beam search algorithm.

dimension of the vocabulary. Therefore, we have $\mathbf{v} \in \mathbb{R}^{D_v \times N}$ and $\mathbf{a} \in \mathbb{R}^{D_a \times T}$. Given a video, our model will predict one word at a time until we generate a textual sentence to describe the input video. In detail, in the tth time step, our model utilizes $\mathbf{v}$ and the previous words $a_{<t}$ to predict a word a_t with the maximal probability $P(a_t|a_{<t}, \mathbf{v})$ until we reach the end of the sentence. In addition, we set a mark $a_{T+1} =< eos >$ as the end of sentence.

B. Encoder

The goal of an encoder is to compute feature vectors that are compact and representative and can capture the most related visual information for the decoder. Specifically, it encodes the input $\mathbf{v}$ into a continuous representation, which may be a variable-sized set $\mathbf{v} = \{v_1, v_2, \ldots, v_i, \ldots, v_N\}$. Thanks to the rapid development of deep CNNs, which have made a great success in a large-scale image recognition task [22], object detection [53], and visual captioning [25], high-level features can be extracted from upper or intermediate layers of a deep CNN network. Therefore, a set of well-tested CNN networks, such as the ResNet-152 model [22] which has achieved the best performance in ImageNet Large-Scale Visual Recognition Challenge, can be used as candidate encoders for our framework. With a pretrained deep CNN (ResNet-152 or GoogLeNet in our experiments) on the ImageNet data set, we can apply it to each frame to extract representative frame-level features.

For encoding the sentence, because of the sparsity of one-hot vectors $\mathbf{a} = \{a_1, a_2, \ldots, a_t, \ldots, a_T\}$, like previous works [27] and [28], we process one-hot vector with an "embedding" method. We set a parameter matrix $\mathbf{U}_s \in \mathbb{R}^{D_s \times D_a}$ to map the one-hot vectors $\mathbf{a}$ to $\mathbf{s}$ as follows:

$$\mathbf{s} = \mathbf{U}_s \mathbf{a}. \tag{3}$$

The $\mathbf{s} \in \mathbb{R}^{D_s \times T}$ and $\mathbf{s} = \{s_1, s_2, \ldots, s_t, \ldots, s_T\}$ will be input to the next step. In addition, the end of sentence $a_{T+1} =< eos >$ is mapped to s_{T+1}.

C. Decoder With MS-RNN

The MS-RNN consists of three core components as shown in Fig. 2: a basic LSTM layer for extracting word-level features, an M-LSTM layer for encoding multiview information (visual and textual features) simultaneously and chronologically, and a backward S-LSTM layer to adequately introduce latent variables.

1) LSTM for Word Features: In our MS-RNN model, we use a basic LSTM layer to take $\mathbf{s} = \{s_1, s_2, \ldots, s_t, \ldots, s_T\}$ as input and output word features $\mathbf{s}' = \{s'_1, s'_2, \ldots, s'_t, \ldots, s'_T\}$ with encoded temporal information

$$s'_t = \mathrm{LSTM}\left(s_t, s'_{t-1}\right), \quad t \in \{1, 2, \ldots, T\} \tag{4}$$

where $s'_0 = \mathbf{0}$. More specifically, a standard LSTM unit consists of three gates: a "forget gate" (f_t) that decides what information we are going to throw away from an LSTM unit; an "input gate" (i_t) that decides what new information we are going to store in the cell state; and an "output gate" o_t that controls the extent to which the value in memory is used to compute the output activation of the block. A standard LSTM can be defined as

$$\begin{aligned} f_t &= \sigma\left(W_{xf} s_t + W_{hf} s'_{t-1} + b_f\right) \\ i_t &= \sigma\left(W_{xi} s_t + W_{hi} s'_{t-1} + b_i\right) \\ o_t &= \sigma\left(W_{xo} s_t + W_{ho} s'_{t-1} + b_o\right) \\ g_t &= \phi\left(W_{xg} s_t + W_{hg} s'_{t-1} + b_g\right) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ s'_t &= o_t \odot \phi(c_t) \end{aligned} \tag{5}$$

where $\sigma(\cdot)$ is a sigmoid function, $\phi(\cdot)$ denotes a hyperbolic tangent function, c_t is a cell state vector, s'_t is an output vector, g_t is a sigmoid gate, W_* is a set of parameters, $\odot$ denotes the elementwise multiplication, and b_* is a set of bias values. Then, for each word s_t, we extracted its word features as s'_t.

2) Multimodal LSTM Layer: Next, an M-LSTM layer takes $\mathbf{s}'$ and a video-level feature $\overline{v}$ as inputs to fuse a high-level features l_t

$$l_t = M\_\mathrm{LSTM}\left(s'_t, \overline{v}, l_{t-1}\right) \quad t \in \{1, 2, \ldots, T\}. \tag{6}$$

Here, instead of using advanced but complex temporal or spatial attention mechanism to select a video-level feature, we use the basic mean pooling strategy to obtain one $\overline{v}$

$$\overline{v} = \frac{1}{N} \sum_{i=1}^{N} v_i, \quad v_i \in \mathbf{v}. \tag{7}$$

The motivation is that if our model using the basic way to utilize the visual features can improve the performance of video captioning, the advantages of our MS-RNN are manifest. However, as shown in [28] and [29], the attention mechanism can further boost the performance of video captioning.

M-LSTM is a novel variant of LSTM, and it not only inherits the numerical stability of LSTM but also generates plausible features from multiview sources. We choose LSTM as our basic RNN unit due to the following reasons: 1) it achieved great success in machine translation, speech recognition, and image and video caption [25], [54], [55] and 2) compared with basic RNN units, it is absolutely capable of handing the "long-term dependences" problem.

Given two modalities $\mathbf{s}' = \{s'_1, s'_2, \ldots, s'_t, \ldots, s'_T\}$ and $\overline{v}$ as the inputs, and two initialized vectors l_0 and c_0, an M-LSTM can be used to fuse them and extract a higher level feature. An M-LSTM unit can be described as follows:

$$\begin{aligned} l_t &= M\_\mathrm{LSTM}\left(s'_t, \overline{v}, l_{t-1}\right) \\ &= \mathrm{LSTM}\left([s'_t, \overline{v}], l_{t-1}\right) \end{aligned} \tag{8}$$

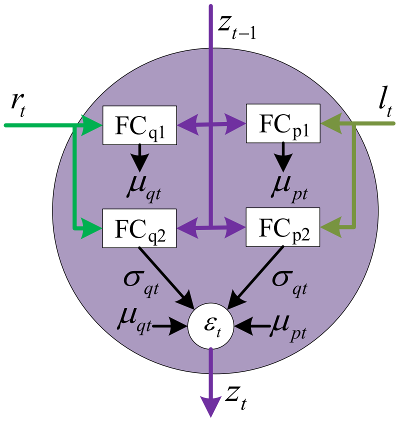

z_{t-1}
r_t
l_t
$\mathrm{FC}_{\mathrm{q1}}$
$\mathrm{FC}_{\mathrm{p1}}$
μ_{qt}
μ_{pt}
$\mathrm{FC}_{\mathrm{q2}}$
$\mathrm{FC}_{\mathrm{p2}}$
σ_{qt}
σ_{qt}
μ_{qt}
ε_t
μ_{pt}
z_t

Fig. 3. Stochastic cell of S-LSTM. The cell receives information from r_t, l_t, and z_{t-1}, uses r_t and z_{t-1} to generate μ_{q_t} and σ_{q_t}, uses l_t and z_{t-1} to generate μ_{p_t} and σ_{p_t} through fully connected layers, and finally generates z_t from μ_{q_t}, σ_{q_t}, μ_{p_t}, σ_{p_t}, and random variables ϵ_t.

where $[s'_t, \overline{v}]$ is a concatenation vector between s'_t and $\overline{v}$. To obtain an abstract concept from two modalities, the M-LSTM needs to first project s'_t and $\overline{v}$ into a common feature space, and then, the inside gates can add them together with an activation function. Then, in each time step t, we extracted a higher level feature l_t.

3) Backward Stochastic LSTM: In this section, we introduce our backward S-LSTM to take the output of M-LSTM to approximate the posterior distributions over latent variables defined as $\mathbf{z} = \{z_1, z_2, \ldots, z_T\}$, where $z_t \in \mathbb{R}^{D_z}$. The S-LSTM consists of two units: a backward LSTM unit and a stochastic unit. We define the output of the backward LSTM as r_t.

For the backward LSTM unit in time step t, its output is defined as

$$r_t = B\_\mathrm{LSTM}(s_{t+1}, l_t, r_{t+1}) \quad t \in \{1, 2, \ldots, T\} \tag{9}$$

where l_t is the output of M-LSTM at time step t, s_{t+1} is the output of embedding layer, and r_{t+1} is initialized to zero vector. The form of B_LSTM is similar to M_LSTM, but it processes sequence with backward direction. We can see that the output of backward LSTM in time step t depends on the present input l_t and s_{t+1}, and future output r_{t+1}. This is because in the stochastic units, the posterior distribution of z_t, which is calculated with (15), does not depend on the past outputs and deterministic states but depend on the present and future ones. Therefore, we propose to use the backward LSTM to extract the future information and incorporate it with a stochastic layer to achieve our goal.

Fig. 3 demonstrates the stochastic unit structure. To obtain z_t, we utilize an "reparameterization trick" introduced in [34]. This trick randomly samples a set of values $\epsilon_t \in \mathbb{R}^{D_z}$ from a standard Gaussian distribution. Therefore, $\epsilon_t \sim \mathcal{N}(0, 1)$. If we assume $z_t \sim \mathcal{N}(\mu_t, \mathrm{diag}(\sigma_t^2))$, we can use $z_t = \mu_t + \sigma_t \odot \epsilon_t$ to calculate z_t. Next, we need to solve the problem of how to learn μ_t and σ_t for z_t.

In detail, the stochastic unit takes l_t and z_{t-1} as input to approximate μ_{p_t} and σ_{p_t} by two feedforward networks (i.e., FC_{p1} and FC_{p2}). In addition, each of them contains two fully connected layers

$$\begin{aligned} \mu_{p_t} &= \mathrm{FC}_{p1}([z_{t-1}, l_t]) \\ \sigma_{p_t} &= \exp(0.5 \times \mathrm{FC}_{p2}([z_{t-1}, l_t])). \end{aligned} \tag{10}$$

$[z_{t-1}, l_t]$ is a concatenation operation. In addition, the stochastic unit also takes r_t and z_{t-1} to approximate μ_{q_t} and σ_{q_t} by two feedforward networks (i.e., FC_{q1} and FC_{q2})

$$\begin{aligned} \mu_{q_t} &= \mathrm{FC}_{q1}([z_{t-1}, r_t]) \\ \sigma_{q_t} &= \exp(0.5 \times \mathrm{FC}_{q2}([z_{t-1}, r_t])). \end{aligned} \tag{11}$$

For training, we set $z_t = \mu_{q_t} + \mu_{p_t} + \sigma_{q_t} \odot \epsilon_t$, and this method, introduced in [51], can improve the posterior approximation by using the prior mean, while for testing, we set $z_t = \mu_{p_t} + \sigma_{p_t} \odot \epsilon_t$, and we set z_0 as zero vector at the beginning. To output a symbol a_t, a probability distribution over a set of possible words is obtained using $\mathbf{U}_p$ and z_t

$$P(a_{t+1}|z_t, l_t) = \mathrm{softmax}\left(\mathbf{U}_p[z_t, l_t] + \mathbf{b}\right) \tag{12}$$

where $\mathbf{U}_p$ and $\mathbf{b}$ are the parameters to be learned. Next, we can interpret the output of the softmax layer $P(a_{t+1}|z_t, l_t)$ as a probability distribution over words.

D. Loss Function

Based on the variational inference and CVAE proposed in [48], we define the following loss function:

$$\log P(\mathbf{a}|\mathbf{l}) \geq E_Q[\log P(\mathbf{a}|\mathbf{z}, \boldsymbol{l})] - \mathrm{KL}[Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})||P(\mathbf{z}|\boldsymbol{l})] := \mathcal{L} \tag{13}$$

where $\mathcal{L}$ is the evidence lower bound of the log likelihood. The distribution $Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})$ is an approximate posterior distribution, which aims to approximate the intractable true posterior distribution. The first term $E_Q[\log P(\mathbf{a}|\mathbf{z}, \boldsymbol{l})]$, which is an expected log likelihood under $Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})$, is written as

$$\begin{aligned} &E_Q[\log P(\mathbf{a}|\mathbf{z}, \boldsymbol{l})] \\ &\quad = E_Q\left[\sum_{t=1}^{T} \log P(a_{t+1}|z_t, l_t)\right] \\ &\quad = \sum_{t=1}^{T} \log P(a_{t+1}|z_t, l_t). \end{aligned} \tag{14}$$

Here, we process the concatenation vector $[z_t, l_t]$ with a softmax layer, mentioned by (12), to approximate $P(a_{t+1}|z_t, l_t)$.

The second term $\mathrm{KL}[Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})||P(\mathbf{z}|\boldsymbol{l})]$, namely KL term, is the Kullback–Leibler divergence, which measures the nonsymmetric difference between two probability distributions (i.e., $Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})$ and $P(\mathbf{z}|\boldsymbol{l})$). And in this paper, we choose the variational model introduced in [49] to factorize the posterior distribution. The posterior and prior distributions are factorized as follows:

$$Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l}) = \prod_{t=1}^{T} Q(z_t|z_{t-1}, a_{>t}, l_{\geq t}) Q(z_0|a_{>0}, l_{\geq 0}) \tag{15}$$

$$P(\mathbf{z}|\boldsymbol{l}) = \prod_{t=1}^{T} P(z_t|z_{t-1}, l_t) P(z_0|l_0). \tag{16}$$

For approximating $Q(z_t|z_{t-1}, a_{>t}, l_{\geq t})$ and $P(z_t|z_{t-1}, l_t)$, we first use a backward LSTM layer to encode s_{t+1} [we have encoded a_{t+1} to s_{t+1} mentioned in (3)] and l_t to r_t, and then utilize the method, mentioned in Section III-C3, to approximate the means and the variances of $Q(z_t|z_{t-1}, a_{>t}, l_{\geq t})$ and $P(z_t|z_{t-1}, l_t)$. So, we can use the following function to calculate the Kullback–Leibler divergence at the tth time step:

$$\begin{aligned} &\mathrm{KL}[Q_t||P_t] \\ &\quad = \sum_{i=1}^{D_z} \log Q(z_{t_i}|z_{t-1}, a_{>t}, l_{\geq t}) \frac{P(z_{t_i}|z_{t-1}, l_t)}{Q(z_{t_i}|z_{t-1}, a_{>t}, l_{\geq t})} \\ &\quad = \sum_{i=1}^{D_z} \log \frac{\sigma_{p_{t_i}}}{\sigma_{q_{t_i}}} + \frac{\sigma_{q_{t_i}}^2 + (\mu_{q_{t_i}} - \mu_{p_{t_i}})^2}{2\sigma_{p_{t_i}}^2} - \frac{1}{2}. \end{aligned} \tag{17}$$

For the whole sentence generation, we calculate the global Kullback–Leibler divergence $\mathrm{KL}[Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})||P(\mathbf{z}|\boldsymbol{l})]$ by

$$\mathrm{KL}[Q(\mathbf{z}|\mathbf{a}, \boldsymbol{l})||P(\mathbf{z}|\boldsymbol{l})] = \sum_{t=1}^{T} \mathrm{KL}[Q_t||P_t]. \tag{18}$$

In this paper, we maximize the above-proposed loss function to learn all the parameters. More specifically, we use a backpropagation through time algorithm to compute the gradients and conduct the optimization with ADADELTA [56].

IV. Experiment

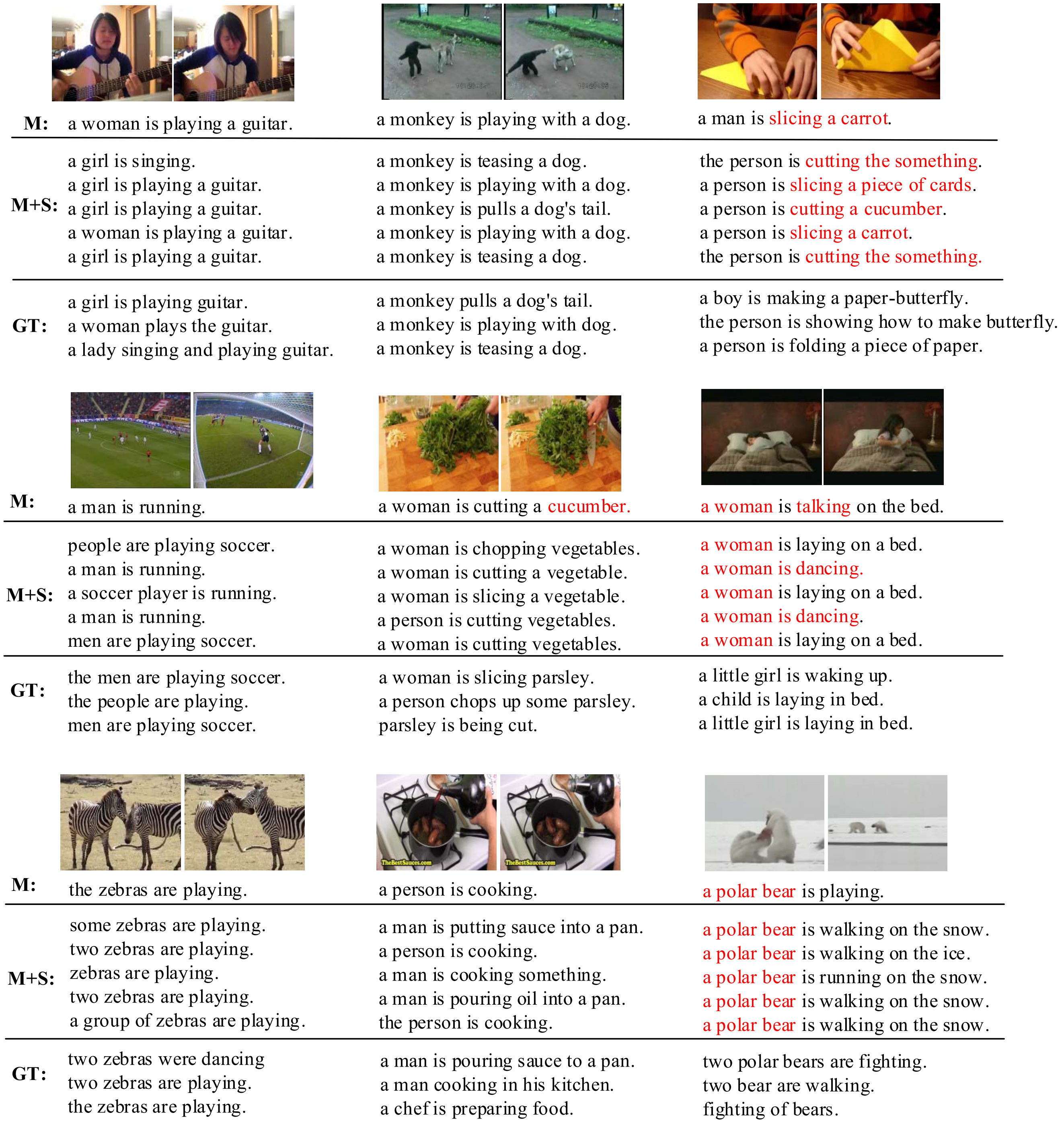
We evaluate our model on two standard video captioning benchmark data sets: the widely used MSVD [58] and the large-scale MSR-VTT [59].

MSVD: This data set consists of 1970 short video clips collected from YouTube, with an average length of about 9 s. In addition, this data set contains about 80 000 clip-description pairs labeled by Amazon Mechanical Turkers (AMT). In other words, each clip has multiple sentence descriptions. In total, all the descriptions contain nearly 16 000 unique vocabularies. Following previous works [27], [33], and [34], we split this data set into a training, a validation, and a testing data set with 1200, 100, and 670 video clips, respectively.

MSR-VTT: This data set was proposed by Xu *et al.* [59] in 2016. They aim to provide a new large-scale video benchmark for supporting video understanding, especially for the task of translating videos into text. In total, this data set contains 10k Web video clips and 200k clip-sentence pairs in total. Each clip is annotated with 20 natural sentences by 1327 AMT workers. This data set is collected from a commercial video search engining, and so far, it covers the most comprehensive categories and diverse visual content, representing the largest data set in terms of sentences and vocabularies. We run our experiments on their updated version with sentence quality control. This data set is divided into three subsets: 65% for training, 5% for validating, and 30% for testing.

A. Evaluation Metrics

To evaluate the performance of our model, we utilize the following four evaluation metrics: bilingual evaluation understudy [59], Metric for Evaluation of Translation with Explicit ORdering (METEOR) [60], consensus-based image description evaluation (CIDEr) [61], and Recall-Oriented Understudy



M:	a woman is playing a guitar.	a monkey is playing with a dog.	a man is slicing a carrot.
M+S:	a girl is singing. a girl is playing a guitar. a girl is playing a guitar. a woman is playing a guitar. a girl is playing a guitar.	a monkey is teasing a dog. a monkey is playing with a dog. a monkey is pulls a dog's tail. a monkey is playing with a dog. a monkey is teasing a dog.	the person is cutting the something. a person is slicing a piece of cards. a person is cutting a cucumber. a person is slicing a carrot. the person is cutting the something.
GT:	a girl is playing guitar. a woman plays the guitar. a lady singing and playing guitar.	a monkey pulls a dog's tail. a monkey is playing with dog. a monkey is teasing a dog.	a boy is making a paper-butterfly. the person is showing how to make butterfly. a person is folding a piece of paper.
M:	a man is running.	a woman is cutting a cucumber.	a woman is talking on the bed.
M+S:	people are playing soccer. a man is running. a soccer player is running. a man is running. men are playing soccer.	a woman is chopping vegetables. a woman is cutting a vegetable. a woman is slicing a vegetable. a person is cutting vegetables. a woman is cutting vegetables.	a woman is laying on a bed. a woman is dancing. a woman is laying on a bed. a woman is dancing. a woman is laying on a bed.
GT:	the men are playing soccer. the people are playing. men are playing soccer.	a woman is slicing parsley. a person chops up some parsley. parsley is being cut.	a little girl is waking up. a child is laying in bed. a little girl is laying in bed.
M:	the zebras are playing.	a person is cooking.	a polar bear is playing.
M+S:	some zebras are playing. two zebras are playing. zebras are playing. two zebras are playing. a group of zebras are playing.	a man is putting sauce into a pan. a person is cooking. a man is cooking something. a man is pouring oil into a pan. the person is cooking.	a polar bear is walking on the snow. a polar bear is walking on the ice. a polar bear is running on the snow. a polar bear is walking on the snow. a polar bear is walking on the snow.
GT:	two zebras were dancing two zebras are playing. the zebras are playing.	a man is pouring sauce to a pan. a man cooking in his kitchen. a chef is preparing food.	two polar bears are fighting. two bear are walking. fighting of bears.

Fig. 4. Demonstration of our results, which are generated by repeatedly inputting each video five times into our trained model on the MSVD data set. Our model is able to generate different captions based on the different hidden stochastic variables.

for Gisting Evaluation [62]. In addition, Microsoft COCO evaluation server [63] has implemented these metrics, so we directly call such evaluation functions to test the performance of video captioning.

B. Experimental Settings

1) Video Appearance Feature Extraction: The experimental results obtained by Xu *et al.* [59] show that applying different pooling methods (i.e., single frame, meaning pooling, and soft attention) obtains different performance. Both mean pooling and soft attention perform significantly better than the single frame. The soft attention performs slightly better than mean pooling with 0.6% BULE@4 and 0.6% METEOR increase, but it involves more operations. Therefore, we apply a mean pooling to a set of frame-level features to generate a representative video-level feature. In addition, we follow previous work [27] to uniformly sample $K = 28$ frames from each clip for controlling video frames duplication. Deep CNNs achieved a great success in image feature extraction. Therefore, in this paper, we, respectively, use the ResNet-152[1] and GoogLeNet,[2] the two state-of-the-art CNNs, to extract video-frame level features to analyze our model.

About GoogLeNet, Szegedy *et al.* [39] introduced an inception module, an optimal local sparse structure in convolutional vision networks, and stacked these modules to construct a 22-layer inception network. The inception module is made up of 1 × 1 convolutions, 3 × 3 convolutions, 5 × 5 convolutions, and 3×3 max pooling layers. They used asynchronous stochastic gradient descent (SGD) with 0.9 momentum and decreased the learning rate by 4% every 8 epochs to learn the parameters in GoogLeNet. About ResNet, He *et al.* [22] introduced a deep residual learning framework, which is constructed by building blocks, to solve the degradation problem of training accuracy. A building block is made up of three convolutions layer: 1 × 1, 3 × 3, and 1 × 1 convolutions. They trained the entire network by SGD with backpropagation. They set the learning rate as 0.1 and divided it by 10 when the error plateaus. In this paper, we used the 152-layer ResNet with 5.7% validation top-5 error on ImageNet.

Like most previous works on video captioning [25], [27], [33], we extract the video-frame level features with pretrained deep CNNs and store these features on disks first and then feed them into the MS-RNN model both for training and testing. We did not fine-tune or retrain these deep CNNs but directly extract features from the pool5 layer with parameters shared at GitHub. The results show that ResNet-152 features perform well.

2) Sentence Preprocessing: For the MSVD data set, we tokenize it by first converting all words to lowercases and second utilizing the WordPunct function from NLTK toolbox to tokenize sentences and remove punctuations. As a result, we obtain a vocabulary with 13 010 words from the MSVD training data set. For the MSR-VTT data set, after tokenization, we obtain a 23 662 size vocabulary from its training data set. For each data set, we use the one-hot vector (1-of-N encoding, where N is the vocabulary size) to represent each word.

3) Training Details: For dealing with sentences with an arbitrary size, we add a begin-of-sentence tag ⟨bos⟩ to start each sentence and an end-of-sentence tag ⟨eos⟩ to end each sentence. During training, we maximize the loss function by taking the video and its corresponding ground-truth sentence label as the inputs.

In addition, in our experiments, we use ADADELTA, which can dynamically adjust the learning rate, to learn parameters and set the beam search size as 5. Empirically, we set all the M-LSTM unit sizes as 512, all the B-LSTM unit sizes as 512, the dimension of latent variables as 256, and the word embedding size as 512. Our objective function (13) is optimized over the whole training video sentence pairs with mini-batch 64 in size of MSVD and 256 in size of MSR-VTT. We stop training our model until 500 epochs are reached or until the evaluation metric does not improve on the validation set at the patience of 20. In addition, we multiply the KL term by a scalar, which starts at 0.01 and linearly increases to 1 over the first 20 epochs.

TABLE I

PERFORMANCES OF OUR MS MODEL OBTAINED BY REPEATEDLY INPUTTING TEST VIDEOS INTO OUR MODEL FIVE TIMES

Time	B@1	B@2	B@3	B@4	M	C	RL
1	**83.2**	**72.8**	63.5	53.4	33.9	73.7	**70.4**
2	82.7	72.6	**63.7**	53.6	**34.0**	75.2	70.3
3	82.4	72.1	62.9	52.8	33.6	74.7	69.8
4	83.0	72.8	63.6	**53.8**	**34.0**	76.6	**70.4**
5	83.1	72.7	63.5	53.1	33.6	73.9	70.0
mean	82.9	72.6	63.5	53.3	33.8	74.8	70.2

4) Testing Details: During testing, our model takes the video and a begin-of-sentence tag ⟨bos⟩ as the inputs to generate sentences to describe the input video. After the parameters are learned, we perform the generation with beam search [64]. All experiments are conducted on Ubuntu 14.04 with an Intel(R) Core(TM) i7-5930K CPU, a GeForce GTX TITAN Z GPU, and 64-GB memory cards. And Theano [65] library is utilized to construct models.

In addition, our model incorporates latent variables for ascertaining the true nature about video caption and has potential to describe video from different aspects. Thus, we have repeatedly input the test videos into our trained model five times. Each time we obtain a performance showing in Table I. Finally, we obtain an average performance. Moreover, Fig. 4 shows some output examples.

C. Results on MSVD Data Set

In this paper, we propose to utilize the probability distribution of latent variables to depict uncertainty; thus, for each time, our model may generate different descriptions. In this section, we run the testing five times and report the results in Table I. The performance of each testing is quite stable and reasonable. By checking the generated sentences (see Fig. 4), we can see that our model can describe a video from various aspects, and likely in real life, human provides various sentences to describe one video to fit their intents.

D. Component Analysis

In this paper, we design two core components: an M-LSTM layer and an S-LSTM layer, which affect the performance of our algorithm. In this section, we study their performance variance with the following two settings:

1) only using M-LSTM for video captioning (M);
2) incorporating M-LSTM and S-LSTM for video captioning (M+S).

In this subexperiment, we first conduct the experiments on the MSVD data set and use ResNet to extract frame features.

Table II lists the results, which demonstrate that our MS-RNN model with both M-LSTM and S-LSTM outperforms M-LSTM only on all evaluation metrics, with a 1.3% M, 3.3% C, and 1% RL performance increase.

In Fig. 4, we show some example sentences generated by our approach, with only M-LSTM and with both M-LSTM

[1] https://github.com/KaimingHe/deep-residual-networks

[2] https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet

TABLE II

EXPLORING MS-RNN. THE TOP MODEL USES ONLY M-LSTM, WHILE THE BOTTOM MODEL INTEGRATES M-LSTM AND S-LSTM. B, M, C, AND RL ARE SHORT FOR BLUE, METEOR, CIDER, AND ROUGE-L, RESPECTIVELY. ALL VALUES ARE REPORTED AS PERCENTAGE (%)

Model	B@1	B@2	B@3	B@4	M	C	RL
M	82.7	71.8	62.7	52.2	32.5	71.5	69.2
M+S	**82.9**	**72.6**	**63.5**	**53.3**	**33.8**	**74.8**	**70.2**

TABLE III

COMPARISON OF THE COMPUTATIONAL COST. THE TOP MODEL USES ONLY M-LSTM, WHILE THE BOTTOM MODEL INTEGRATES M-LSTM AND S-LSTM. THE TIME COSTS OF AVERAGE FORWARD-BACKPROPAGATION AND AVERAGE FORWARD ARE CALCULATED DURING ONE UPDATE (BATCH SIZE = 64). THE TESTING DURATION IS CALCULATED WITH THE BEAM SEARCH ALGORITHM (BEAMWIDTH = 5) ON THE MSVD TESTING DATA SET

Model	AVG Forward-Backpropragation	AVG Forward	Testing
M	187.3 ms	15.4 ms	71.41 s
M+S	362.4 ms	35.4 ms	74.50 s

and S-LSTM, respectively. From Fig. 4, we have the following observations.

1) Both M-LSTM and M-LSTM+S-LSTM are able to generate accurate descriptions for a video. In addition, the results generated by M-LSTM+S-LSTM are generally better than the M-LSTM method, which is consistent with the results reported in Table II.
2) M-LSTM is deterministic, and it can only generate one sentence, while M-LSTM+S-LSTM can produce different sentences.
3) In general, M-LSTM+S-LSTM can provide more specific, comprehensive, and accurate descriptions than M-LSTM. For example, in the top-left example, M-LSTM generates "a women is playing a guitar," while M-LSTM+S-LSTM provides "a girl is singing" and "a women is playing with a guitar." From the middle bottom, we can see that M-LSTM provides a wrong description "cucumber," while M-LSTM+S-LSTM generates "vegetables" and a set of verbs "slicing, chopping, and cutting."
4) Our MS-RNN model may produce duplicate and comprehensive results, which is consistent with the nature of video captioning.
5) The last column shows some wrong examples. For the top-right example, both the methods provide wrong descriptions, "cutting a cucumber" and "slicing a carrot." This is mainly because the MSVD data set contains many videos about cooking and few videos about folding paper, which leads to an overfitting problem, In addition, the right middle is also inaccurate. This is because both our models only take video appearance features as inputs and ignores the motion features. For the right bottom example, our model does not correctly identify the number of objects in some cases.

TABLE IV

COMPARING THE QUALITY OF SENTENCE GENERATION ON DIFFERENT VIDEO SPATIAL REPRESENTATIONS ON THE MSVD DATA SET. (V), (G), AND (R) STANDS FOR THE VGGNET, GOOGLENET, AND RESNE, RESPECTIVELY. THIS EXPERIMENT IS CONDUCTED ON THE MSVD DATA SET. ALL THE VALUES ARE REPORTED AS PERCENTAGE (%)

Model	B@1	B@2	B@3	B@4	M	C
LSTM-E(V)[27]	74.9	60.9	50.6	40.2	29.5	-
h-RNN(V)[34]	77.3	64.5	54.6	44.3	31.1	62.1
SA(G)[28]	79.1	63.2	51.2	40.6	29.0	-
MFA-LSTM(R)[30]	81.3	69.8	60.5	50.4	32.2	69.8
MS-RNN(G)	80.3	68.4	58.7	48.0	31.0	66.6
MS-RNN(R)	**82.9**	**72.6**	**63.5**	**53.3**	**33.8**	**74.8**

E. Comparison of Computational Cost

In this section, we compare computational time cost between M-LSTM and M-LSTM+S-LSTM and show the results in Table III. The training time cost of M-LSMT+S-LSTM is longer than M-LSTM, and the testing time cost is close to that of M-LSTM. The results are reasonable because there are three LSTM layers in MS-RNN during the training phase, but two LSTM layers during the testing phase.

F. Comparison Results on MSVD Data Set

In this section, we conduct experiments to examine how different video representations work on video captioning, as well as comparing our model with existing approaches. In addition, all the approaches in these subexperiments only take one type video representation extracting from VggNet (V), GoogleNet (G), or ResNet (R). We conduct our experiments on the MSVD data set.

Table IV lists the experimental results. From Table IV, we have following observations.

1) With only appearance features, our MS-RNN (R) model achieves the best performance on all evaluation metrics. Compared with the state-of-the-art method MFA-LSTM (R), our model achieves significantly better performance with 1.6%, 2.8%, 3%, 2.9%, 1.6%, and 5% increase on B@1, B@2, B@3, B@4, M, and C, respectively.
2) For video captioning task, the RestNet-based video representation performs better than both VggNet-based and GoogleNet-based video features. Specifically, our model RestNet feature performs better than GoogleNet features. For the whole experimental results, the approaches (SCN-LSTM and MFA-LSTM) with ResNet-based features perform better than the methods with GoogleNet or VggNet-based features.
3) Compared with the methods using attention mechanisms, e.g., temporal attention [27], our MS-RNN (R) achieves even better results with 3.8%, 9.4%, 12.3%, 12.7%, and 4.8% increase on B@1, B@2, B@3, B@4, and M by using a simple mean pooling strategy. This indicates the advantages of our proposed MS-LSTM.

We also compare our methods with the others using multiple features. Specifically, in this section, we compare our model using only appearance features with six state-of-the-art methods: LSTM-E(V+C) [26], spatial attention (SA) (G+3-DCNN) [27], HRNE-AT(G+C) [32], h-RNN(V+C) [33],

TABLE V

PERFORMANCE COMPARISON WITH METHODS USING BOTH APPEARANCE AND MOTION VIDEO FEATURES. THIS EXPERIMENT IS CONDUCTED ON THE MSVD DATA SET

Model	B@1	B@2	B@3	B@4	M	C
LSTM-E(V+C)[27]	78.8	66.0	55.4	45.3	31.0	-
SA(G+3D)[28]	80.0	64.7	52.6	42.2	29.6	51.7
h-RNN(V+C)[34]	81.5	70.4	60.4	49.9	32.6	65.8
MFA-LSTM(R+C)[30]	**82.9**	72.0	62.7	52.8	33.4	68.9
SCN-LSTM(R+C) [68]	-	-	-	51.1	33.5	**77.7**
MS-RNN(R)	**82.9**	**72.6**	**63.5**	**53.3**	**33.8**	**74.8**

TABLE VI

EXPERIMENT RESULTS ON THE MSR-VTT DATA SET. SA-LSTM RUNS EMPLOY SOFT ATTENTION OVER THE FRAME-LEVEL FEATURES EXTRACTED FROM DEEP NETWORK, WHILE MP-LSTM AND OUR METHOD UTILIZE MEAN POOLING OVER THE FRAME-LEVEL VIDEO FEATURES

Model	B@4	M	C	RL
MP-LSTM(V)[41]	34.8	24.8	-	-
MP-LSTM(C)	35.4	24.8	-	-
MP-LSTM(V+C)	35.8	25.3	-	-
SA-LSTM(V)[28]	35.6	25.4	-	-
SA-LSTM(C)	36.1	25.7	-	-
SA-LSTM(V+C)	36.6	25.9	-	-
MFA-LSTM(R+C)[30]	39.2	**26.9**	**44.6**	**60.1**
MS-RNN(R)	**39.8**	26.1	40.9	59.3

MFA-LSTM(R+C) [29], and SCN-LSTM(R+C) [66], which make use of both appearance and motion video features. Here, V and R are short for VggNet and ResNet, which are used to extract appearance features. 3-D and C are short for 3-DCNN and C3D, which are used to generate video motion features.

The experimental results are shown in Table V. Although our model only uses appearance features, it performs better than the existing methods on B@2 (72.6%), B@3 (63.5%), B@4 (53.3%), and M (33.8%) and achieves comparable results on B@1 (82.9%) and C (74.8%).

G. Comparison Results on MSR-VTT Data Set

In this section, we compare our method with MP-LSTM [40] and SA-LSTM [27] on the MSR-VTT data set. In addition, to obtain the appearance features, the MP-LSTM and our MS-RNN are based on the mean pooling strategy, while SA-LSTM is based on a soft-attention mechanism. In theory, soft attention is more complex than mean pooling but usually provides better visual features. The experimental results are shown in Table VI, and we have the following observations. MS-RNN

1) gains a promising performance with 39.8% B@4, 26.1% M, 40.9% C, and 59.3% RL on the MSR-VTT data set.
2) Overall with the same visual input (VGG-19, VGG-19+C3D, or C3D), SA-LSTM performs better than MP-LSTM. However, SA is based on the soft attention. In other words, in theory, SA-LSTM takes better visual features as inputs. Compared with MP-LSTM, our MS-RNN (R) outperforms MP-LSTM (VGG-19+C3D) with 4% B@4 and 0.8% M increase. Compared with SA-LSTM, our MS-RNN (R) outperforms SA-LSTM(VGG-19+C3D) with 3.2% B@4. Compared with MFA-LSTM(R+C), our model achieves comparable results on B@4, M, and RL by using single feature (R).

V. CONCLUSION AND FUTURE WORK

In this paper, we propose an MS-RNN framework for video captioning. This paper has shown how to extend the modeling capabilities of RNN by approximating both prior distribution and true posterior distribution with a nonlinear latent layer (S-LSTM). In addition, MS-RNN achieves the state-of-the-art performance with only mean video appearance features and is comparable with the counterparts, which take both video appearance and motion features. Last but not least, the proposed model can be applied to a wide range of video analysis applications.

In the future, we will integrate the state-of-the-art attention mechanism [27] with our model to further improve the video captioning performance. Moreover, the motion feature will be considered.

REFERENCES

[1] L. Zhang, Y. Han, Y. Yang, M. Song, S. Yan, and Q. Tian, "Discovering discriminative graphlets for aerial image categories recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5071–5084, Dec. 2013.

[2] J. Song, L. Gao, F. Nie, H. T. Shen, Y. Yan, and N. Sebe, "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.

[3] S. Cai, W. Zuo, and L. Zhang, "Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization," in *Proc. ICCV*, Oct. 2017, pp. 511–520.

[4] D. Tao, Y. Guo, Y. Li, and X. Gao, "Tensor rank preserving discriminant analysis for facial recognition," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 325–334, Jan. 2018.

[5] X. Zhu, X.-Y. Jing, X. You, W. Zuo, S. Shan, and W.-S. Zheng, "Image to video person re-identification by learning heterogeneous dictionary pair with feature projection matrix," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 3, pp. 717–732, Mar. 2018.

[6] D. Tao, Y. Wen, and R. Hong, "Multicolumn bidirectional long short-term memory for mobile devices-based human activity recognition," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1124–1134, Dec. 2016.

[7] X. Wang, L. Gao, P. Wang, X. Sun, and X. Liu, "Two-stream 3-D convNet fusion for action recognition in videos with arbitrary size and length," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 634–644, Mar. 2018.

[8] X. Wang, L. Gao, J. Song, X. Zhen, N. Sebe, and H. T. Shen, "Deep appearance and motion learning for egocentric activity recognition," *Neurocomputing*, vol. 275, pp. 438–447, Jan. 2018.

[9] D. Tao, Y. Guo, M. Song, Y. Li, Z. Yu, and Y. Y. Tang, "Person re-identification by dual-regularized kiss metric learning," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2726–2738, Jun. 2016.

[10] T. Liu, D. Tao, M. Song, and S. J. Maybank, "Algorithm-dependent generalization bounds for multi-task learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 227–241, Feb. 2017.

[11] X. Liu, D. Tao, M. Song, L. Zhang, J. Bu, and C. Chen, "Learning to track multiple targets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1060–1073, May 2015.

[12] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe, "Quantization-based hashing: A general framework for scalable image and video retrieval," *Pattern Recognit.*, vol. 75, pp. 175–187, Mar. 2018.

[13] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2017.

[14] J. Song, H. Zhang, X. Li, L. Gao, M. Wang, and R. Hong, "Self-supervised video hashing with hierarchical binary auto-encoder," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3210–3221, Jul. 2018.

[15] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[16] 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.

[17] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *Int. J. Comput. Vis.*, vol. 50, no. 2, pp. 171–184, Nov. 2002.

[18] M. W. Lee, A. Hakeem, N. Haering, and S.-C. Zhu, "SAVE: A framework for semantic annotation of visual events," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[19] M. U. G. Khan, L. Zhang, and Y. Gotoh, "Human focused video description," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1480–1487.

[20] P. Hanckmann, K. Schutte, and G. J. Burghouts, "Automated textual descriptions for a wide range of video events with 48 human actions," in *Proc. ECCV*, 2012, pp. 372–380.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2014, pp. 1–14.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

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

[24] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, 2014.

[25] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—Video to text," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 4534–4542.

[26] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4594–4602.

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

[28] Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, and H. T. Shen, "Attention-based LSTM with semantic consistency for videos captioning," in *Proc. ACM MM*, 2016, pp. 357–361.

[29] X. Long, C. Gan, and G. de Melo, "Video captioning with multi-faceted attention," *CoRR*, 2016.

[30] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4651–4659.

[31] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," *CoRR*, 2016.

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

[33] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu, "Video paragraph captioning using hierarchical recurrent neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4584–4593.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014.

[35] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[36] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH,* 2010, pp. 1045–1048.

[37] A. Farhadi *et al.*, "Every picture tells a story: Generating sentences from images," in *Proc. ECCV*, 2010, pp. 15–29.

[38] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele, "Translating video content to natural language descriptions," in *Proc. Int. Conf. Comput. Vis.*, Dec. 2013, pp. 433–440.

[39] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[40] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proc. NAACL HLT*, 2015, pp. 1494–1504.

[41] 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.

[42] J. Song, Z. Gao, L. Guo, W. Liu, D. Zhang, and H. T. Shen, "Hierarchical LSTM with adjusted temporal attention for video captioning," in *Proc. IJCAI*, 2017, pp. 2737–2743.

[43] Z. Wang *et al.*, "Diverse image captioning via grouptalk," in *Proc. IJCAI*, 2016, pp. 2957–2964.

[44] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. NIPS*, 2017, pp. 5580–5590.

[45] Y. Li and Y. Gal, "Dropout inference in Bayesian neural networks with alpha-divergences," in *Proc. ICML*, 2017, pp. 2052–2061.

[46] Y. Gal, R. McAllister, and C. E. Rasmussen, "Improving PILCO with Bayesian neural network dynamics models," in *Proc. Data-Efficient Mach. Learn. Workshop (ICML)*, 2016, pp. 1–7.

[47] L. Uusitalo, A. Lehikoinen, I. Helle, and K. Myrberg, "An overview of methods to evaluate uncertainty of deterministic models in decision support," *Environ. Model. Softw.*, vol. 63, pp. 24–31, Jan. 2015.

[48] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. NIPS*, 2015, pp. 3483–3491.

[49] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep Kalman filters," *CoRR*, 2015.

[50] I. V. Serban *et al.*, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Proc. AAAI*, 2017, pp. 3295–3301.

[51] M. Fraccaro, S. K. Sønderby, U. Paquet, and O. Winther, "Sequential neural models with stochastic layers," in *Proc. NIPS*, 2016, pp. 2199–2207.

[52] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. NIPS*, 2015, pp. 2980–2988.

[53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[54] X. Chen and C. L. Zitnick, "Learning a recurrent visual representation for image caption generation," *CoRR*, 2014.

[55] H. Fang *et al.*, "From captions to visual concepts and back," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 1473–1482.

[56] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *CoRR*, 2012.

[57] D. L. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proc. ACL HLT*, 2011, pp. 190–200.

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

[59] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. ACL*, 2002, pp. 311–318.

[60] M. J. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. Workshop Statist. Mach. Transl.*, 2014, pp. 376–380.

[61] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4566–4575.

[62] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization Branches Out*, 2004, p. 10.

[63] X. Chen *et al.*, "Microsoft COCO captions: Data collection and evaluation server," *Comput. Sci.*, 2015.

[64] D. Furcy and S. Koenig, "Limited discrepancy beam search," in *Proc. IJCAI*, Jul. 2005, pp. 125–131.

[65] J. Bergstra *et al.*, "Theano: A CPU and GPU math expression compiler," in *Proc. Python Sci. Comput. Conf. (SciPy)*, Austin, TX, USA, Jun. 2010.

[66] Z. Gan *et al.*, "Semantic compositional networks for visual captioning," *CoRR*, 2016.

Jingkuan Song received the Ph.D. degree in information technology from The University of Queensland, Brisbane, QLD, Australia, in 2014.

He was with Columbia University, New York, NY, USA, as a Post-Doctoral Research Scientist, from 2016 to 2017, and the University of Trento, Trento, ON, Canada, as a Research Fellow, from 2014 to 2016. He is currently a Professor with the University of Electronic Science and Technology of China, Chengdu, China. His current research interests include large-scale multimedia retrieval, image/video segmentation, and image/video annotation using hashing, graph learning, and deep learning techniques.

Dr. Song is a Guest Editor of IEEE TRANSACTIONS ON MULTIMEDIA and *World Wide Web Journal* and the Area Chair of ACM Multimedia 2018.

Yuyu Guo is currently pursuing the master's degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China.

He is currently involved in image/video understanding and image/video captioning.

Lianli Gao received the Ph.D. degree in information technology from The University of Queensland, Brisbane, QLD, Australia.

She is currently an Associate Professor in computer science with the Future Media Center and the School of Information Technology and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include machine learning, deep learning, and computer vision.

Xuelong Li (M'02–SM'07–F'12) is currently a Full Professor with the Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.

Alan Hanjalic (M'99–SM'08–F'16) is currently a Professor and the Head of the Multimedia Computing Group, Delft University of Technology, Delft, The Netherlands. His current research interests include multimedia search, recommender systems, and social media analytics.

Dr. Hanjalic was the Chair of the Steering Committee of the IEEE TRANSACTIONS ON MULTIMEDIA. He was a Keynote Speaker at the IEEE International Workshop on Multimedia Signal Processing 2013, the International Multimedia Modeling Conference 2012, and the Pacific-Rim Conference on Multimedia 2007. He has also been a General or Program (Co-)Chair of the organizing committees of multimedia conferences, such as ACM Multimedia, ACM International Conference on Content-Based Image and Video Retrieval/ACM International Conference on Multimedia Retrieval, and IEEE International Conference on Multimedia and Expo. He has been a member of the Editorial Board of several scientific journals in the multimedia field, including the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, the *ACM Transactions on Multimedia*, and the *International Journal of Multimedia Information Retrieval*. He is an Associate Editor-in-Chief of the *IEEE Multimedia Magazine*.

Heng Tao Shen received the B.Sc. degree (Hons.) and the Ph.D. degree from the Department of Computer Science, National University of Singapore, Singapore, in 2000 and 2004, respectively.

He then joined The University of Queensland, Brisbane, QLD, Australia, as a Lecturer, a Senior Lecturer, a Reader, and became a Professor in 2011. He is currently a Professor of the National Thousand Talents Plan, the Dean of the School of Computer Science and Engineering, and the Director of the Center for Future Media, University of Electronic Science and Technology of China, Chengdu, China. He is also an Honorary Professor with The University of Queensland. His current research interests include multimedia search, computer vision, artificial intelligence, and big data management. He has made continuous contributions to big data indexing and retrieval and developed the first real-time near-duplicate video retrieval system. He has published over 200 peer-reviewed papers, among which over 140 appeared in Chinese Computing Federation a ranked publication venues, such as ACM Multimedia, IEEE Conference on Computer Vision and Pattern Recognition, 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 Data Bases, International Conference on Data Engineering, *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and VLDB Journal.