Delft University of Technology

Video Captioning by Adversarial LSTM

Yang, Yang; Zhou, Jie; Ai, Jiangbo; Bin, Yi; Hanjalic, Alan; Shen, Heng Tao

DOI
10.1109/TIP.2018.2855422

Publication date
2018

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Image Processing

Citation (APA)
Yang, Y., Zhou, J., Ai, J., Bin, Y., Hanjalic, A., & Shen, H. T. (2018). Video Captioning by Adversarial LSTM. *IEEE Transactions on Image Processing*, *27*(11), 5600-5611. Article 8410586. https://doi.org/10.1109/TIP.2018.2855422

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

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

Video Captioning by Adversarial LSTM

Yang Yang, *Member, IEEE*, Jie Zhou, Jiangbo Ai, Yi Bin, Alan Hanjalic, *Fellow, IEEE*, Heng Tao Shen, *Senior Member, IEEE*, and Yanli Ji

***Abstract*— In this paper, we propose a novel approach to video captioning based on adversarial learning and long short-term memory (LSTM). With this solution concept, we aim at compensating for the deficiencies of LSTM-based video captioning methods that generally show potential to effectively handle temporal nature of video data when generating captions but also typically suffer from exponential error accumulation. Specifically, we adopt a standard generative adversarial network (GAN) architecture, characterized by an interplay of two competing processes: a "generator" that generates textual sentences given the visual content of a video and a "discriminator" that controls the accuracy of the generated sentences. The discriminator acts as an "adversary" toward the generator, and with its controlling mechanism, it helps the generator to become more accurate. For the generator module, we take an existing video captioning concept using LSTM network. For the discriminator, we propose a novel realization specifically tuned for the video captioning problem and taking both the sentences and video features as input. This leads to our proposed LSTM–GAN system architecture, for which we show experimentally to significantly outperform the existing methods on standard public datasets.**

***Index Terms*— Video captioning, adversarial training, LSTM.**

I. Introduction

VIDEO captioning is referred to as the problem of generating a textual description for a given video content. The interdisciplinary nature of this problem opens vast new possibilities for interacting with video collections and there has been increased research effort on this topic observable over the past years [1]–[5]. This interdisciplinary nature, however, also poses significant research challenges at the intersection between the fields of natural language processing and computer vision. Typically, these challenges have been pursued as extrapolations of the solutions proposed earlier for image captioning [6]–[8]. These solutions perform classification in the visual domain with the goal to generate salient regions, linking these regions with some predefined textual attributes and then synthesize a sentence by completing a predefined generative model using the recognized attribute [1], [3], [9]–[12].

Different from static pictures [13], [14], the content of a video is significantly more rich and unlikely to be captured well by simply extrapolating the methods developed for images. This richness comes mainly through the temporal aspect of video content. And comparing to the retrieval and annotation technology [15]–[18], the captioning task relies more on narrowing the semantic gap between the visual and textual information. In order to take this aspect into account, Yao *et al.* [1] proposed a 3D Convolutional Neural Network (3D-CNN) structure, applying convolution not only spatially, but also temporally. However, 3D-CNN can only capture the information over a short period of time due to the limit of the convolution kernel size. Venugopalan *et al.* [3] implemented a Long-Short Term Memory (LSTM) network, a variant of a Recurrent Neural Network (RNNs), to model the global temporal structure in an entire video snippet. However, this method was shown to accumulate the grammatical errors exponentially and to result in decreasing association among the generated words with the increasing video length. Furthermore, the traditional caption generative models usually select words with maximum probability, which typically results in a rather monotonous set of generated sentences, like for instance around the words “playing” and “doing”, both appearing rather often in the common training data sets.

In order to eliminate this deficiency of LSTM, in this paper, we propose a novel approach that expands the LSTM concept towards an *adversarial learning* concept. As illustrated in Fig. 1, this expansion involves adding a “discriminator” module to the system architecture, which acts as an adversary with respect to the sentence generator. While the generator has the objective to make the generated sentences as close to its existing generative model as possible, the discriminator has the objective to ensure that generated sentences are reasonable and natural for people better understanding. This interplay of two concurring processes has recently been introduced as adversarial learning, with a Generative Adversarial Network (GAN) [19]–[21] as a basic realization. Consequently, we refer to our proposed approach as LSTM-GAN. We will demonstrate that this expansion of the LSTM concept will enable the video captioning process to improve the accuracy and diversity of generated captions and their robustness to increasing video length.

Applying a GAN to the context of video captioning is, however, not straightforward. A GAN is designed for real-valued, but continuous data and may have difficulty handling sequences of discrete words or tokens as mentioned

Manuscript received November 6, 2017; revised May 4, 2018; accepted July 5, 2018. This work was supported in part by the National Natural Science Foundation of China under Projects 61572108 and 61632007 and in part by the 111 Project under Grant B17008. *(Corresponding author: Heng Tao Shen.)*

Y. Yang, J. Zhou, J. Ai, Y. Bin, H. T. Shen, and Y. Ji are with the Center for Future Media, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: dlyyang@gmail.com; jiezhou0714@gmail.com; jiangboml@gmail.com; yi.bin@hotmail.com; shenhengtao@hotmail.com; yanliji@uestc.edu.cn).

A. Hanjalic is with the Multimedia Computing Group, Intelligent Systems Department, Delft University of Technology, 2628CD Delft, The Netherlands (e-mail: a.hanjalic@tudelft.nl).

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

Digital Object Identifier 10.1109/TIP.2018.2855422

© 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

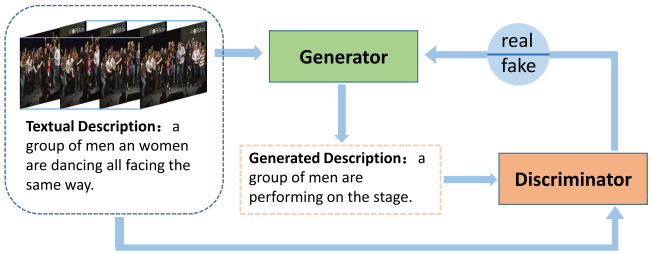

Fig. 1. An illustration of the modular structure of the proposed video captioning model characterizerd by an interplay of the generator G that generates text sentences and the discriminator D (adversary) that verifies the sentences. The optimization goal is that G deceives D, by generating sentences that are not distinguishable from reference sentences.

in [22]. The reason lies in that the gradient of the loss from the discriminator based on the output of the generator is used to move the generator to slightly change the way the sentences are generated. However, if the output of the generator consists of discrete tokens, the slight change guidance by the discriminator may not work because there may be no token in the used dictionary to signal the desired level of change towards the generator [20].

In order to overcome this problem, we propose a embedding layer which can transform the discrete outputs into a consecutive representation [23]. Besides that, since the outputs of our generative model are a sequence, ordinary discriminative model, consisted of several fully connected layers, has a poor ability for classifying the sequence-sentence. For solving this problem, we propose a new realization of the discriminative model. Specifically, we replace the fully connected layer, as originally proposed in [19], with a novel convolutional structure, previously proposed by Zhang *et al.* [25], Kim [26], and Collobert *et al.* [27]. Our discriminative model consists of convolutional layer, max-pooling layer and fully connected layer. The convolutional layer will produce local features and retain the local coherence around each word of the sequence-sentence. After max-pooling layer, the most important information of the sentence will be effectively extracted. Those informations are denoted by a fixed length of vector. Additionally, we also introduce a multimodal input for the discriminative model. We sent not only the sentence to the discriminative module but also the video feature generated from our first LSTM layer (Encoder) of generative module. The novel methods for incorporating the original inputs with the video feature will help to generate more relevant discriptions about the input video. This method has been confirmed to be effective by our experiment.

To our knowledge, we are the first to propose the method for generating video description via advesarial learning. The remainder of this paper is organized as follows. In Section II, we review the existing work related to the problem of video captioning and position our proposed solution with respect to it. Section III presents our proposed LSTM-GAN video captioning model. The model is evaluated experimentally in Section IV where we compare our approach with the relevant existing methods on four public datasets: MSVD, MSR-VTT, M-VAD and MSR-VTT. This comparison revealed that our approach significantly outperforms the existing methods on these datasets, which justifies our methodological and algorithmic design choices. Section V concludes the paper by a discussion of the obtained video captioning performance and pointers to future work on this topic.

II. Related Work

A. Video Captioning

Generating a textual description for a given video content, also called video captioning, has been showing increasingly strong potential in computer vision. The primary challenges of this research lie in two aspects: adequately extracting the information from the video sequences and generating grammar-correct sentences easy for the human to understand. The early research for generating video descriptions mainly focused on extracting useful information *e.g.*, object, attribute, and preposition, from given video content. Krishnamoorthy *et al.* [27] aim at generating more precise words to describe the objects in the video. Their method includes a content planning stage and a surface realization stage. In their work, object detection and activity recognition modular are used to extract the related words about the video and then it applies a template-based approach to generate sentences. Reference [28] is another work about using template-based approach. Different from [27], the biggest contribution of this paper is building a hierarchical semantic model to classify different words. Besides, this model can even detect unseen verbs with the help of knowledge mining from web-scale textual corpora. No doubt that the performances of those methods are limited by the accuracy of the detection of the word and the robustness of the template. In order to overcome the problem abovementioned, some novel approaches based on RNNs [28]–[35] are proposed with the development of deep learning. Different from the template-based approach in [27], Pan *et al.* [10] take advantage of the CNN architecture to get the representation of every frame and then applies an average pooling operation to those frame presentations. For generating video description of the inputted video, they take advantage of the RNNs architecture to produce sequence output with an end-to-end training pattern. What's more, it is noteworthy that they innovatively embed the sentence and the video content into a feature with a uniform dimension. And by measuring the feature diversity of the embedded sentences and input videos, they get the value of relevance loss between these two features. By minimizing this loss, the semantic gap will become narrowed. With the popularization of Convolutional Neural Network for an impressive performance in feature extraction, generating the description for a video has made great breakthroughs. Pan *et al.* [10] used both 2D CNN and 3D CNN to process video clips and average pooling operation to process over all the clips to generate a single D_v-dimension vector which will be used to generate descriptions for video. But all the abovementioned methods could not work well for a short video clip containing multi-event. Yu *et al.* [36] proposed an attention model which use an attention mechanism to select the most important video features and has yielded impressive performance in multi-event videos.

The combination of CNN and LSTM architecture has become the mainstream form for video captioning research

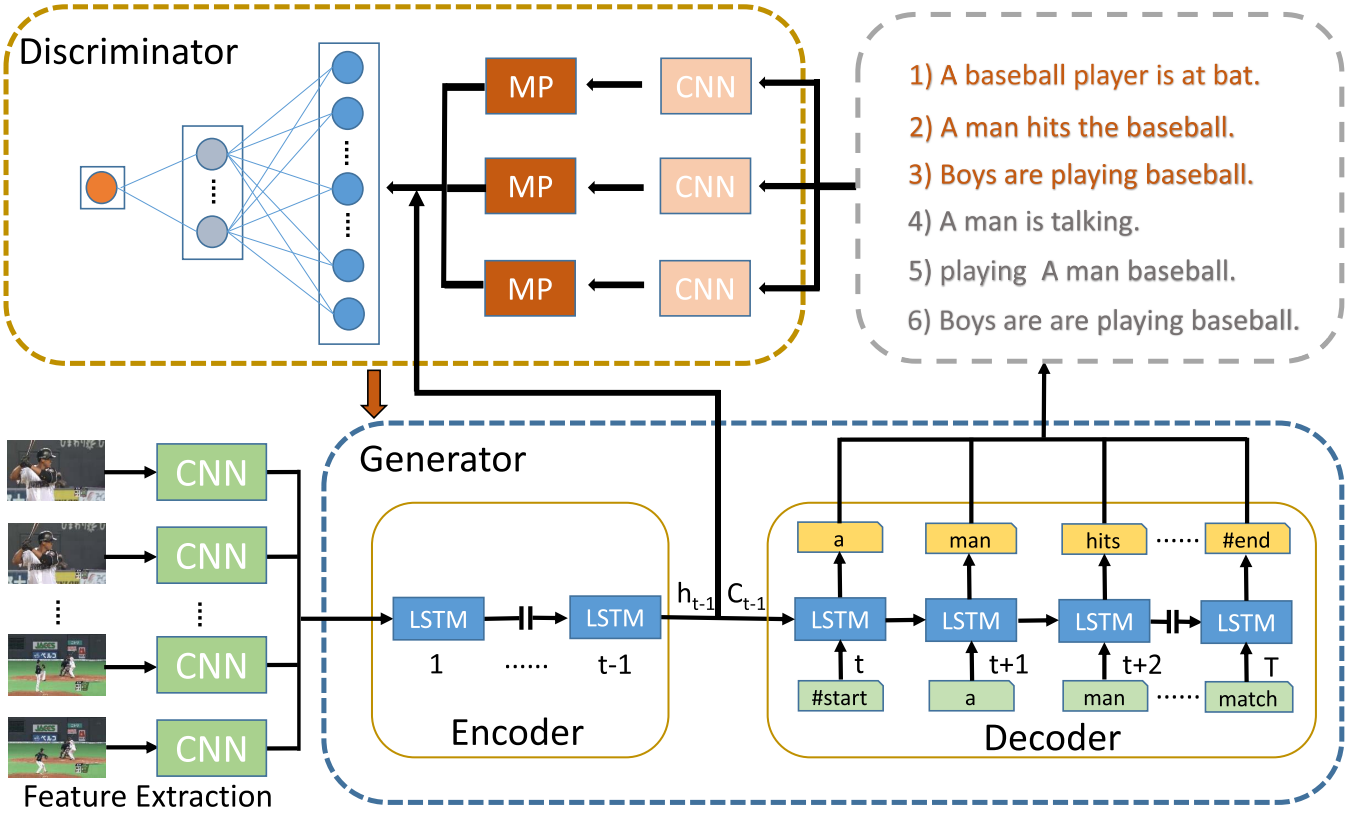

Fig. 2. We propose LSTM-GAN incorporating a joint LSTMs with adversarial learning. Our model consists of generative model and discriminative model. The generative model tries to generate a sentence for the video as accurately as possible, but the discriminative model tries to distinguish whether the input sentences is from reference sentence or generated sentences. The orange input sentences for discriminative model represent the reference sentences, otherwise badly constructed sentences or uncorrelated sentences generated by generative model. MP in the figure denotes the max-pooling.

which also demonstrated a excellent performance. Both feature extraction and words generation methods are important for the quality of description for the input video. Although the LSTM scheme has proved promising performance for handling the temporal nature of video data in the temporal process, the LSTM scheme critical deficiency is shown to accumulate the grammatical errors exponentially and may result in decreasing association among the generated words with the increasing video length. Based on the problem, we consider if there is a structure that can discriminate whether the generated descriptions are reasonable and relevant to the video. Inspired by the generative adversarial network firstly for generating an image, we proposed our model LSTM-GAN incorporating a joint LSTMs with adversarial learning. This model consists of a generative model and discriminative model. The generative model is used for encoding the video clips and generates sentences, while the discriminative model is trying to distinguish whether the input sentences are from reference sentence or generated sentences (as in Fig. 2).

B. GAN in Natural Language Processing

Generative Adversarial Networks (GAN), introduced by Goodfellow *et al.* [19], has achieved promising success in generating realistic synthetic real-valued data. On account of impressive performance, GAN has been widely applicated in computer vision (CV) and natural language processing (NLP). Original Generative Adversarial Networks consist of generative model and discriminative model. The generative model tries to produce data, which is to mix the spurious with the genuine. While the discriminative model learns to determine whether the data is from genuine distribution or not. Typical applications can be found in, for instance, image synthesis [22] where discrete data with normal distribution are used to generate realistic images. Besides showing the excellent performances in the image processing, GAN structures also work well in natural language processing recently. Li *et al.* [37] proposed a dialog generation method using Adversarial reinforce model. They adopted policy gradient training method to encourage the generator to generate utterances that are indistinguishable from human-generated dialogs. Later, Press *et al.* [38] presented a text generating model with RNNs for both generator and discriminator. Different from policy gradient training in [37], this work applied curriculum learning, as a result, it vastly improves the quality of generated sequences. For captioning research, Dai *et al.* [39] firstly presented an image captioning method, which can generate sentences much more closely to what human saying than original caption models.

In summary, our work presents the first effort to incorporate the GAN with the video caption model. Our model can

overcome the aforementioned problem and the performance of our model is also excellent through our experimental results.

III. LSTM-GAN for Video Captioning

Fig. 2 depicts our LSTM-GAN architecture which consists of a generator G, and a discriminator D. The generator G uses an encoder-decoder architecture to generate descriptions for relevant videos under the umbrella of incorporating a CNN architecture as the discriminator D which is for evaluating whether the generated sentences reasonable or not. Specifically, we begin this section by introducing the fundamental Long Short-Term Memory Networks (LSTM) and Generative Adversarial Network (GAN) model briefly. In the remainder of the module, we elaborate on the algorithm's theory and design choices underlying the proposed framework in more detail.

A. Long Short-Term Memory Networks

Traditional RNNs [40], [41] is designed to learn complex temporal dynamics by mapping the input sequences to a sequence of hidden states and then generating outputs via the following recurrence equations as Eq. 1:

$$\begin{aligned} h_t &= \psi(W_h x_t + U_h h_{t-1} + b_h), \\ o_t &= \psi(U_h h_t + b_o), \end{aligned} \tag{1}$$

where the W_h, U_h denote the weight matrices and b denotes the bias, ψ is an element-wise non-linear function, such as RELU or hyperbolic tangent. x_t is the input, $h_t \in \mathbb{R}^M$ is the hidden state with M hidden units, o_t is the output at time t. The traditional RNNs have proven successful in text generation and speech recognition. But it is difficult to handle well about long-range temporal dependencies videos because of the exploding and vanishing gradient problem. The LSTM network proposed by Hochreiter and Schmidhuber [41] has demonstrated to be able to effectively prevent the gradient vanishing and explosion problems [41] during back-propagation through time (BPTT) [42]. This is because it incorporates memory units, which facilitate the network to learn long-range temporal dependencies, to forget previously hidden states and to update them given new information. More specifically, as illustrated in Fig. 3 and implemented in our framework, LSTM incorporates several control gates and a memory cell. Let x_t, c_t and h_t represent the input, cell memory and hidden control states at each time t respectively. Given a sequence of inputs $(x_1, \ldots, x_T)$, the LSTM will compute the hidden control sequence $(h_1, \ldots, h_T)$ and the cell memory sequence $(c_1, \ldots, c_T)$. Formally, this process can be described by the following set of equations:

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i), \\ f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f), \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o), \\ g_t &= \varphi(W_g[h_{t-1}, x_t] + b_g), \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\ h_t &= o_t \odot tanh(c_t). \end{aligned} \tag{2}$$

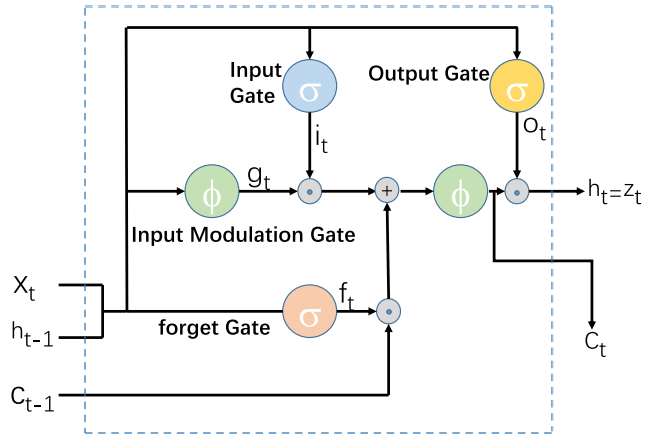

Fig. 3. A diagram of a basic LSTM memory cell used in our paper, where the input gate, forget gate and output gate represents by i_t, g_t, o_t respectively.

Here, $\odot$ stands for element-wise product and W_j-like matrices are the LSTM weight parameters. Additionally, σ and φ denote the sigmoid and hyperbolic non-linear functions respectively.

B. Generative Adversarial Networks

In a nutshell, a GAN-based learning approach [19] involves simultaneous training of two network models, generator G and discriminator D. In the field of image super-resolution, image generation, image to image translation, Neuro-Linguistic Programming, GAN has made great contributions [43]–[46]. Through the interplay of the two learning processes, the G and D models facilitate each other interactively to individually reach their goals. The generator G tries to generate real data given a noize $z \sim P_{noise}(z)$, while the discriminator $D \in [0, 1]$ aims at classifing the real data $x \sim p_{data}(x)$ and the fake data $G(z) \sim P_G(z)$ generated from G. More specifically, the generator G also learns to generate samples from the generator distribution P_G by transforming a noise data $z \sim P_{noise}(z)$ into a sample G_z (z will be replaced with input video to generate corresponding descriptions in our later experiments). Similarly, the goal of D is to distinguish between samples from the true data distribution P_{data} and the generator's distribution P_G as accurate as possible. After a period of competition between the two processes, the two network models will achieve some degree of balance with the optimal discriminator being $D(x) = P_{data}(x)/(P_{data}(x) + P_G(z))$ and with the generator being able to generate data which are difficult for the discriminator D to distinguish whether from training data or synthetic data. This "game" is steered towards convergence by the optimization criteria expressed through loss functions, designed for both G and D. We define the following loss functions L_G and L_D:

$$\begin{aligned} \mathcal{L}_G &= -\frac{1}{m}\sum_{i=1}^{m} \log(1 - D(G(z^{(i)}))), \\ \mathcal{L}_D &= -\frac{1}{m}\sum_{i=1}^{m} [\log(D(x^i)) + (\log(1 - D(G(z^{(i)}))))], \end{aligned} \tag{3}$$

where $\{z^{(1)}, \ldots, z^{(m)}\}$ are a mini-batch of m noise samples by random initialization and $\{x^{(1)}, \ldots, x^{(m)}\}$ are a mini-batch of m samples from true data distribution $p_{data}(\mathbf{x})$.

C. Problem Definition

Consider a video $\boldsymbol{V}$ including a sequence of n sample frames where $\boldsymbol{V} = \{v_1, v_2, \ldots, v_n\}$, with associated caption $\boldsymbol{S}$ where $\boldsymbol{S} = \{w_1, w_2, \ldots, w_m\}$ consisting of m words. Let $v_i \in \mathbb{R}^{D_v}$ and $w_j \in \mathbb{R}^{D_w}$ denote the D_v-dimensional visual presentations of the i-th frame in video $\boldsymbol{V}$ and the D_w-dimensional textual features of the j-th word in sentence $\boldsymbol{S}$, respectively. In our work, our goal is to maximize the conditional probability of an output sequence $(w_1, \ldots, w_m)$ given an input sequence $(v_1, \ldots, v_n)$. The conditional probabilities over the sentences can be defined as follows:

$$p(\boldsymbol{s}|\boldsymbol{v}) = p(w_1, \ldots, w_m|v_1, \ldots, v_n). \tag{4}$$

This problem is similar to the problem of machine translation in natural language processing, where a sequence of words serves as input into a generative model that outputs a sequence of words as the translation result. What is different from aforementioned is that, in our work, we replace the textual input by our video frames and look forward to a sequence of caption as output. What is more, we not only expect to get the relevant description of the input videos but also to make the sentences natural and reasonable for people to understand.

D. Proposed Solution

For the sake of overcoming the above-mentioned problem, in this section, we devise our model to generate video description under the umbrella of an adversarial system. Specifically, our overall framework consists of a generative model G and discriminative model D. The generative model G, similar to sequence-to-sequence models [47], defines the policy that generates a sequence of the relevant description given a short video. The discriminative model D is a binary classifier that takes a sequence of sentences $\{s, y\}$ as input and outputs a label $D(S) \in [0, 1]$ indicating whether the sentence is natural, reasonable and grammatical correct. In particular, several variants of our designed model are utilized to compare with other methods. We now elaborate on the implementation of our designed architecture.

1) Objective Function: In order to achieve faster convergence of the objective, we firstly pre-training the generative model G and the discriminative model D, respectively. For G, similar to sequence-to-sequence models [47], our goal is to estimate the conditional probability $p(\boldsymbol{S}|\boldsymbol{V})$ where $\boldsymbol{V} = (v_1, v_2, \ldots, v_t)$ is an input sequence consisting of a sample of frames and $\boldsymbol{S} = (w_1, w_2, \ldots, w_{t_1})$ is the corresponding output sequence as a descriptive texture for the input video. t and t_1 represents the length of the video and the input sentence respectively. As sequence-to-sequence models [3], we conclude the follow objective function:

$$\begin{aligned} p(\boldsymbol{S}|\boldsymbol{V}) &= p(w_1, w_2, \ldots, w_{t_1}|v_1, v_2, \ldots, v_t) \\ &= \prod_{i=1}^{t_1} p(w_i|\boldsymbol{V}, w_1, \ldots, w_{i-1}). \end{aligned} \tag{5}$$

From Equation (5) for our model with θ and output sequence $\boldsymbol{S} = (w_1, w_2, \ldots, w_{t_1})$, we could get the optimal θ with the follow formula:

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{t} \log p(w_t|h_{n+t-1}, w_1, \ldots, w_{t-1}; \theta). \tag{6}$$

where h_{n+t-1} denotes the hidden state at time step $n + t - 1$ which will be introduced detaily in the next section. For D, our primary purpose is to train a classifier which can be used for sentence encoding and mapping the input sentence to an output $D(\boldsymbol{S}) \in [0, 1]$ representing the probability of $\boldsymbol{S}$ is from the ground-truth captions, rather than from adversarial generator. The objective function of D for pre-training can be formalized into a cross-entropy loss as follow:

$$\begin{aligned} \mathcal{L}_D(\boldsymbol{Y}, D(\boldsymbol{S})) = -\frac{1}{m}\sum_{i=1}^{m}[(Y_i)\log(D(\boldsymbol{S_i})) \\ + (1 - (Y_i)(\log(1 - D(\boldsymbol{S_i})))]. \end{aligned} \tag{7}$$

where m denote the number of examples in a batch, $\boldsymbol{Y_i}$ and $D(\boldsymbol{S_i}))$ represent the real label and predicted value of discriminator respectively.

Hence, when we train our overall framework called LSTM-GAN, the whole training procedure for the LSTM-GAN is same as in Fig. 2. We aim at minimizing the log likelihood formulated as follows:

$$\begin{aligned} minimizing: \mathcal{L}(\boldsymbol{S}|\boldsymbol{V}) = \mathbb{E}_{s\sim P(s), v\sim P(v)}[\log P(\boldsymbol{S}|\boldsymbol{V})] \\ + \mathbb{E}_{s\sim P(s)}[\log(1 - D(G(\boldsymbol{S})))]. \end{aligned} \tag{8}$$

2) Generative Model: As mentioned before, We use a joint recurrent neural networks, also called encoder-decoder LSTM similar to sequence-to-sequence models [47], as the generative model. The encoder architecture is used to encode the video features into a fixed dimension vector. While the decoder architecture decodes the vector into natural sentences. To begin with, we adopt VGG16 [48] as the CNN architecture to map the sequence frames $\boldsymbol{V} = (v_1, v_2, \ldots, v_t)$ into a feature matrix $W_v \in \mathbb{R}^{D_d \times D_t} = (w_{D_1}, \ldots, w_{D_t})$. D_d and D_t denote the dimensions of a feature vector and the number of frames, respectively. Particularly, the encoder LSTM net, referred to as "encoder", maps the input embedding presentations $w_{D_1}, \ldots, w_{D_t}$, namely features matrix, into a sequence of hidden states $h_1, h_2, \ldots, h_t$ mentioned in previous section by using the update function as eq. (2) recursively.

According to the above description, we make it clear that the last status h_t as the presentations of the whole video, generated from "encoder", will be sent to the decoder LSTM which is referred to as "decoder". Specifically, given the h_t (in our Fig. (2) marked as h_{t-1} for straightforward) and a corresponding sentence $\boldsymbol{S}$ also referred to textual description in Fig. 1, encoded with one-hot vectors (1-of-N encoding, where N is the size of the vocabulary), our decoder LSTM is conditioned step by step on the tth word and on the previous h_{t-1}, and is trained to produce the next word of the description for input video. We commit ourselves to minimize objective function as eq. (5) to generate an excellent performance generative model. The probability of those words is modeled via a softmax function applied on the output of the decoder. As we know, those

words should be in an one-hot format which not only are high dimensionality but also are discrete and discontinuous. Passing those words with a one-hot format to the D (Discriminator) will make it difficult to pass the gradient update. Although score function based algorithms, such as REINFORCE [49] obtains unbiased gradient estimation for discrete variables by using Monte Carlo estimation. However, the variance of the gradient estimation could be large [50]. In order to cope with this problem, we adopt a soft-argmax function similar to the one proposed in [24]:

$$w_{t-1} = \boldsymbol{\varepsilon}_{w_e}(\text{softmax}\,\langle Vh_{t-1} \odot L\rangle, W_e). \tag{9}$$

Here, $W_e \in \mathbb{R}^{Z\times C}$ is a word embedding matrix (to be learned) which is similar to the GloVe [51] and transforms the one-hot encoding of words to a dense lower dimensional embedding, C is the dimension of the embedded word (1024 in our experiments) and Z is the size of vocabulary in our training data. V is the set of parameters and encodes the h_{t-1} to a vector. w_{t-1} represents the generated word of LSTM at t^{th} step. L is a big enough integer which would make the vector of softmax $\langle Vh_{t-1} \odot L\rangle$ closes to a one-hot form. Each value of it is constrained to be either approximately 0 or 1 which can help the w_{t-1} more close to $W_e[t-1]$ (suppose the value at the $t-1$ position is the largest of Vh_{t-1}) and also help the word embedding to be more smooth and speed up the loss function to convergence. $\boldsymbol{\varepsilon}$ denotes a function that maps the decoder output space to a word space.

3) Discriminative Model: In the discriminator D, our primary purpose is to maximize the probability of assigning the correct label to both training sentences and generated sentences from G. As well konwn, deep discriminative models such as deep neural network (DNN) [52], convolutional neural network (CNN) [25] and recurrent convolutional neural network (RCNN) [53] have shown an impressive performance in complicated sequence classification tasks. In our paper, referring to [54] which has recently been shown of great performance in text classification using CNN, we choose the CNN as our discriminator.

As illustrated in Fig. 2, our discriminator consists of a convolution layer and a max-pooling operation, which can capture the most useful local features produced by the convolutional layers [25], [26], [54], over the entire sentence for each feature map. The input sentences to our discriminator contain both the ground-truth sentences as the true label (also called textual description in Fig. 1) and generated sentences generated by our generator as the false label. For convenience, we fix the length of input sentences by adopting the length of longest sentence in a mini-batch (padded 0 when necessary). A sentence of length T is represented as a matrix $X_d \in \mathbb{R}^{C\times T} = (x_{d1}, \ldots, x_{dT})$ by concatenating the word embeddings as columns, where T is the length of sentence and C is the dimension of a word. Then a kernel $W_c \in \mathbb{R}^{C\times l}$ applies a convolution operation to a window size of T words to produce a feature map as one of the representations of the input sentence. The specific process like Fig. 4 could be formulated as follow:

$$Out = f(X * W_c + b) \in \mathbb{R}^{T-l+1}. \tag{10}$$

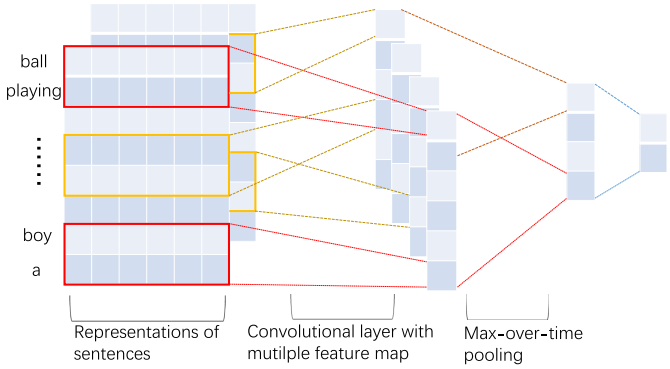

Fig. 4. The convolution process of input sentence in discriminative model.

where f(·) is a nonlinear activation function (in our experiments, we use the RELU), $b \in \mathbb{R}^{T-l+1}$ is the bias vector and * represents the convolution operator. We then apply a max-over-time pooling operation over the generated feature maps and take its maximum value, $\widetilde{Out} = max\{Out_1, \ldots, Out_{T-l+1}\}$. As proved by Collobert *et al.* [26], the max-pooling operation can not only help capture the most important feature by effectively filtering out less informative compositions of words, but also guarantees that the extracted features are independent of the length of the input sentence. To be more persuasive, we conduct a contrast experiment using max-pooling and mean-pooling respectively on discriminator. The accuracy of classification with max-pooling improves over the mean-pooling by 1.8% which proves that max-pooling have a better capacity to classify the sentences. The dataset for classification consists of the ground-truth of corresponding video and the false sentences generated from the generator.

The above process describes how the features are extracted from CNN for the sentences. Although the present CNN architecture process a impressive performance in complicated sequence classification tasks. But in this paper, we devote to generate the sentence that not only needs to be natural and reasonable for people to understand but also can describe the input video precisely. In order to overcome this difficulty, we adopt a linear connection method which could integrate the video feature and pooled feature from different kernels into a new representation. Given a video feature $F \in \mathbb{R}^H$ extracted from the last hidden layer in encoder of generator, where H is the dimension of hidden layer, we concatenate it with its corresponding textual feature $\widetilde{Out} \in \mathbb{R}^{H1}$, where $H1$ is the dimension of $\widetilde{Out}$. We will get a synthetic feature vector $F_{new} \in \mathbb{R}^{H+H1}$. We then pass F_{new} to a fully connected softmax layer to get the probability $D(X_d) \in [0, 1]$, an output close to 1 indicates a bigger probability that X is drawn from the real data distribution or not.

Previous literature [19], [55] has discussed the difficulty in training GAN model using the gradient-based method. In order to reduce the instability in training process, we initialize the LSTM parameters for generator and CNN parameters for discriminator by pre-training a standard encoder-decoder LSTM model and a standard CNN classification model aforementioned.

E. Attention Mechanism

Attention structure is a kind of extension of LSTM. It has been widely used in many previous jobs [6], [56], [57]. Rather than compress an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. For example, when a captioning model uses "a child is running on the ground" to describe a picture, the attention model will focus on the area where the child located in the picture when the model generate the word "child", while generating the word "ground", the model will focus on the ground in the picture.

In our model with attention, we use attention structure in Generative Model when generating words. At each time step t, the attention model accept the video's visual information, which is a n by d matrix, where "n" is the number of the visual vectors and "d" is the dimension of visual vector. In the attention structure, each visual vector is multipeld by different weight $\boldsymbol{\alpha}$, which reflecting the importance of the corresponding visual vector in that time step. After that, the vectors are added together and become the input of LSTM unit to generate a new word. For example, when generating the word "boy", the weight for visual vectors with boy's information will be larger; While generating the word "soccer", the weight for visual vectors with soccer's information will be larger. The visual information mentioned above are the output of hidden layers of Encoder illustrated in Fig. 2 in our paper. The weight $\boldsymbol{\alpha}$ at timestamp t is the result of softmax for the combination for hidden state at timestamp t-1 and the visual information. For a video with n visual contexts $C = c_1, c_2, \ldots, c_n$, we have:

$$\begin{aligned} e_{t,i} &= h'_{t-1} U_c c_i \\ \alpha_{t,i} &= \frac{exp(e_{t,i})}{\sum_{j=1}^{n} exp(e_{t,j})} \end{aligned} \tag{11}$$

where h'_{t-1} is the transposed vector of hidden state at the last time stamp, U_c is the mapping matrix and c_i is i^{th} of visual vector.

In our model with attention, we use attention mechanism for generating the description of the input video. At each time step t, the attention unit accepts video visual information vectors c_t, which are the output of hidden layers of Encoder structure illustrated in Fig. 2 in our paper. After the encoding stage, we get the $(c_1, \ldots, c_n)$ where n denotes the num of frames of input video.(problem definition use the a n, need to check) As eq.12, the vectors are multiplied by different weights in attention unit and these weights can determine which frame of input video s_t should be concerned with current time step. After that, the y_t and the last hidden status represent the new input for the next LSTM unit to generate the new word. The weights mentioned above are calculated dynamically and the sum of those weights is 1.

$$\begin{aligned} y_t &= \sum_{i=1}^{n} \alpha_{t,i} c_i, \\ \sum_{i=1}^{n} \alpha_{t,i} &= 1. \end{aligned} \tag{12}$$

And eq. 2 with attention can be written as follows:

$$\begin{aligned} i_y, t &= \sigma(W_i[h_{t-1}, x_t, y_t] + b_i), \\ f_y, t &= \sigma(W_f[h_{t-1}, x_t, y_t] + b_f), \\ o_y, t &= \sigma(W_o[h_{t-1}, x_t, y_t] + b_o), \\ g_y, t &= \varphi(W_g[h_{t-1}, x_t, y_t] + b_g), \end{aligned} \tag{13}$$

IV. Experimental Validation

In this section we describe the experimental validation of our proposed video captioning approach in detail.

A. Datasets

To verify the impressive performance of our video captioning by abversarial training approach, we evaluate and compare our experimental results on four large public datasets, including MSVD [9], MSR-VTT [58], M-VAD [59] and MPII-MD [60].

1) MSVD: MSVD dataset consists of 1970 short video snippets downloaded from YouTube. Each video snippet is annotated with around 40 textual descriptions collected via crowdsourcing. This results in 80839 sentences. In our experiments, we split the data into train, validation and test sets, containing 1200, 100 and 670 videos snippets, respectively.

2) MSR-VTT: MSR-VTT is a new large-scale benchmark video captioning dataset specially suitable for the video-to-text translation task. This dataset was created by using 257 popular queries from a commercial video search engine and by collecting 118 videos for each query. Each video is annotated with about 20 natural sentences provided by 1,327 crowdsourcing workers. In total, MSR-VTT provides 10K web video clips (41.2 hours) and 200K clip-sentence pairs, covering various semantic categories and diverse visual content.

3) M-VAD and MPII-MD: Montreal Video Annotation Dataset (M-VAD) and MPII Movie Description Corpus (MPII-MD), are two datasets which contain Holly-wood movie snippets with descriptions sourced from script data and audio description. M-VAD contains about 49000 DVD movie snippets extracted from 92 DVD movies. And MPII-MD is composed of about 68000 movie snippets from 94 movies. Each snippet is equipped with a single sentence from movie scripts and DVS.

B. Evaluation Metrics and Baselines

Similar to traditional machine translation, the generated descriptive sentences for the correlative video also can be measured by comparing a set of reference sentences. Recently, some common metrics in machine translation also are used for evaluating visual captioning, i.e., BLEU [61],ROUGEL [62] and METEOR [63]. ROUGE-L simply to compile statistics the maximum matches of generated sentences and reference sentences. BLEU-N (N = 1, 2, 3, 4) usually measures the precision of N-gram matches. METEOR is used to measure semantic matcher which is more robust for more consistent with human's judgment. In our experiment, for convenient comparison and robust results, we adopt BLEU and METEOR to evaluate our proposed approach following [5] and [66].

We used the evaluation script provided by Chen *et al.* [66] to compute scores on our datasets.

To empirically verify the merit of our LSTM-GAN models, we compared the following state-of-the-art methods on the four mentioned datasets.

- LSTM [11]: LSTM, incorporating CNN with RNN framework, attempts to directly translate from video pixels to natural language. The video representation is generated by performing mean pooling over the frame features across the entire video.
- S2TV [3]: S2VT adopts a stack of two LSTMs for the encoding and decoding of the inputs respectively and word presentations are learnt jointly in a parallel manner. Besides that, S2VT incorporates both RGB and optical flow inputs.
- LSTM-E [10]: LSTM-E integrates 2D CNN and 3D CNN to extract video feature representation, and simultaneously explores the learning of LSTM and visual-semantic embedding for video captioning.
- TA [1]: TA combines the frame representation from GoogleNet and video clip representation based on a 3D CNN trained on hand-crafted descriptors. whats more, the model add a weighted attention mechanism to dynamically attend to specific temporal regions of the video while generating sentence.

C. Experimental Setup

For video representation, we extract the video features from VGG16 network. We take the output of the 4096-way fc7 layer from VGG16 pre-trained on the ImageNet ILSVRC12 dataset [10] as the input representation. Before training the overall model LSTM-GAN, we first separately pre-train the G and D to speed up the convergence of the overall model. For the pre-training process of G, we set the video features as the input of encoder and the corresponding reference sentences as the output of the decoder. For the pre-training process of D, we first collect both the reference sentences and generated sentences from the output of G as the training data. Following [24], we used a confusion training strategy. In detail, we randomly swap two or three words to construct tweaked counterpart sentences as negative samples. Referring to the generated sentences from pre-traing G which generates repeated word frequently about some specific words, we also select those sentences including aforementioned word to copy twice or three times as the incorrect sentence for D. Besides that, we employ filter windows (l of W_c) of sizes $3, 4, 5$ with 300 feature maps each, hence each sentence is represented as a 900-dimensional vector. For ensuring the correlation between generated description and input video, we concatenate the video representation to the sentence vector as a multi-modal input [5]. In this way, we can make D have the ability to distinguish the sentences with grammar mistakes from correct ones.

For the training process, we adopt the Stochastic Gradient Descent (SGD) optimization function, for which we set the momentum attribute to 0.9. We first set the learning rate as 0.001. When the loss became unstable and started to fluctuate repeatedly, we decrease the learning rate to 0.0001 for follow-up training. In order to avoid over-fitting, we set the dropout ratio to 0.5 for all full-connected layers. In addition, we also added the weight decay and set the value to 0.01. The bath-size in our experiment is 128. If the memory capacity is large enough, we can also set a bigger value. All experiments were implemented in Theano, using a NVIDIA GTX1080 GPU with 8GB memory. The model was trained for about one day.

TABLE I

METEOR AND BLEU4 SCORES OF OUR LSTM-GAN AND OTHER EXISTING METHODS ON MSVD DATASET. ALL VALUES ARE REPORTED AS PERCENTAGE

Model	METEOR	BLEU@4
LSTM [11]	26.9	-
LSTM without GAN (Ours)	29.2	39.3
S2VT (RGB) [3]	29.2	-
S2VT (Optical Flow) [3]	**29.8**	-
LSTM-E (VGG) [10]	29.5	-
LSTM-GAN (Ours, without attention)	29.7	**40.3**
TA [1]	29.6	41.9
LSTM-GAN (Ours, with attention)	**30.4**	**42.9**

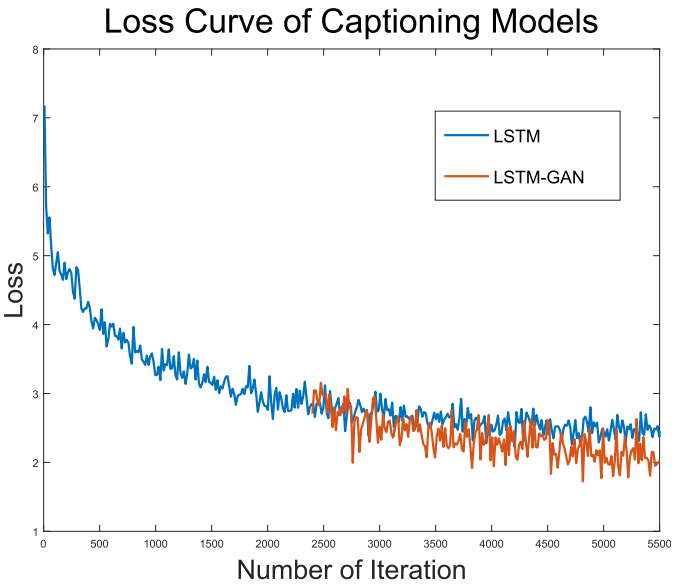

Fig. 5. Log-likelihood convergence performance of LSTM-GAN and LSTM, denoted by orange and blue curve respectively. And before 2300 of iterative times, there is a pre-training process.

D. Performance Comparison

1) Performance on MSVD Dataset: Table I summarizes the obtained results and Fig. 5 shows the log-likelihood of the convergence performance when training the network. Overall, the results on two evaluation metrics consistently demonstrate that our proposed LSTM-GAN achieves better performance than all the existing techniques including non-attention models (LSTM, S2VT, LSTM-E) and attention-based approaches (TA). Comparing to the two the metrics of METEOR and BLEU@4, we note that our LSTM-GAN method with attention achieved the best score of 30.4 and

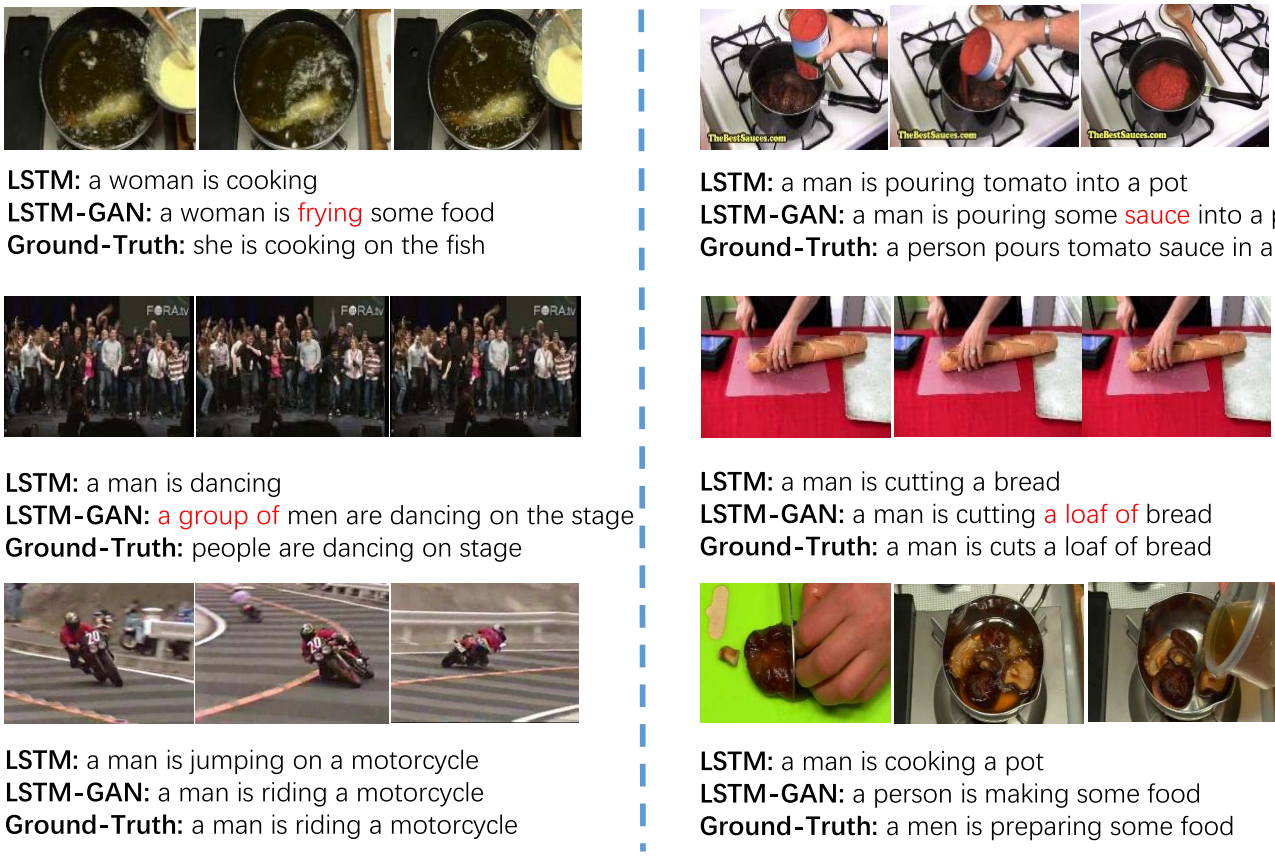

Fig. 6. Examples to demostrate the effectiveness of our model to generate much richer lexicon. We display the caption from ground-truth, LSTM-GAN(our model), and LSTM model respectively. We can observe that our model can describe the event of the video appropriately and generate more representative words like "sauce, frying, loaf".

42.9 respectively, outperforming all other methods. Table I shows the experimental result compared to other methods. By additionally incorporating attention mechanism to LSTM model, LSTM-GAN leads to a performance boost, demonstrating that adversarial training has the ability to improve the performance of our caption model. Additionally, we notice that our proposed LSTM-GAN (without attention) makes also a relative improvement over S2VT (RGB) which has a stack of two LSTMs layer one for encoding video and the other for decoding. The result effectively indicates that LSTM incorporating with adversarial training do benefit the learning of video sentence generation. But we also notice the method with the performance closest to this was S2VT with optical flow feature. Good performance of S2VT (Optical Flow) may be due to the usage of optical flow features, which are important for depicting the motion information of object in video. By incorporating with optical flow features, the model may generate more relevant descriptions to the video. After using attention mechanism to better handle the video features, our LSTM-GAN model has both sentence correction capability and feature processing capability. So that our LSTM-GAN with attention model outperforms all other models, including TA, which also incorporates with a weighted attention mechanism. Fig. 6 provides some examples come from ground-truth, LSTM-GAN (our model), and LSTM (without adversarial training) respectively. we can notice that our LSTM-GAN model can generate richer lexicon, which also is logically correct and more relevant to the video.

TABLE II
METEOR SCORES OF OUR LSTM-GAN AND OTHER EXISTING METHODS ON MSR-VTT DATASET. ALL VALUES ARE REPORTED AS PERCENTAGE

Model	METEOR	BLEU@4
LSTM [11]		
-(VGG-16)	24.7	34.7
LSTM-GAN (Ours,VGG-16)	**25.2**	34.5
TA [1]		
-(AlexNet)	23.8	34.8
-(GoogleNet)	25.2	35.2
-(VGG-16)	25.4	35.6
-(VGG-19)	25.4	35.6
LSTM-GAN (Ours, attention,VGG-16)	**26.1**	**36.0**

2) Performance on MSR-VTT Dataset: Table II lists the statistics and comparison in MSR-VTT datasets. We compare our experimental results with the baseline SA-LSTM proposed

TABLE III
METEOR SCORES OF OUR LSTM-GAN AND OTHER EXISTING METHODS ON M-VAD AND MPII-MD DATASET. ALL VALUES ARE REPORTED AS PERCENTAGE. (a) M-VAD DATABESE. (b) MPII-MD DATABASE.

(a)

Method	METEOR
TA [1]	4.3
LSTM [11]	6.1
LSTM without GAN (Ours)	5.9
LSTM with GAN (Ours)	**6.3**

(b)

Method	METEOR
SMT [60]	5.6
LSTM [11]	6.7
Visual-Labels [67]	7.0
S2VT [3]	7.1
LSTM without GAN (Ours)	6.6
LSTM with GAN (Ours)	**7.2**

by Xu *et al.* [60] which achieve the attention mechanism on MSR-VTT datasets. From the statistics in Table II, our proposed approaches LSTM-GAN achieves 26.1 and 36.0 on METEOR and B4 respectively which outperforms all other methods based on different feature extraction approaches. But we also notice that our LSTM-GAN model without attention has the better score about METEOR but has poor performance in B4 compared with the LSTM. The cause of the problem is most likely our LSTM-GAN generates longer sentences which cause the relatively lower score about B4. Actually, our model with adversarial learning generates more words (in total about 2000 words for all test data) than the model without adversarial learning. When added the attention mechanism which will help to select most important and relevant information about the video clips, our model outperforming all other methods. The fact shows that, with the help of attention mechanism, our LSTM-GAN can generate more reasonable descriptions for videos although with a longer sequence output.

3) Performance on M-VAD and MPII-MD Dataset: Table III lists the statistics and comparison in M-VAD and MPII-MD datasets, which are both more challenging due to the high diversity of visual and textual content. For M-VAD dataset, we compare our experimental results with the baseline TA [1] and LSTM [11]. From the first part of statistics in Table III, our proposed approaches LSTM-GAN achieves 6.3 on METEOR, which improves over the TA by 2.0% and over the LSTM by 0.2%. Comparing to our baseline model (LSTM without GAN), our model improves over it from 5.9% to 6.3%, proving the effectiveness of our proposed model.

For MPII-MD datasets, we achieve METEOR score of 7.2%, outperforming all the existing methods including SMT [60], LSTM [11], Visual-Labels [67] and S2VT [3]. Similar to the observations on MSVD, and MSR-VTT, Our LSTM with GAN model exhibits better performance than LSTM without for video captioning.

V. CONCLUSIONS

In this paper, we presented a first attempt to introduce the concept of adversarial learning in solving the video capturing problem. We believe that this concept has potential to significantly improve the quality of the captions, which is due to its ability to better control the capture generation process. This control is in this case done by the discriminator module, acting as an adversary to the caption generation module. In addition to making the fundamental adversarial learning framework based on the GAN paradigm suitable for dealing with discrete generator outputs, with our novel realization of the discriminator, we further improved the control mechanism. This was achieved by making the input into the discriminator multimodal. In this way, the sentences coming out of the generator were not only validated for grammatical correctness, but also for their relevance to the video content. The potential of our LSTM-GAN framework to improve the quality and diversity of captions was also demonstrated experimentally, through an elaborate experimental study involving multiple baseline approaches, four popular datasets, and two widely used evaluation metrics. We believe that the performance of LSTM-GAN could further be improved by relying on Reinforcement Learning. Reinforcement Learning has proven effective for tasks similar to video captioning, like for instance dialog generation [20].

REFERENCES

[1] L. Yao *et al.*, "Describing videos by exploiting temporal structure," in *Proc. ICCV*, 2015, pp. 4507–4515.

[2] J. Donahue *et al.*, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. CVPR*, 2015, pp. 2625–2634.

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

[4] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, and K. Saenko. (2015). "A multi-scale multiple instance video description network." [Online]. Available: https://arxiv.org/abs/1505.05914

[5] 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.

[6] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, vol. 14, 2015, pp. 77–81.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. CVPR*, 2015, pp. 3156–3164.

[8] A. F. Smeaton and I. Quigley, "Experiments on using semantic distances between words in image caption retrieval," in *Proc. SIGIR*, 1996, pp. 174–180.

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

[10] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui, "Jointly modeling embedding and translation to bridge video and language," in *Proc. CVPR*, 2016, pp. 4594–4602.

[11] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. (2014). "Translating videos to natural language using deep recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1412.4729

[12] G. Li, S. Ma, and Y. Han, "Summarization-based video caption via deep neural networks," in *Proc. ACM MM*, 2015, pp. 1191–1194.

[13] M. Hu, Y. Yang, F. Shen, L. Zhang, H. T. Shen, and X. Li, "Robust Web image annotation via exploring multi-facet and structural knowledge," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4871–4884, Oct. 2017.

[14] 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.

[15] 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.

[16] 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.

[17] F. Shen, Y. Yang, L. Liu, W. Liu, D. Tao, and H. T. Shen, "Asymmetric binary coding for image search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2022–2032, Sep. 2017, doi: 10.1109/TMM.2017.2699863.

[18] Y. Luo, Y. Yang, F. Shen, Z. Huang, P. Zhou, and H. T. Shen, "Robust discrete code modeling for supervised hashing," *Pattern Recognit.*, vol. 75, pp. 128–135, Mar. 2018.

[19] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[20] L. Yu, W. Zhang, J. Wang, and Y. Yu. (2016). "SeqGAN: Sequence generative adversarial nets with policy gradient." [Online]. Available: https://arxiv.org/abs/1609.05473

[21] H. Kwak and B.-T. Zhang. (2016). "Generating images part by part with composite generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1607.05387

[22] F. Huszár. (2015). "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" [Online]. Available: https://arxiv.org/abs/1511.05101

[23] 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.

[24] Y. Zhang, Z. Gan, and L. Carin, "Generating text via adversarial training," in *Proc. NIPS Workshop Adversarial Training*, 2016, pp. 1–6.

[25] Y. Kim. (2014). "Convolutional neural networks for sentence classification." [Online]. Available: https://arxiv.org/abs/1408.5882

[26] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Aug. 2011.

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

[28] S. Guadarrama *et al.*, "YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition," in *Proc. ICCV*, 2013, pp. 2712–2719.

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

[30] Y. Bin, Y. Yang, F. Shen, N. Xie, H. T. Shen, and X. Li, "Describing video with attention-based bidirectional LSTM," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2831447.

[31] Z. Gan *et al.*, "Semantic compositional networks for visual captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2017, pp. 5630–5639.

[32] 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.

[33] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating attractive visual captions with styles," in *Proc. CVPR*, 2017, pp. 955–964.

[34] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proc. CVPR*, 2017, pp. 6504–6512.

[35] Y. Pu, M. R. Min, Z. Gan, and L. Carin, "Adaptive feature abstraction for translating video to text," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16320

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

[37] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky. (2017). "Adversarial learning for neural dialogue generation." [Online]. Available: https://arxiv.org/abs/1701.06547

[38] O. Press, A. Bar, B. Bogin, J. Berant, and L. Wolf. (2017). "Language generation with recurrent generative adversarial networks without pre-training." [Online]. Available: https://arxiv.org/abs/1706.01399

[39] B. Dai, D. Lin, R. Urtasun, and S. Fidler. (2017). "Towards diverse and natural image descriptions via a conditional GAN." [Online]. Available: https://arxiv.org/abs/1703.06029

[40] F. Gers, "Long short-term memory in recurrent neural networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. Hannover, Hannover, Germany, 2001.

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

[42] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[43] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. (2016). "Amortised MAP inference for image super-resolution." [Online]. Available: https://arxiv.org/abs/1610.04490

[44] S. Ravanbakhsh, F. Lanusse, R. Mandelbaum, J. Schneider, and B. Póczos, "Enabling dark energy science with deep generative models of galaxy images," in *Proc. AAAI*, 2017, pp. 1488–1494.

[45] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[46] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. (2015). "Generating sentences from a continuous space." [Online]. Available: https://arxiv.org/abs/1511.06349

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

[48] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[49] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 229–256, 1992.

[50] C. J. Maddison, A. Mnih, and Y. W. Teh. (2016). "The concrete distribution: A continuous relaxation of discrete random variables." [Online]. Available: https://arxiv.org/abs/1611.00712

[51] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

[52] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. INTERSPEECH*, 2013, pp. 2345–2349.

[53] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI*, vol. 333, 2015, pp. 2267–2273.

[54] X. Zhang and Y. LeCun. (2015). "Text understanding from scratch." [Online]. Available: https://arxiv.org/abs/1502.01710

[55] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. NIPS*, 2016, pp. 2234–2242.

[56] X. Long, C. Gan, and G. de Melo. (2016). "Video captioning with multi-faceted attention." [Online]. Available: https://arxiv.org/abs/1612.00234

[57] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. CVPR*, 2016, pp. 4651–4659.

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

[59] A. Torabi, C. Pal, H. Larochelle, and A. Courville. (2015). "Using descriptive video services to create a large data source for video annotation research." [Online]. Available: https://arxiv.org/abs/1503.01070

[60] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3202–3212.

[61] 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.

[62] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. ACL*, Barcelona, Spain, vol. 8, 2004, pp. 1–8.

[63] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 376–380.

[64] H. Fang *et al.*, "From captions to visual concepts and back," in *Proc. CVPR*, 2015, pp. 1473–1482.

[65] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. CVPR*, 2015, pp. 3128–3137.

[66] X. Chen *et al.* (2015). "Microsoft COCO captions: Data collection and evaluation server." [Online]. Available: https://arxiv.org/abs/1504.00325

[67] A. Rohrbach, M. Rohrbach, and B. Schiele, "The long-short story of movie description," in *Proc. German Conf. Pattern Recognit.*, 2015, pp. 209–221.