Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Video Captioning with Listwise Supervision

Yuan Liu,[†] **Xue Li,**[‡] **Zhongchao Shi** [†]
[†]Ricoh Software Research Center (Beijing) Co., Ltd., Beijing, China [‡]Ricoh Company, Ltd., Yokohama, Japan
yuanliu.ustc@gmail.com, Setsu.Ri@nts.ricoh.co.jp, Zhongchao.Shi@srcb.ricoh.com

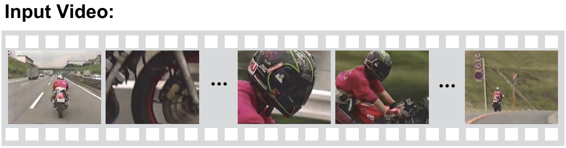
Abstract

Automatically describing video content with natural language is a fundamental challenging that has received increasing attention. However, existing techniques restrict the model learning on the pairs of each video and its own sentences, and thus fail to capture more holistically semantic relationships among all sentences. In this paper, we propose to model relative relationships of different video-sentence pairs and present a novel framework, named Long Short-Term Memory with Listwise Supervision (LSTM-LS), for video captioning. Given each video in training data, we obtain a ranking list of sentences w.r.t. a given sentence associated with the video using nearest-neighbor search. The ranking information is represented by a set of rank triplets that can be used to assess the quality of ranking list. The video captioning problem is then solved by learning LSTM model for sentence generation, through maximizing the ranking quality over all the sentences in the list. The experiments on MSVD dataset show that our proposed LSTM-LS produces better performance than the state of the art in generating natural sentences: 51.1% and 32.6% in terms of BLEU@4 and METEOR, respectively. Superior performances are also reported on the movie description M-VAD dataset.

Introduction

Accelerated by the tremendous increase in Internet bandwidth and storage space, video data has been generated, published and spread explosively, becoming an indispensable part of today's big data. This has encouraged the development of advanced techniques for a broad range of video understanding applications. A fundamental issue that underlies the success of these technological advances is the recognition. Previous research has predominantly focused on recognizing videos with a predefined set of individual words (tags). Recently, researchers have strived to automatically describe video content with a complete and natural sentence, which is illustrated in Figure 1. However, the need to understand not only video content (e.g., key objects, scenes and motions) but also express their spatio-temporal relationships in a natural sentence makes the task very challenging.

Despite the difficulty of this problem, there have been several attempts being proposed for attacking video captioning (Venugopalan et al. 2015a; Yao et al. 2015; Pan et al. 2016a) and its closely related task of image captioning (Vinyals et al. 2015; Yao et al. 2016). The basic idea is to employ Convolutional Neural Networks (CNN) to encode video/image content and Recurrent Neural Networks (RNN) to decode a sentence. We follow this philosophy for generating video description in this work.



Output Sentence:

- **LSTM:** a man is riding a car.
- **LSTM-LS:** a man is riding a motorcycle.
- **Ground Truth:** someone is riding a motorcycle. / a man is riding his motorcycle. / a man is riding on a motor bike.

Figure 1: Examples of video description generation.

While encouraging performances are reported, most existing works perform model learning on labeled video-sentence pairs separately, leaving the semantic relationships between sentences associated with different videos not fully exploited. A relative relationship indicates the correctness and logic of a sentence describing a video with respect to other sentences. Indeed, we observe that relative relationship is a semantically rich way by which humans describe and compare visual properties in the world. For example, in Figure 1, it is difficult to predict the correct objective "motorcycle" in the sentence generated by LSTM model (Venugopalan et al. 2015b) trained locally on video-sentence pairs. By leveraging relative relationship, i.e., "a man is riding a motorcycle" is more accurate than "a man is riding a car" to describe the given video, we could allow access to more human supervision and thus generate more informative sentence.

How can we learn relative properties for video captioning? In this paper, we present a novel Long Short-Term Memory with Listwise Supervision (LSTM-LS) architecture, as shown in Figure 2. Specifically, given a video, a 2-D and/or 3-D Convolutional Neural Networks (CNN) is utilized to extract visual features of selected video frames/clips, while video representation is produced by mean pooling over these visual features. For each sentence associated with the given video, a sentence list is obtained by ranking all the sentences in terms of textual similarity in between and the list is regarded as a ground-truth ranking list. Then, a LSTM for generating video sentence is learnt and a probability score is produced for each sentence in the list. We formulate the learning of the LSTM model as an optimization problem, in which the objective is to minimize the difference between the ground-truth ranking list and the ranking derived from the probability scores of sentences. As such, the relative strength of different sentences relevant to the given video could be estimated and integrated into the sentence generation model.

The main contribution of this work is the proposal of LSTM-LS framework by incorporating semantic relationships among all sentences for boosting video captioning. This issue also leads to a view of how to model and exploit the relative relationships of different video-sentence pairs for sentence generation, which is not yet fully explored in the literature.

Related Work

There are generally two categories of methods for video captioning: template-based models (Kojima, Tamura, and Fukunaga 2002; Rohrbach et al. 2013; 2014; Guadarrama et al. 2013; Xu et al. 2015) and sequence learning models (e.g., RNN) (Donahue et al. 2015; Pan et al. 2016a; Venugopalan et al. 2015a; Yao et al. 2015; Venugopalan et al. 2015b; Pan et al. 2016b). The former predefines the special rule for language grammar and then parses the sentence into several parts (e.g., subject, verb, object). With such sentence fragments, the bulk of works associate each part with detected words from visual content by object recognition and then generate a sentence with the templates. The latter is to utilize sequence learning model as a decoder to directly generate sentence conditioned on video content.

Template-based Model

Most works in this direction primarily rely on the predefined templates of sentence and always generate sentence with syntactical structure. For example, (Kojima, Tamura, and Fukunaga 2002) builds a concept hierarchy of actions for natural language description of human activities. CRF is leveraged in (Rohrbach et al. 2013) to model the relationships between different components of the input video for video captioning. Furthermore, by incorporating semantic unaries and hand-centric features, Rohrbach *et al.* utilize CRF-based approach to generate coherent video description (Rohrbach et al. 2014). In (Guadarrama et al. 2013), Guadarrama *et al.* utilize semantic hierarchies to choose an appropriate level of the specificity and accuracy of sentence fragments. Recently, Xu *et al.* design a unified framework in (Xu et al. 2015), which consists of a compositional semantics language model, a deep video model and an embedding model to capture the joint video-language relationship for video sentence generation.

Sequence Learning Model

Different from template-based models, sequence learning methods utilize RNN decoder to generate novel sentence with more flexible syntactical structure. Donahua *et al.* employ a CRF to predict activity, object, and location from the input video and then concatenate them into an input sequence, followed by LSTM model for sentence generation (Donahue et al. 2015). Later in (Venugopalan et al. 2015b), an end-to-end LSTM-based system is designed to generate video descriptions with the input sequence of frames. The framework is then extended by inputting both frames and optical flow images into an encoder-decoder LSTM in (Venugopalan et al. 2015a). Furthermore, Pan *et al.* additionally consider the relevance between sentence semantics and video content as a regularizer in LSTM based architecture (Pan et al. 2016a). Unlike the method of producing video representations by mean pooling the visual features over all frames in (Venugopalan et al. 2015b), Yao *et al.* propose to utilize the temporal attention mechanism to exploit temporal structure for video captioning (Yao et al. 2015). Most recently, in (Pan et al. 2016b), high-level semantic attributes are shown to be complementary knowledge of video representations for enhancing video captioning when injected into existing RNN-based sequence learning models.

In summary, our work belongs to sequence learning model. Different from these pervious models which independently utilize video-sentence pairs for training, our approach contributes by guiding our sequence learning model with listwise supervision derived from video and its corresponding sentence ranking lists.

Video Captioning with Listwise Supervision

The main goal of our Long Short-Term Memory with Listwise Supervision (LSTM-LS) is to guide the learning of LSTM for sequence modeling in video captioning with listwise supervision. The training of LSTM-LS is performed by minimizing the difference between the ranking list generated from the sentences pool and the ranking produced by the log probabilities of corresponding sentences given the same target video. The approach overview is shown in Figure 2. In the following, we will first define the representation of video and the sequential words in sentence respectively, and the natural semantic ranking list for sentence, followed by sequence modeling in video captioning. Then the triplet representation for ground-truth sentence ranking list is provided. Finally, the overall objective and training strategy of LSTM-LS are presented.

Notation

Suppose we have a video $\mathcal{V}$ with N_v sample frames/clips (uniform sampling) to be described by a textual sentence $\mathcal{S}$, where $\mathcal{S} = \{w_1, w_2, ..., w_{N_s}\}$ consisting of N_s words. In order to effectively represent the visual content of a video, we first use a 2-D and/or 3-D CNN, which is powerful to produce a rich representation of each sampled frame/clip from the video. Then, we perform "mean pooling" process over all the frames/clips to generate a single D_v-dimensional vector $\mathbf{v}$ for each video $\mathcal{V}$. As a sentence consists of a sequence of words, a sentence can be represented by a $D_w \times N_s$ matrix $\mathbf{W} \equiv [\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_{N_s}]$, with each word in the sentence as its column vector. Furthermore, we denote another feature vector $\mathbf{s}$ for representing a sentence as a whole, which

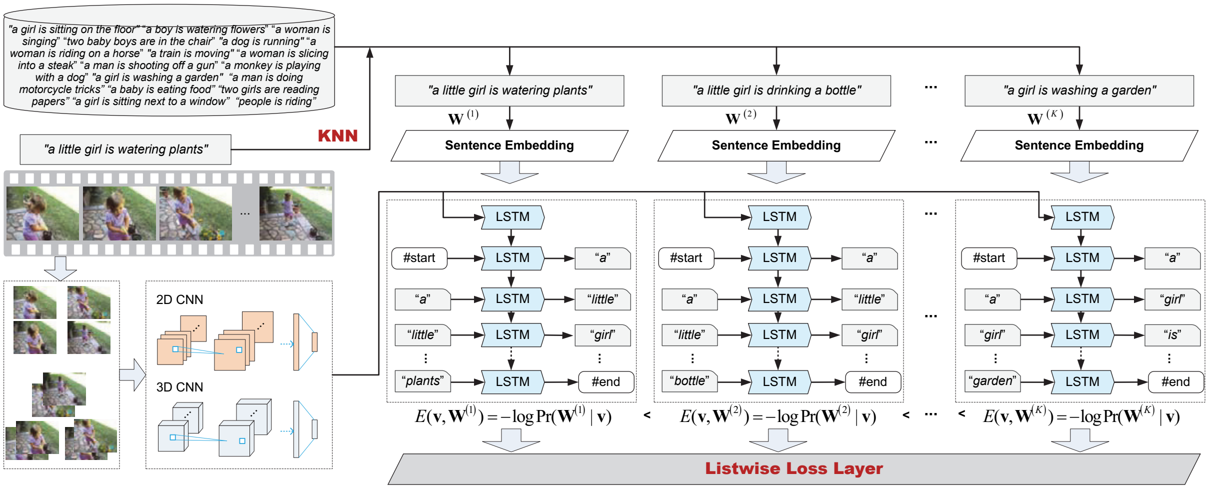

"a girl is sitting on the floor" "a boy is watering flowers" "a woman is singing" "two baby boys are in the chair" "a dog is running" "a woman is riding on a horse" "a train is moving" "a woman is slicing into a steak" "a man is shooting off a gun" "a monkey is playing with a dog" "a girl is washing a garden" "a man is doing motorcycle tricks" "a baby is eating food" "two girls are reading papers" "a girl is sitting next to a window" "people is riding"
"a little girl is watering plants"
KNN
2D CNN
3D CNN
"a little girl is watering plants"
"a little girl is drinking a bottle"
"a girl is washing a garden"
W(1)
W(2)
W(K)
Sentence Embedding
LSTM
#start
"a"
"little"
"girl"
"plants"
"bottle"
"garden"
"is"
#end
E(v, W(1)) = −log Pr(W(1) | v) < E(v, W(2)) = −log Pr(W(2) | v) < … < E(v, W(K)) = −log Pr(W(K) | v)
Listwise Loss Layer

Figure 2: The overview of Long Short-Term Memory with Listwise Supervision (LSTM-LS) for video captioning.

is produced by the feature vectors $\mathbf{w}_t$ $(t = 1, 2, ..., N_s)$ of each word in the sentence. We first encode each word w_t as "one-hot" vector (binary index vector in a vocabulary), thus the dimension of feature vector $\mathbf{w}_t$, i.e. D_w, is the vocabulary size. Then the binary TF weights are calculated over all words of the sentence to produce the integrated representation of the entire sentence, denoted by $\mathbf{s} \in \mathbb{R}^{D_w}$, with the same dimension as $\mathbf{w}_t$.

For any specific video $\mathcal{V}$, we can get the ranking list of sentences $\mathcal{L} = \left\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(K)}\right\}$ by performing nearest neighbor search on the sentence representations between the video's corresponding ground truth sentence $\mathbf{W}$ and sentences from sentence pool, where K is the number of returned top semantically similar sentences for $\mathbf{W}$. Let $d\left(\mathbf{W}, \mathcal{L}\right) = \{d_1, d_2, \ldots, d_K\}$ denote the list of Euclidean distances for the associated sentences. If $d_i < d_j$, it indicates that sentence $\mathbf{W}^{(i)}$ is more semantically relevant to sentence $\mathbf{W}$ than sentence $\mathbf{W}^{(j)}$ and vice versa.

Sequence Modeling in Video Captioning

Inspired by the recent successes of probabilistic sequence models leveraged in machine translation (Bahdanau, Cho, and Bengio 2015; Sutskever, Vinyals, and Le 2014), we aim to formulate our video captioning model in an end-to-end fashion based on RNN which first encodes the given video into a fixed dimensional vector and then decodes it to the target output sentence, which consists of sequential words. Hence, given the video, the problem about sequence modeling for target sentence we exploit here can be formulated by minimizing the following energy loss function:

$$E(\mathbf{v}, \mathbf{W}) = -\log \Pr\left(\mathbf{W}|\mathbf{v}\right), \tag{1}$$

which is the negative log probability of the correct textual sentence given the video. As the model produces the sentence word by word, it is natural to apply chain rule to model the joint probability over the sequential words. Thus, the log probability of the sentence is given by the sum of the log probabilities over the word:

$$\log \Pr\left(\mathbf{W}|\mathbf{v}\right) = \sum_{t=1}^{N_s} \log \Pr\left(\mathbf{w}_t|\,\mathbf{v}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1}\right). \tag{2}$$

By minimizing this loss, the contextual relationship among the words in the sentence can be guaranteed given the video.

As our video captioning task is formulated as a variable-length sequence to sequence problem, we naturally model the parametric distribution $\Pr\left(\mathbf{w}_t|\,\mathbf{v}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1}\right)$ in Eq.(2) with Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997). The vector formulas for a LSTM layer forward pass are given below. For timestep t, $\mathbf{x}^t$ and $\mathbf{h}^t$ are the input and output vector respectively, $\mathbf{T}$ are input weights matrices, $\mathbf{R}$ are recurrent weight matrices and $\mathbf{b}$ are bias vectors. Sigmoid σ and hyperbolic tangent ϕ are element-wise non-linear activation functions. The dot product of two vectors is denoted with $\odot$. Given inputs $\mathbf{x}^t$, $\mathbf{h}^{t-1}$ and $\mathbf{c}^{t-1}$, the LSTM unit updates for timestep t are:

$$\mathbf{g}^t = \phi(\mathbf{T}_g\mathbf{x}^t + \mathbf{R}_g\mathbf{h}^{t-1} + \mathbf{b}_g),\ \mathbf{i}^t = \sigma(\mathbf{T}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{h}^{t-1} + \mathbf{b}_i),$$
$$\mathbf{f}^t = \sigma(\mathbf{T}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{h}^{t-1} + \mathbf{b}_f),\ \mathbf{c}^t = \mathbf{g}^t \odot \mathbf{i}^t + \mathbf{c}^{t-1} \odot \mathbf{f}^t,$$
$$\mathbf{o}^t = \sigma(\mathbf{T}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{h}^{t-1} + \mathbf{b}_o),\ \mathbf{h}^t = \phi(\mathbf{c}^t) \odot \mathbf{o}^t,$$

where $\mathbf{g}^t$, $\mathbf{i}^t$, $\mathbf{f}^t$, $\mathbf{c}^t$, $\mathbf{o}^t$, and $\mathbf{h}^t$ are cell input, input gate, forget gate, cell state, output gate, and cell output of the LSTM.

As mentioned above, the LSTM model is utilized to predict each word in the sentence given the video content and previous words. We inject the embedded video representation at the initial time to inform the whole memory cells in LSTM about the visual content. Given the video $\mathbf{v}$ and the corresponding sentence $\mathbf{W} \equiv [\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_{N_s}]$, the LSTM updating procedure is as following:

$$\mathbf{x}^{-1} = \mathbf{T}_v\mathbf{v}, \tag{3}$$

$$\mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t, t \in \{0, \ldots, N_s - 1\}, \tag{4}$$

$$\mathbf{h}^t = f\left(\mathbf{x}^t\right), t \in \{0, \ldots, N_s - 1\}, \tag{5}$$

where D_e is the dimensionality of LSTM input, and $\mathbf{T}_v \in \mathbb{R}^{D_e \times D_v}$ and $\mathbf{T}_s \in \mathbb{R}^{D_e \times D_w}$ are the transformation matrices for video representation and textual feature of word, respectively, and f is the updating function within LSTM unit. Please note that for the input sentence $\mathbf{W} \equiv [\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_{N_s}]$, we take $\mathbf{w}_0$ as the start sign word to inform the beginning of sentence and $\mathbf{w}_{N_s}$ as the end sign word which indicates the end of sentence, both of the special sign words are included in our vocabulary. Most specifically, at the initial time step, the video representation is transformed as the input for LSTM, and then in the next steps, word embedding $\mathbf{x}^t$ will be input into the LSTM along with the previous step's hidden state $\mathbf{h}^{t-1}$. In each time step (except the initial step), we use the LSTM cell output $\mathbf{h}^t$ to predict the next word. Here a softmax layer is applied after the LSTM layer to produce a probability distribution over all the D_w words in the vocabulary as

$$\Pr_{t+1}(w_{t+1}) = \frac{\exp\left\{\mathbf{T}_h^{(w_{t+1})}\mathbf{h}^t\right\}}{\sum\limits_{w \in \mathcal{W}} \exp\left\{\mathbf{T}_h^{(w)}\mathbf{h}^t\right\}}, \tag{6}$$

where $\mathcal{W}$ is the word vocabulary space and $\mathbf{T}_h^{(w)}$ is the parameter matrix in softmax layer.

Triplet Representation of Ranking List

Here we translate the sentence ranking list $\mathcal{L}$ generated from sentences pool into a set of ranking triplets, which can be represented as a ranking triplet matrix and fed into the sequence learning paradigm. Given the list of Euclidean distances $d(\mathbf{W}, \mathcal{L})$ for sentece $\mathbf{W}$, we use a ranking triplet $S\left(\mathbf{W}; \mathbf{W}^{(i)}, \mathbf{W}^{(j)}\right) \in \mathbb{R}$ to represent the listwise supervision, which is defined as

$$S\left(\mathbf{W}; \mathbf{W}^{(i)}, \mathbf{W}^{(j)}\right) = \begin{cases} 1, & d_i < d_j \\ 0, & d_i = d_j \\ -1, & d_i > d_j \end{cases} . \tag{7}$$

Hence, we can represent the ranking list as a ground-truth ranking triplet matrix $S_{(\mathbf{W})} \in \mathbb{R}^{K \times K}$ with its element $S_{(\mathbf{W})}(i,j) = S\left(\mathbf{W}; \mathbf{W}^{(i)}, \mathbf{W}^{(j)}\right)$.

LSTM with Listwise Supervision

Different from previous video captioning models which always model the LSTM learning with video-sentence pairs for training, our LSTM-LS architecture further incorporates listwise supervision into LSTM to better guide the LSTM learning with video and corresponding sentence ranking list. In the training stage, given the video $\mathbf{v}$ and its top K sentences ranking list $\mathcal{L} = \left\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \ldots, \mathbf{W}^{(K)}\right\}$, to represent sentences ranking list by rank triplets for each given video, the negative log probability of each sentence given the video in Eq.(1) is directly utilized to measure the similarity of video and sentence. Accordingly, we can get the LSTM-based ranking list $\mathcal{L}_E = \left\{E(\mathbf{v}, \mathbf{W}^{(1)}), E(\mathbf{v}, \mathbf{W}^{(2)}), \ldots, E(\mathbf{v}, \mathbf{W}^{(K)})\right\}$. Here we again use triplets to assess the quality of the ranking list $\mathcal{L}_E$ based on video-sentence similarity measured by LSTM. If sentence $\mathbf{W}^{(i)}$ is ranked higher than sentence $\mathbf{W}^{(j)}$, indicated by the ground-truth ranking triplet as $S\left(\mathbf{W}; \mathbf{W}^{(i)}, \mathbf{W}^{(j)}\right) = 1$, it is expected that the LSTM-based video-sentence similarity should satisfy $E(\mathbf{v}, \mathbf{W}^{(i)}) < E(\mathbf{v}, \mathbf{W}^{(j)})$, otherwise $E(\mathbf{v}, \mathbf{W}^{(i)}) > E(\mathbf{v}, \mathbf{W}^{(j)})$. Hence, we can compute the ranking triplet for the LSTM-based ranking list $\mathcal{L}_E$ as

$$\widetilde{S}_{(\mathbf{v})}(i,j) = -sgn\left(E(\mathbf{v}, \mathbf{W}^{(i)}) - E(\mathbf{v}, \mathbf{W}^{(j)})\right). \tag{8}$$

The objective function is designed to measure the quality of the LSTM-based ranking list $\mathcal{L}_E$ by

$$\min_{\mathbf{T}_v, \mathbf{T}_s, \mathbf{T}_h, \theta} - \sum_{i,j=1}^{K} \widetilde{S}_{(\mathbf{v})}(i,j)\, S_{(\mathbf{v})}(i,j), \tag{9}$$

where θ are the parameters of LSTM, $\widetilde{S}_{(\mathbf{v})}(i,j)$ is the LSTM-based ranking triplet and $S_{(\mathbf{v})}(i,j)$ is the ground-truth ranking triplet.

Please note that the non-differentiable terms (i.e., $sgn(\bullet)$) in Eq.(8) makes the optimization problem difficult to be solved. To address this problem, we relax the ranking triplet for the LSTM-based ranking list by replacing the signum function in Eq.(8) with its signed magnitude as

$$\widetilde{S}_{(\mathbf{v})}(i,j) \approx -\left(E(\mathbf{v}, \mathbf{W}^{(i)}) - E(\mathbf{v}, \mathbf{W}^{(j)})\right). \tag{10}$$

Let N denote the number of video and its corresponding sentences list in the training set, the overall objective function can be written as

$$\begin{array}{c} \min\limits_{\mathbf{T}_v, \mathbf{T}_s, \mathbf{T}_h, \theta} -\frac{1}{N} \sum\limits_{m=1}^{N} \sum\limits_{i,j=1}^{K} \widetilde{S}_{(\mathbf{v}^{(\mathbf{m})})}(i,j)\, S_{(\mathbf{v}^{(\mathbf{m})})}(i,j) \\ + \|\mathbf{T}_v\|_2^2 + \|\mathbf{T}_s\|_2^2 + \|\mathbf{T}_h\|_2^2 + \|\theta\|_2^2 \end{array}, \tag{11}$$

where the first term is the overall loss for listwise supervision, while the rest are regularization terms for video embedding, sentence embedding, softmax layer and LSTM.

To solve the optimization according to overall loss objective in Eq.(11), we design a listwise loss layer on the top of each LSTM with shared parameters for video and its specific sentences from sentence ranking list, which does not have any parameter. During training, this listwise loss layer evaluates the model's violation of listwise supervision information, and back-propagates the gradients to LSTM so that LSTM and the lower layers can adjust their parameters to minimize the overall loss.

Experiments

We evaluate and compare our proposed LSTM-LS with state-of-the-art approaches by conducting video captioning on two benchmarks, i.e., Microsoft Research Video Description Corpus (MSVD) (Chen and Dolan 2011) and Montreal Video Annotation Dataset (M-VAD) (Torabi et al. 2015).

Datasets and Settings

MSVD. MSVD contains 1,970 video snippets collected from YouTube. There are roughly 40 available English descriptions per video. In our experiments, we follow the setting used in prior works (Guadarrama et al. 2013; Pan et al.

Table 1: BLEU@N and METEOR scores on the MSVD dataset. All values are reported as percentage (%).

Model	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR
LSTM (Venugopalan et al. 2015b)	-	-	-	33.3	29.1
SA (Yao et al. 2015)	80.0	64.7	52.6	41.9	29.6
S2VT (Venugopalan et al. 2015a)	-	-	-	-	29.8
LSTM-E (Pan et al. 2016a)	78.8	66.0	55.4	45.3	31.0
LSTM-LS (VGG)	78.1	65.9	56.4	46.5	31.2
LSTM-LS (C3D)	79.2	66.3	56.7	46.9	31.4
LSTM-LS (VGG+C3D)	80.2	69.0	60.1	51.1	32.6

2016a), taking 1,200 videos for training, 100 for validation and 670 for testing.

M-VAD. M-VAD is a recent collection of large-scale movie description dataset. It is composed of about 49,000 DVD movie snippets, which are extracted from 92 DVD movies. Each movie clip is accompanied with single sentence from semi-automatically transcribed descriptive video service (DVS) narrations.

Settings. In the experiment, we compare our LSTM-LS approach with one 2-D CNN of 19-layer VGG (Simonyan and Zisserman 2015) network pre-trained on Imagenet ILSVRC12 dataset (Russakovsky et al. 2015), and one 3-D CNN of C3D (Tran et al. 2015) pre-trained on Sports-1M video dataset (Karpathy et al. 2014). Specifically, we take the output of 4096-way fc6 layer from the 19-layer VGG and 4096-way fc6 layer from C3D as the frame and clip representation, respectively. The size of hidden layer in LSTM is set to 1,024. The number of nearest sentences K is empirically set to 4.

Compared Methods

To fully evaluate our model, we compare our LSTM-LS model with the following non-trivial baseline methods.

(1) Long Short-Term Memory (LSTM) (Venugopalan et al. 2015b): LSTM attempts to directly translate from video pixels to natural language with a single deep neural network including both convolutional and recurrent structure. The video representation is firstly generated by performing mean pooling over the frame features across the entire video and then injected into LSTM every timestep.

(2) Soft-Attention (SA) (Yao et al. 2015): SA combines frame representation from GoogleNet (Szegedy et al. 2015) and video clip representation based on a 3-D CNN trained on Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF) and Motion Boundary Histogram (MBH) hand-crafted descriptors. Furthermore, a weighted attention mechanism is used to dynamically attend to specific temporal segments of the video while generating sentence.

(3) Sequence to Sequence - Video to Text (S2VT) (Venugopalan et al. 2015a): S2VT incorporates both RGB and optical flow inputs. The encoding of inputs and decoding of each word in the description are jointly learnt in parallel.

(4) Long Short-Term Memory with visual-semantic Embedding (LSTM-E) (Pan et al. 2016a): LSTM-E utilizes both 2-D CNN and 3-D CNN to learn an effective spatio-temporal video representation, and jointly explores the learning of LSTM and visual-semantic embedding for video captioning.

Table 2: METEOR scores (%) on M-VAD dataset.

Model	METEOR
SA (Yao et al. 2015)	4.3
LSTM (Venugopalan et al. 2015b)	6.1
S2VT (Venugopalan et al. 2015a)	6.7
LSTM-E (Pan et al. 2016a)	6.7
LSTM-LS (VGG+C3D)	6.9

(5) Long Short-Term Memory with Listwise Supervision (LSTM-LS): We design three runs for our proposed approach, i.e., LSTM-LS (VGG), LSTM-LS (C3D), and LSTM-LS (VGG+C3D). The input frame/clip features of the first two runs are from VGG and C3D network respectively. The input of the last one is to concatenate the features from VGG and C3D.

Performance Comparison

Table 1 shows the BLEU@N (Papineni et al. 2002) and METEOR (Banerjee and Lavie 2005) performance of all runs on MSVD dataset. Overall, our proposed LSTM-LS outperforms the other runs. Specifically, LSTM-LS (C3D) outperforms LSTM-LS (VGG) and reach 31.4% METEOR. When combining with VGG, LSTM-LS (VGG+C3D) further improves the performance to 32.6%, which makes the relative improvement over the two state-of-the-art methods S2VT by 9.3% and LSTM-E by 5.2%, respectively.

There is a performance gap among three runs LSTM, SA and S2VT. Though three runs are all purely based on MSVD dataset, they are fundamentally different in the way of modeling temporal structure in the video. The performance of LSTM and SA is as a result of linearly fusing visual representations of video frames by mean pooling and soft attention respectively, while S2VT is by encoding the video frames in a sequential manner. As indicated by our results, the strategy of sequence to sequence learning performs better. Compared to LSTM, LSTM-E which additionally incorporates the relevance between sentence semantics and video content as a regularizer in LSTM exhibits significantly better performance. Furthermore, LSTM-LS performs consistently better than LSTM-E, which verifies the advantage of exploring semantic relationships holistically between all sentences and the video than in the video-sentence pairwise manner.

The METEOR scores on M-VAD are given in Table 2. Our LSTM-LS (VGG+C3D) approach consistently outperforms the state-of-the-art methods on the movie dataset. In particular, the METEOR scores of LSTM-LS (VGG+C3D)

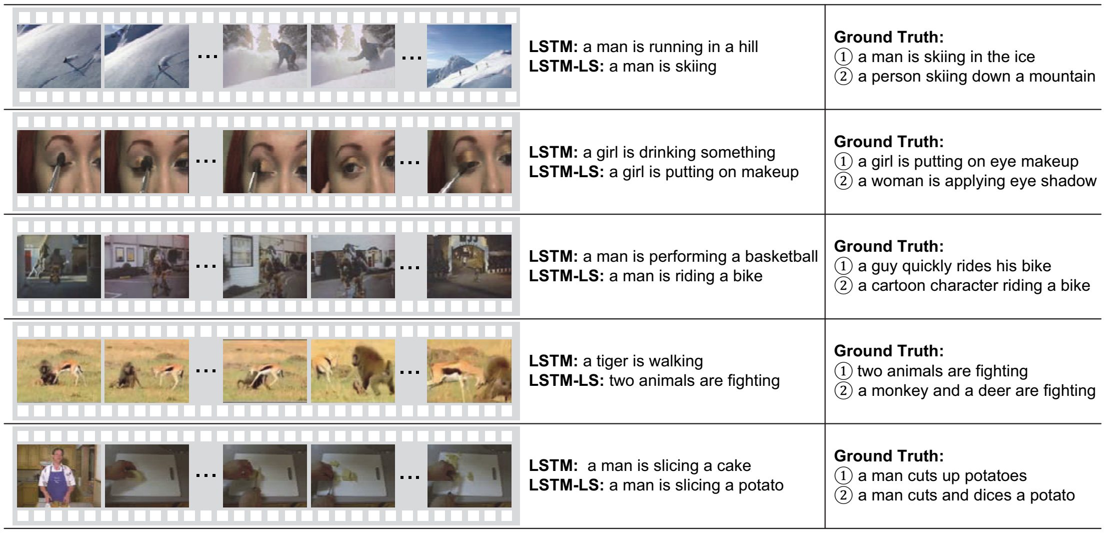

LSTM: a man is running in a hill
LSTM-LS: a man is skiing
Ground Truth:
① a man is skiing in the ice
② a person skiing down a mountain
LSTM: a girl is drinking something
LSTM-LS: a girl is putting on makeup
Ground Truth:
① a girl is putting on eye makeup
② a woman is applying eye shadow
LSTM: a man is performing a basketball
LSTM-LS: a man is riding a bike
Ground Truth:
① a guy quickly rides his bike
② a cartoon character riding a bike
LSTM: a tiger is walking
LSTM-LS: two animals are fighting
Ground Truth:
① two animals are fighting
② a monkey and a deer are fighting
LSTM: a man is slicing a cake
LSTM-LS: a man is slicing a potato
Ground Truth:
① a man cuts up potatoes
② a man cuts and dices a potato

Figure 3: Examples of sentence generation results on MSVD. The videos are represented by sampled frames, the output sentences generated by 1) LSTM, 2) our proposed LSTM-LS, and 3) Ground Truth: Randomly selected two ground truth sentences.

Table 3: The effect of the number of nearest sentences K in our LSTM-LS framework on MSVD dataset.

k	BLEU@4	METEOR
2	49.7	32.3
3	50.0	32.3
4	51.1	32.6
5	49.8	32.4

can achieve 0.069, which is so far the highest performance reported on M-VAD dataset.

Figure 3 shows a few sentence examples generated by different methods and human-annotated ground truth (GT). From these exemplar results, it is easy to see that all of these automatic methods can generate somewhat relevant sentences. When looking into each word, our LSTM-LS predict more relevant Subject, Verb and Object (SVO) terms. For example, compared to verb term "running," "skiing" is more precise to describe the video content in the first video. Similarly, the predicted object term "potato" is more relevant than "cake" in fifth video.

Effect of the Number of Nearest Sentences

The number of nearest sentences is an important parameter for modeling listwise supervision. In the previous experiments, the number is fixed to 4. Next, we conduct experiments to evaluate the performances of LSTM-LS on MSVD dataset with the number of nearest sentences in the range of {2, 3, 4, 5}. The BLEU@4 and METEOR of LSTM-LS with different number of nearest sentences are shown in Table 3. As illustrated in the table, the optimal K is happened at 4 for LSTM-LS on MSVD dataset. This is also expected, as for few nearest sentences may not be enough to represent the relative relationships between sentences and the video while too many sentences will include noisy ones.

Table 4: The effect of hidden layer size in our LSTM-LS framework on MSVD dataset.

Hidden layer size	BLEU@4	METEOR	Parameter number
128	45.4	31.4	3.6M
256	46.9	31.6	7.5M
512	48.7	31.9	16.0M
1024	51.1	32.6	36.2M

The Size of Hidden Layer of LSTM

In order to show the relationship between the performance and hidden layer size of LSTM, we compare the results of the hidden layer size in the range of 128, 256, 512 and 1024. The results shown in Table 4 indicate increasing the hidden layer size can generally lead to performance improvements. Considering that the number of parameters increases exponentially at the meantime, the hidden layer size is empirically set to 1,024 and no further increased in our experiments.

Conclusions

In this paper, we have proposed a novel model named LSTM-LS to generate video description. In particular, listwise supervision from sentence lists generated for each video is additionally incorporated into LSTM learning. As such, the semantic relationships between all sentences and each video are holistically measured. On the popular MSVD and M-VAD datasets, the results of our experiments demonstrate the success of our approach, outperforming the current state-of-the-art models with a clear margin on sentence generation. Our future works are as follows. First, as a video is a temporal sequence, we will further explore the way of utilizing RNN to better represent the videos. Second, how to leverage largely available image captioning data in our framework for boosting video captioning is worth trying.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Banerjee, S., and Lavie, A. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.

Chen, D. L., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*.

Donahue, J.; Hendricks, L. A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Guadarrama, S.; Krishnamoorthy, N.; Malkarnenkar, G.; Venugopalan, S.; Mooney, R.; Darrell, T.; and Saenko, K. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

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

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

Kojima, A.; Tamura, T.; and Fukunaga, K. 2002. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2016a. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

Pan, Y.; Yao, T.; Li, H.; and Mei, T. 2016b. Video captioning with transferred semantic attributes. *arXiv preprint arXiv:1611.07675*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Rohrbach, M.; Qiu, W.; Titov, I.; Thater, S.; Pinkal, M.; and Schiele, B. 2013. Translating video content to natural language descriptions. In *ICCV*.

Rohrbach, A.; Rohrbach, M.; Qiu, W.; Friedrich, A.; Pinkal, M.; and Schiele, B. 2014. Coherent multi-sentence video description with variable level of detail. In *GCPR*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV*.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*.

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

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

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

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; and Saenko, K. 2015b. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT*.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Xu, R.; Xiong, C.; Chen, W.; and Corso, J. J. 2015. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*.

Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C.; Larochelle, H.; and Courville, A. 2015. Describing videos by exploiting temporal structure. In *ICCV*.

Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; and Mei, T. 2016. Boosting image captioning with attributes. *arXiv preprint arXiv:1611.01646*.