Jointly Modeling Embedding and Translation to Bridge Video and Language

Yingwei Pan [‡], Tao Mei [†], Ting Yao [†], Houqiang Li [‡], and Yong Rui [†]

[†] Microsoft Research, Beijing, China

[‡] University of Science and Technology of China, Hefei, China

`panyw.ustc@gmail.com, {tmei, tiyao, yongrui}@microsoft.com, lihq@ustc.edu.cn`

Abstract

Automatically describing video content with natural language is a fundamental challenge of multimedia. Recurrent Neural Networks (RNN), which models sequence dynamics, has attracted increasing attention on visual interpretation. However, most existing approaches generate a word <u>locally with given previous words and the visual content</u>, while the relationship between sentence semantics and visual content is not holistically exploited. <u>As a result, the generated sentences may be contextually correct but the semantics (e.g., subjects, verbs or objects) are not true.</u>

This paper presents a novel unified framework, named <u>Long Short-Term Memory with visual-semantic Embedding (LSTM-E)</u>, which can simultaneously explore the <u>learning of LSTM</u> and <u>visual-semantic embedding</u>. The former aims to locally maximize the probability of generating the next word given previous words and visual content, while the latter is to create a visual-semantic embedding space for enforcing the relationship between the semantics of the entire sentence and visual content. Our proposed LSTM-E consists of three components: a <u>2-D</u> and/or <u>3-D</u> deep convolutional neural networks for learning powerful video representation, a deep <u>RNN</u> for generating sentences, and a joint embedding model for exploring the relationships between visual content and sentence semantics. The experiments on YouTube2Text dataset show that our proposed LSTM-E achieves to-date the best reported performance in generating natural sentences: 45.3% and 31.0% in terms of BLEU@4 and METEOR, respectively. We also demonstrate that LSTM-E is superior in predicting <u>Subject-Verb-Object (SVO)</u> triplets to several state-of-the-art techniques.

1. Introduction

Video has become ubiquitous on the Internet, broadcasting channels, as well as personal devices. This has encouraged the development of advanced techniques to analyze the semantic video content for a wide variety of applications. Recognition of videos has been a fundamental

Input Video:
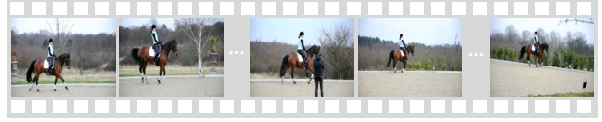
Output Sentence:
- **LSTM:** a man is riding a horse.
- **LSTM-E:** a woman is riding a horse.
- **Humans:** a woman gallops on a horse. / a woman is riding a horse along a road. / the girl rode her brown horse.

Figure 1. Examples of video description generation. Input: a short video. Output: a natural language sentence describing the main content of the input video.

challenge of multimedia for decades. Previous research has predominantly focused on recognizing videos with a predefined yet very limited set of individual words. Thanks to the recent development of Recurrent Neural Networks (RNN), researchers have strived to automatically describe video content with a complete and natural sentence, which can be regarded as the ultimate goal of video understanding.

Figure 1 shows the examples of video description generation. Given an input video, the generated sentences are to describe video content, ideally encapsulating its most informative dynamics. There is a wide variety of video applications based on the description, ranging from editing, indexing, search, to sharing. However, the problem itself has been taken as a ground challenge for decades in the research communities, as the description generation model should be powerful enough not only to <u>recognize key objects from visual content</u>, but also <u>discover their spatio-temporal relationships and the dynamics expressed in a natural language</u> as well.

Despite the difficulty of the problem, there have been a few attempts to address video description generation [6, 32, 35], and image caption generation [7, 14, 17, 20, 33], which are mainly inspired by recent advances in machine translation using Recurrent Neural Networks (RNN) [1, 27]. The standard RNN is a nonlinear dynamical system that maps sequences to sequences. Although the gradients of the RNN are easy to compute, RNN models are difficult to

train, especially when the problems have long-range temporal dependencies, due to the well-known "vanishing gradient" effect [4, 21]. As such, the Long Short-Term Memory (LSTM) model was proposed to overcome the vanishing gradients problem by incorporating memory units, which allow the network to learn when to forget previous hidden states and when to update hidden states [12]. LSTM has been successfully adopted to several tasks, e.g., speech recognition [9], language translation [1] and image caption [20, 33]. Thus, we follow this elegant recipe and use LSTM as our RNN model to generate the video sentence in this paper.

Moreover, existing video description generation approaches mainly optimize the next word given the input video and previous words locally, while leaving the relationship between the semantics of the entire sentence and video content unexploited. As a result, the generated sentences can suffer from robustness problem. It is often the case that the output sentence from existing approaches may be contextually correct but the semantics (e.g., subjects, verbs or objects) in the sentence are not true. For example, the sentence generated by LSTM-based model for the video in Figure 1 is "a man is riding a horse," which is correct in logic but the subject "man" is not relevant to the video content.

To address the above issues, we leverage the semantics of the entire sentence and visual content to learn a visual-semantic embedding model, which holistically explores the relationships in between. Specifically, we present a novel Long Short-Term Memory with visual-semantic Embedding (LSTM-E) framework to bridge video content and natural language, as shown in Figure 2. Given a video, a 2-D and/or 3-D Convolution Neural Networks (CNN) is utilized to extract visual features of selected video frames/clips, while the video representation is produced by mean pooling over these visual features. Then, a LSTM for generating video sentence and a visual-semantic embedding model are jointly learnt based on the video representation and sentence semantics. The spirit of LSTM-E is to generate video sentence from the viewpoint of mutual reinforcement between *coherence* and *relevance*. *Coherence* expresses the contextual relationships among the generated words with video content which is optimized in LSTM, while *relevance* conveys the relationship between the semantics of the entire sentence and video content which is measured in the visual-semantic embedding.

In summary, this paper makes the following contributions:

- We present an end-to-end deep model for automatic video description generation, which incorporates both spatial and temporal structures underlying video.

- We propose a novel Long Shot-Term Memory with

visual-semantic Embedding (LSTM-E) framework, which considers both the contextual relationship among the words in sentence, and the relationship between the semantics of the entire sentence and video content, for generating natural language of a given video.

- The proposed model is evaluated on the popular Youtube2Text corpus and outperforms the-state-of-the-art in terms of both Subject-Verb-Object (SVO) triplet prediction and sentence generation.

The remaining parts of the paper are organized as follows. Section 2 reviews related work. Section 3 presents the problem of video description generation, while Section 4 details our solution of jointly modeling embedding and translation. In Section 5, we provide empirical evaluations, followed by the discussions and conclusions in Section 6.

2. RELATED WORK

There are mainly two directions for translation from visual content. The first direction predefines the special rule for language grammar and split sentence into several parts (e.g. subject, verb, object). With such sentence fragments, many works align each part with visual content and then generate the sentence for corresponding visual content: [19] use Conditional Random Field (CRF) model to produce sentence for image and in [8], a Markov Random Field (MRF) model is proposed to attach a descriptive sentence to the given image. For video translation, Rohrbach *et al.* [23] learn a CRF to model the relationships between different components of the input video and generate descriptions for video. Guadarrama *et al.* [11] use semantic hierarchies to choose an appropriate level of the specificity and accuracy of sentence fragments. This direction is highly depended on the templates of sentence and can only generate sentence with syntactical structure.

Another direction is to learn the probability distribution in the common space of visual content and textual sentence. In this direction, several works explore such probability distribution using topic models [3, 13] and neural networks [6, 16, 20, 31, 32, 33, 35]. They can generate sentence more flexibly. Most recently, several methods have been proposed for visual to sentence task based on the neural networks and most of them are utilizing the RNN due to its successful use in sequence to sequence learning for machine translation [1, 27]. Kiros *et al.* [16] firstly take the neural networks to generate sentence for image by proposing a image-text multimodal log-bilinear neural language model. In another work by Mao *et al.* [20], a multimodal Recurrent Neural Networks (m-RNN) model is proposed for image to caption, which directly models the probability of generating a word given previous words and image. In [33], Vinyals *et al.* propose an end-to-end neural networks
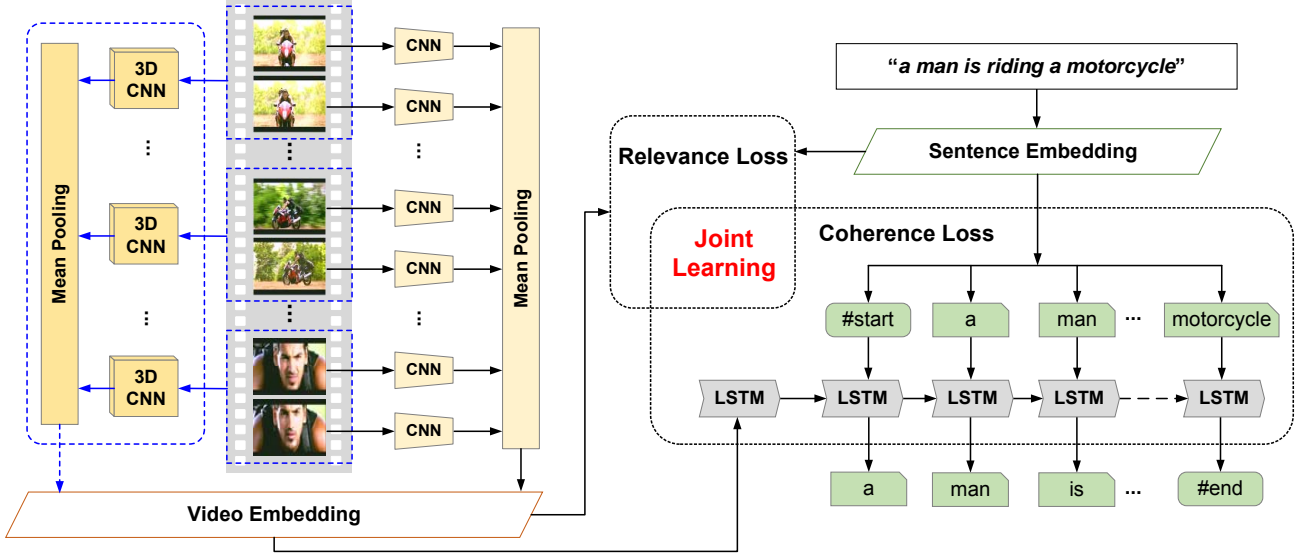
Figure 2. An overview of our LSTM-E framework with a language generating LSTM and a visual-semantic embedding model (better viewed in color). The video representation is produced by mean pooling over the visual features of frames/clips, extracted by a 2-D/3-D CNN. The *relevance loss* is to measure the relationships between the semantics of the entire sentence and video content in the embedding space, while the *coherence loss* is to characterize the contextual relationships among the generated words in the sentence in LSTM. Both LSTM and visual-semantic embedding are jointly learnt by minimizing the two losses.

system by utilizing LSTM to generate sentence for image. For video translation, an end-to-end LSTM based model is proposed in [32], which only reads the sequence of video frames and then generates a natural sentence. The model is further extended by inputting both frames and optical flow in [31]. Yao *et al.* propose to use a 3-D convolutional neural networks for modeling video clip dynamic temporal structure and an attention mechanism to select the most relevant temporal clips [35]. Then, the resulting video representations are fed into the text-generating RNN.

Our work belongs to the second direction. However, most of the above approaches in this direction mainly focus on optimizing the contextual relationship among words to generate sentence given visual content, while the relationship between the semantics of the entire sentence and visual content is not fully explored. Our work is different that we claim to generate video sentence by jointly exploiting the two relationships, which characterize the complementary properties of *coherence* and *relevance* of a generated sentence, respectively.

3. Video Description Generation

Our goal is to generate language sentences for videos. What makes a good sentence? Beyond describing important persons, objects, scenes, and actions by words, it must also convey how one word leads to the next. Specifically, we define a good sentence as a coherent chain of words in which each word influences the next through contextual information. Furthermore, the semantics of the entire sentence must

be relevant to the video content. We begin this Section by presenting the problem formulation, and followed by the proposal of two losses on measuring *coherence* and *relevance*, respectively.

3.1. Problem Formulation

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. Let $\mathbf{v} \in \mathbb{R}^{D_v}$ and $\mathbf{w}_t \in \mathbb{R}^{D_w}$ denote the D_v-dimensional visual features of a video $\mathcal{V}$ and the D_w-dimensional textual features of the t-th word in sentence $\mathcal{S}$, respectively. As a sentence consists of a sequence of words, a sentence can be represented by a $D_w \times N_s$ matrix $\mathbf{W} \equiv [\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_{N_s}}]$, with each word in the sentence as its column vector. Furthermore, we denote another feature vector $\mathbf{s}$ in the text space for representing a sentence as a whole.

In the video description generation problem, on one hand, the generated descriptive sentence must be able to depict the main contents of a video precisely, and on the other, the words in the sentence should be organized coherently in language. Therefore, we can formulate the video description generation problem by minimizing the following energy loss function

$$E(\mathcal{V}, \mathcal{S}) = (1 - \lambda) \times E_r(\mathbf{v}, \mathbf{s}) + \lambda \times E_c(\mathbf{v}, \mathbf{W}), \quad (1)$$

where $E_r(\mathbf{v}, \mathbf{s})$ and $E_c(\mathbf{v}, \mathbf{W})$ are the *relevance loss* and *coherence loss*, respectively. The former measures the relevance degree of the video content and sentence semantics

and we build an visual-semantic embedding for this purpose, which is introduced in Section 3.2. The latter estimates the contextual relationships among the generated words in the sentence and we use LSTM-based RNN as our model, which is presented in Section 3.3. The tradeoff between these two competing losses is captured by linear fusion with a positive parameter λ.

3.2. Visual-Semantic Embedding

In order to effectively represent the visual content of a video, we first use a 2-D and/or 3-D deep convolutional neural networks (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-dimension vector $\mathbf{v}$ for each video $\mathcal{V}$. The sentence feature $\mathbf{s}$ 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 of $\mathbf{w}_t$.

We assume that a low-dimensional embedding exists for the representation of video and sentence. The linear mapping function can be derived from this embedding by

$$\mathbf{v}_e = \mathbf{T}_v \mathbf{v} \text{ and } \mathbf{s}_e = \mathbf{T}_s \mathbf{s}, \tag{2}$$

where D_e is the dimensionality of the embedding, and $\mathbf{T}_v \in \mathbb{R}^{D_e \times D_v}$ and $\mathbf{T}_s \in \mathbb{R}^{D_e \times D_s}$ are the transformation matrices that project the video content and semantic sentence into the common embedding, respectively.

To measure the relevance between the video content and semantic sentence, one natural way is to compute the distance between their mappings in the embedding. Thus, we define the relevance loss as

$$E_r(\mathbf{v}, \mathbf{s}) = \|\mathbf{T}_v \mathbf{v} - \mathbf{T}_s \mathbf{s}\|_2^2. \tag{3}$$

We strengthen the relevance between video content and semantic sentence by minimizing the relevance loss. As such, the generated sentence is expected to better manifest the semantics of videos.

3.3. Translation by Sequence Learning

Inspired by the recent successes of probabilistic sequence models leveraged in statistical machine translation [6, 33], we define our coherence loss as

$$E_c(\mathbf{v}, \mathbf{W}) = -\log \Pr(\mathbf{W}|\mathbf{v}). \tag{4}$$

Assuming that a generative model of $\mathbf{W}$ that produces each word in the sequence in order, the $\log$ probability of

the sentence is given by the sum of the $\log$ probabilities over the word and can be expressed as:

$$\log \Pr(\mathbf{W}|\mathbf{v}) = \sum_{t=0}^{N_s} \log \Pr(\mathbf{w}_t | \mathbf{v}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1}). \tag{5}$$

By minimizing the coherence loss, the contextual relationship among the words in the sentence can be guaranteed, making the sentence coherent and smooth.

In video description generation task, both the relevance loss and coherence loss need to be estimated to complete the whole energy function. We will present a solution to jointly model the two losses in a deep recurrent neural networks in the next sections.

4. Joint Modeling Embedding and Translation

Following the *relevance* and *coherence* criteria, this work proposes a Long Short-Term Memory with visual-semantic Embedding (LSTM-E) model for video description generation. The basic idea of LSTM-E is to translate the video representation from a 2-D and/or 3-D deep convolutional network to the desired output sentence by using LSTM-type RNN model. Figure 2 shows an overview of LSTM-E model. In particular, the training of LSTM-E is performed by simultaneously minimizing the *relevance loss* and *coherence loss*. Therefore, the formulation presented in Eq.(1) is equivalent to minimizing the following energy function

$$
\begin{aligned}
E(\mathcal{V}, \mathcal{S}) = &(1 - \lambda) \times \|\mathbf{T}_v \mathbf{v} - \mathbf{T}_s \mathbf{s}\|_2^2 - \\
&\lambda \times \sum_{t=0}^{N_s} \log \Pr(\mathbf{w}_t | \mathbf{v}, \mathbf{w}_0, \ldots, \mathbf{w}_{t-1}; \theta; \mathbf{T}_v; \mathbf{T}_s)
\end{aligned}
\tag{6}
$$

where θ are the parameters of our LSTM-E models.

In the following, we will first present the architecture of LSTM memory cell, followed by jointly modeling with visual-semantic embedding.

4.1. Long Short Term Memory

We briefly introduce the standard Long Short-Term Memory (LSTM) [12], a variant of RNN, which can capture long-term temporal information by mapping input sequences to a sequence of hidden states and then hidden states to outputs. To address the vanishing gradients problem in traditional RNN training, LSTM incorporates a memory cell which can maintain its states over time and non-linear gating units which control the information flow into and out of the cell. As much light has been threw on LSTM recently, many improvements have been made to the LSTM architecture on its original formulation [12]. We adopt the LSTM architecture as described in [36], which omits the peephole connections in previous work [10].

A diagram of the LSTM unit can be seen in Figure 3. It consists of a single memory cell, an input activation function, an output activation function, and three gates (input, forget and output). The hidden state of the cell is recurrently connected back to the input and three gates. The memory cell updates its hidden state by combining the previous cell state which is modulated by the forget gate and a function of the current input and the previous output, modulated by the input gate. The forget gate is a critical component of the LSTM unit, which can control what to be remembered and what to be forgotten by the cell and somehow can avoid the gradient from vanishing or exploding when back propagating through time. Having been updated, the cell state is mapped to (-1,1) range through an output activation function which is necessary whenever the cell state is unbounded. Finally, the output gate determines how much of the memory cell flows into the output. These additions to the single memory cell enable LSTM to capture extremely complex and long-term temporal dynamics which is impossible for traditional RNN.

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. *Logic sigmoid* $\sigma(x) = \frac{1}{1+e^{-x}}$ and *hyperbolic tangent* $\phi(x) = \frac{e^x-e^{-x}}{e^x+e^{-x}}$ are element-wise non-linear activation functions, mapping real numbers to $(0,1)$ and $(-1,1)$ separately. The dot product and sum of two vectors are denoted with $\odot$ and $\oplus$, respectively. Given inputs $\mathbf{x}^t$, $\mathbf{h}^{t-1}$ and $\mathbf{c}^{t-1}$, the LSTM unit updates for timestep t are:

$$\mathbf{g}^t = \phi(\mathbf{T}_g\mathbf{x}^t + \mathbf{R}_g\mathbf{h}^{t-1} + \mathbf{b}_g) \qquad cell\ input$$

$$\mathbf{i}^t = \sigma(\mathbf{T}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{h}^{t-1} + \mathbf{b}_i) \qquad input\ gate$$

$$\mathbf{f}^t = \sigma(\mathbf{T}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{h}^{t-1} + \mathbf{b}_f) \qquad forgetgate$$

$$\mathbf{c}^t = \mathbf{g}^t \odot \mathbf{i}^t + \mathbf{c}^{t-1} \odot \mathbf{f}^t \qquad cell\ state$$

$$\mathbf{o}^t = \sigma(\mathbf{T}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{h}^{t-1} + \mathbf{b}_o) \qquad output\ gate$$

$$\mathbf{h}^t = \phi(\mathbf{c}^t) \odot \mathbf{o}^t \qquad cell\ output$$

4.2. LSTM with Visual-Semantic Embedding

By further incorporating a visual-semantic embedding, our LSTM-E architecture is to jointly model embedding and translation. In the training stage, given the video-sentence pair, the inputs of LSTM are the representations of the video and the words in the sentence after mapping into the embedding. As mentioned above, here we train the LSTM model to predict each word in the sentence given the embedding of visual feature for video and previous words. There are multiple ways that can be used to combine the visual content and words in LSTM unit updating procedure. The first one is to <u>feed the visual content at each time step</u> as an extra
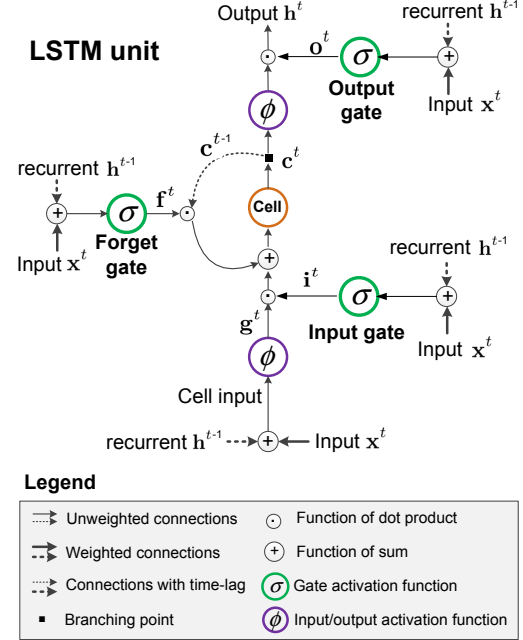

Figure 3. A diagram of an LSTM memory cell.

input for LSTM to emphasize the visual content frequently among LSTM memory cells. The second one only inputs the visual content once at the initial step to inform the whole memory cells in LSTM about the visual content. As empirically verified in [33], feeding the image at each time yields inferior results, due to the fact that the network can explicitly exploit noise and overfits more easily. Therefore, we adopt the second approach to arrange the inputs into LSTM in our architecture. Given the video $\mathbf{v}$ and its corresponding sentence $\mathbf{W} \equiv [\mathbf{w}_0, \mathbf{w}_1, ..., \mathbf{w}_{N_s}]$, the LSTM updating procedure is as following:

$$\mathbf{x}^{-1} = \mathbf{T}_v\mathbf{v} \tag{7}$$

$$\mathbf{x}^t = \mathbf{T}_s\mathbf{w}_t, t \in \{0, \ldots, N_s - 1\} \tag{8}$$

$$\mathbf{h}^t = f\left(\mathbf{x}^t\right), t \in \{0, \ldots, N_s - 1\} \tag{9}$$

where f is the updating function within LSTM unit. Please note that for the input sentence $\mathbf{W} \equiv \{\mathbf{w}_0, \ldots, \mathbf{w}_{N_s}\}$, 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 in the embedding is set 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_s words in

the vocabulary as

$$\Pr_{t+1}(w_{t+1}) = \frac{\exp\left(\mathbf{T}_h^{(w_{t+1})}\mathbf{h}^t\right)}{\sum\limits_{w\in\mathcal{W}}\exp\left(\mathbf{T}_h^{(w)}\mathbf{h}^t\right)}, \qquad (10)$$

where $\mathcal{W}$ is the word vocabulary space, $\mathbf{T}_h^{(w)}$ is the parameter matrix in softmax layer. Therefore, we can obtain the next word based on such probability distribution until the end sign word is emitted.

Accordingly, we define our loss function as follows:

$$E(\mathcal{V},\mathcal{S}) = (1-\lambda)\times\|\mathbf{T}_v\mathbf{v} - \mathbf{T}_s\mathbf{s}\|_2^2 - \\ \lambda\times\sum_{t=1}^{N_s}\log\Pr_t(\mathbf{w}_t) \qquad (11)$$

Let N denote the number of video-sentence pairs in the training dataset, we have the following optimization problem:

$$\min_{\mathbf{T}_v,\mathbf{T}_s,\mathbf{T}_h,\theta}\frac{1}{N}\sum_{i=1}^{N}\left(E(\mathcal{V}^{(i)},\mathcal{S}^{(i)})\right) \\ + \|\mathbf{T}_v\|_2^2 + \|\mathbf{T}_s\|_2^2 + \|\mathbf{T}_h\|_2^2 + \|\theta\|_2^2, \qquad (12)$$

where the first term is the combination of the *relevance loss* and *coherence loss*, while the rest are regularization terms for video embedding, sentence embedding, softmax layer and LSTM, respectively.

The above overall objective is optimized over the whole training video-sentence pairs using stochastic gradient descent. By minimizing this objective function, our LSTM-E model takes into account both the contextual relationships among the words in sentence (*coherence*) and the relationships between the semantics of the entire sentence and video content (*relevance*).

For sentence generation, there are two common strategies to translate the given video. The first approach is to sample the next word from the probability distribution at each timestep and set its representation in embedding space as the LSTM input for next timestep until the end sign word is sampled or the maximum sentence size is reached. Another method is select the top-k best sentence for each timestep and sets them as the candidates for next timestep based on which to generate new top-k best sentence. To make the generation process concise and efficient, we adopt the similar way as the latter one but set k as 1. Therefore, at each timestep, we choose the word with maximum probability as the predicted word and input its embedded feature in the next timestep until the model outputs the end sign word.

5. Experiments

In this section, we will first introduce our experimental setting. Then, the evaluation results compared with state-of-the-arts on two tasks, i.e., Subject-Verb-Object (SVO)

triplet prediction and natural sentence generation tasks, are reported. Finally, the effect of tradeoff parameter between *coherence* and *relevance* and the size of hidden layer in LSTM are presented.

5.1. Experimental Setting

We conduct our experiments mainly on the Microsoft Research Video Description Corpus (YouTube2Text) [5], which have been used in several prior works [11, 29, 34] on action recognition and video description generation tasks. This video corpus contains 1,970 YouTube snippets which cover a wide range of daily activities such as "people doing exercises," "playing music," and "cooking." We use the roughly 40 available English descriptions per video. In our experiments, following the setting used in prior works on video description generation [11, 34], we pick 1,200 videos to be used as training data, 100 videos for validation and 670 videos for testing.

We compare our LSTM-E architecture with two 2-D CNN of AlexNet [18] and the 19-layer VGG [25] network both pre-trained on Imagenet ILSVRC12 dataset [24], and one 3-D CNN of C3D [30] pre-trained on Sports-1M video dataset [15]. Specifically, we take the output of 4096-way fc7 layer from AlexNet, 4096-way fc6 layer from the 19-layer VGG, and 4096-way fc6 layer from C3D as the frame/clip representation, respectively. The dimensionality of the visual-semantic embedding space and the size of hidden layer in LSTM are both set to 512. The tradeoff parameter λ leveraging the relevance loss and coherence loss is empirically set to 0.7. The sensitivity of λ will be discussed in Section 5.3.1.

5.2. Performance Comparison

We empirically verify the merit of our LSTM-E model from two aspects: SVO triplet prediction and sentence generation for the video-language translation.

5.2.1 Compared Approaches

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

- Conditional Random Field (CRF) [34]: CRF model is developed to incorporate subject-verb and verb-object pairwise relationship based on the word pairwise co-occurrence statistics in the sentence pool.

- Canonical Correlation Analysis (CCA) [26]: CCA is to build the video-language joint space and generate the SVO triplet by k-nearest-neighbors search in the sentence pool.

- Factor Graph Model (FGM) [29]: FGM combines knowledge mined from text corpora with visual con-

Table 1. SVO accuracy: Binary accuracy of SVO triplet prediction. We extract SVO triplets from sentences output by LSTM and LSTM-E using a dependency parser.

Model	S%	V%	O%
FGM	76.42	21.34	12.39
CRF	77.16	22.54	9.25
CCA	77.16	21.04	10.99
JEM	78.25	24.45	11.95
LSTM	71.19	19.40	9.70
LSTM-E (Alex)	78.66	24.78	10.30
LSTM-E (VGG)	80.30	27.91	12.54
LSTM-E (C3D)	77.31	28.81	12.39
LSTM-E (VGG+C3D)	**80.45**	**29.85**	**13.88**

fidence using a factor graph and performs probabilistic inference to determine the most likely SVO triplets.

- Joint Embedding Model (JEM) [34]: Proposed most recently, JEM jointly models video and the corresponding text sentences by minimizing the distance of the deep video and compositional text in the joint space.

- Long Shot-Term Memory (LSTM): LSTM attempts to directly translate from video pixels to natural language with a single deep neural network. The video representation is by performing mean pooling over the features of frames using AlexNet.

- Soft-Attention (SA) [35]: SA combines the frame representation from GoogleNet [28] and video clip representation based on a 3-D ConvNet 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 regions of the video while generating sentence.

- Sequence to Sequence - Video to Text (S2VT) [31]: S2VT incorporates both RGB and optical flow inputs, and the encoding and decoding of the inputs and word representations are learnt jointly in a parallel manner.

- Long Shot-Term Memory with visual-semantic Embedding (LSTM-E): We design four runs for our proposed approach, i.e., LSTM-E (Alex), LSTM-E (VGG), LSTM-E (C3D), and LSTM-E (VGG+C3D). The input frame/clip features of the first three runs are from AlexNet, VGG and C3D network respectively. The input of the last one is to concatenate the features from VGG and C3D.

5.2.2 SVO Triplet Prediction Task

As SVO triples can capture the compositional semantics of videos, predicting SVO triplet could indicate the quality of a translation system to a large extent.

We adopt SVO accuracy [34] which measures the exactness of SVO words by binary (0-1 loss), as the evaluation metric. Table 1 details SVO accuracy of compared seven models. Within these models, the former four models (called *Item driven models*) explicitly optimize to identify the best subject, verb and object items for a video; while the later five models (named *Sentence driven models*) focus on training on objects and actions jointly in a sentence and learn to interpret these in different contexts. For the later five sentence driven models, we extract the SVO triplets from the generated sentences by Stanford Parser[1] and the words are also stemmed. Overall, the results across SVO triplet indicate that almost all the four *Item driven models* exhibit better performance than LSTM model which predicts the next word by only considering the contextual relationships with the previous words given the video content. By jointly modeling the relevance between the semantics of the entire sentence and video content with LSTM, LSTM-E significantly improves LSTM. Furthermore, the performances of LSTM-E (VGG), LSTM-E (C3D), and LSTM-E (VGG+C3D) on Subject, Verb and Object are all above that of the four *Item driven models*. The result basically indicates the advantage of further exploring the relevance holistically between the semantics of the entire sentence and video content in addition to LSTM.

Compared to LSTM-E (Alex), LSTM-E (VGG) using a more powerful frame representation brought by a deeper CNN exhibits significantly better performance. In addition, LSTM-E (C3D) which has a better ability in encapsulating temporal information leads to better performance than LSTM-E (VGG) in terms of Verb prediction accuracy. When combining the features from VGG and C3D, LSTM-E (VGG+C3D) further increases the performance gains.

5.2.3 Sentence Generation Task

For *item driven models* including FGM, CRF, CCA and JEM, the sentence generation is often performed by leveraging a series of simple sentence templates (or special language trees) on the SVO triplets [32]. Having verified in [32], using LSTM architecture can lead to a large performance boost against the template-based sentence generation. Thus, Table 2 only shows comparisons of LSTM-based sentence generations. We use the BLEU@N [22] and METEOR scores [2] against all ground truth sentences. Both metrics have been shown to correlate well with human judgement, and widely used in machine translation litera-

---

[1] http://nlp.stanford.edu/software/lex-parser.shtml

Table 2. BLEU@N and METEOR scores for comparing the quality of the sentence generation. All values are reported as percentage (%).

Model	METEOR	BLEU@1	BLEU@2	BLEU@3	BLEU@4
LSTM	26.9	69.8	53.3	42.1	31.2
SA	29.6	**80.0**	64.7	52.6	42.2
S2VT	29.8	-	-	-	-
LSTM-E (Alex)	28.3	74.5	59.8	49.3	38.9
LSTM-E (VGG)	29.5	74.9	60.9	50.6	40.2
LSTM-E (C3D)	29.9	75.7	62.3	52	41.7
LSTM-E (VGG+C3D)	**31.0**	78.8	**66.0**	**55.4**	**45.3**
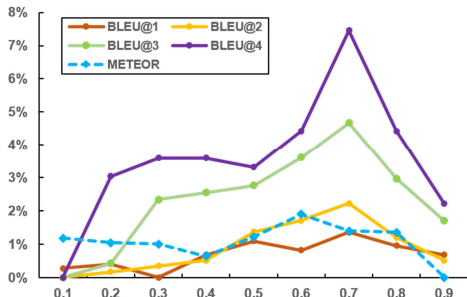
Figure 4. The effect of the tradeoff parameter λ measured by BLEU@N and METEOR.

Table 3. The effect of hidden layer size in our LSTM-E (VGG+C3D) framework measured by BLEU@4 and METEOR.

Hidden layer size	BLEU@4	METEOR	Parameter number
128	38.4	29.0	3.6M
256	40.6	29.6	7.5M
512	45.3	31.0	16.0M

ture. Specifically, BLEU@N measures the fraction of N-gram (up to 4-gram) that are in common between a hypothesis and a reference or set of references, while METEOR computes unigram precision and recall, extending exact word matches to include similar words based on WordNet synonyms and stemmed tokens. As shown in the Table 2, the qualitative results across different N of BLEU and METEOR consistently indicate that the LSTM-E (Alex) significantly outperforms the traditional LSTM model. Moreover, we can find that the performance gain of BLEU@N becomes larger when N increases, where N measures the length of the contiguous sequence in the sentence. This again confirms that LSTM-E is benefited from the way of holistically exploring the relationships between the semantics of the entire sentence and video content by minimizing the distance of their mappings in a visual-semantic embedding. Similar to the observations in SVO prediction task, our LSTM-E (VGG) outperforms LSTM-E (Alex) and can reach 29.5% METEOR. Furthermore, LSTM-E (C3D) achieves 29.9% METEOR and improves the performance to 31.0% when combined with VGG, which makes the improvement over the current two state-of-the-art methods SA by 4.7% and S2VT by 4.0%, respectively.

Figure 5 shows a few sentence examples generated by different methods and human-annotated ground truth. 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, both LSTM-E (Alex) and LSTM-E (VGG+C3D) predict more relevant Subject, Verb and Object (SVO) terms. For example, compared to subject

term "a man", 'People' or "a group of men" is more precise to describe the video content in the second video. Similarly, verb term "singing" presents the fourth video more exactly. The predicted object terms "keyboard" and "motorcycle" are more relevant than "guitar" and "car" in fifth and sixth videos, respectively. Moreover, LSTM-E (VGG+C3D) can offer more coherent sentences. For instance, the generated sentence "a man is talking on a phone" of the third video encapsulates the video content more clearly.

5.3. Experimental Analysis

We will further provide the analysis on the effect of the tradeoff parameter between two losses and the size of hidden layer in LSTM learning.

5.3.1 The Tradeoff Parameter λ

To clarify the effect of the tradeoff parameter λ in Eq.(11), we illustrate the performance curves with a different tradeoff parameter in Figure 4. To make all performance curves fall into a comparable scale, all BLEU@N and METEOR values are specially normalized as follows

$$m'_\lambda = \frac{m_\lambda - \min_\lambda \{m_\lambda\}}{\min_\lambda \{m_\lambda\}} \quad (13)$$

where m_λ and m'_λ denotes original and normalized performance values (BLEU@N or METEOR) with a set of λ, respectively.

From the figures, we can see that all performance curves are like the "∧" shapes when λ varies in a range from 0.1 to 0.9. The best performance is achieved when λ is about 0.7. This proves that it is reasonable to jointly learn the visual-semantic embedding space in the deep recurrent neural networks.

	LSTM: a cat is playing with a mirror **LSTM-E (Alex):** a cat is playing with a watermelon **LSTM-E (VGG+C3D):** a kitten is playing with a toy	**Ground Truth:** ① a kitten is playing with his toy ② a cat is playing on the floor ③ a kitten plays with a toy
	LSTM: a man is dancing **LSTM-E (Alex):** people are dancing **LSTM-E (VGG+C3D):** a group of people are dancing	**Ground Truth:** ① a group of people are dancing ② people are dancing outside ③ many people dance in the street
	LSTM: a woman is talking **LSTM-E (Alex):** a man is talking **LSTM-E (VGG+C3D):** a man is talking on a phone	**Ground Truth:** ① a man is talking on a cell phone ② a man is speaking into a cell phone ③ the man talked on the phone
	LSTM: a man is playing a flute **LSTM-E (Alex):** a man is singing **LSTM-E (VGG+C3D):** a man is singing	**Ground Truth:** ① a man is singing on stage ② a man is singing into a microphone ③ a man sings into a microphone
	LSTM: a man is playing a guitar **LSTM-E (Alex):** a man is playing a keyboard **LSTM-E (VGG+C3D):** a man is playing a piano	**Ground Truth:** ① a person is playing a piano keyboard ② a man plays a keyboard ③ a boy played a keyboard
	LSTM: a man is riding a car **LSTM-E (Alex):** a man is riding a bicycle **LSTM-E (VGG+C3D):** a man is riding a motorcycle	**Ground Truth:** ① someone is riding a motorcycle ② a man is riding his motorcycle ③ a man is riding on a motor bike

Figure 5. Examples of sentence generation results. The videos are represented by sampled frames, the output sentences generated by 1) LSTM, 2) our proposed LSTM-E (Alex) and LSTM-E (VGG+C3D), and 3) Ground Truth: Randomly selected three ground truth sentences.

5.3.2 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, and 512. The results shown in Table 3 indicate increasing the hidden layer size can lead to the improvement of the performance with respect to both BLEU@4 and METEOR. Therefore, in our experiments, the hidden layer size is empirically set to 512, which achieves the best performance.

6. Discussion and Conclusion

In this paper, we have proposed a solution to the video description problem by introducing a novel LSTM-E model structure. In particular, a visual-semantic embedding space is additionally incorporated into LSTM learning. In this way, a global relationship between the video content and sentence semantics is simultaneously measured in addition to the local contextual relationship between the word at each step and the previous ones in LSTM learning. On a popular video description dataset, the results of our experiments demonstrate the success of our approach, outperforming the current state-of-the-art model with a significantly large margin on both SVO prediction and sentence generation.

Our future works are as follows. First, as a video itself is a temporal sequence, the way of better representing the videos by using RNN will be further explored. Moreover, the video description generation might be significantly boosted if we could have sufficient labeled video-sentence pairs to train a deeper RNN.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

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

[3] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 2003.

[4] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. on Neural Networks*, 5(2):157–166, 1994.

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

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014.

[7] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and

G. Zweig. From captions to visual concepts and back. In *CVPR*, 2015.

[8] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[9] A. Graves and N. Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, 2014.

[10] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 2005.

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

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

[13] Y. Jia, M. Salzmann, and T. Darrell. Learning cross-modality similarity for multinomial data. In *ICCV*, 2011.

[14] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

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

[16] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *ICML*, 2014.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. on PAMI*, 2013.

[20] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. In *NIPS Workshop on Deep Learning*, 2014.

[21] J. Martens and I. Sutskever. Learning recurrent neural networks with hessian-free optimization. *ICML*, 2011.

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

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

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

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

[26] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, 2010.

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

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[29] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, 2014.

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3d: Generic features for video analysis. *arXiv preprint arXiv:1412.0767*, 2015.

[31] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. *arXiv preprint arXiv:1505.00487*, 2015.

[32] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

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

[35] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. *arXiv preprint arXiv:1502.08029*, 2015.

[36] W. Zaremba and I. Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.