Computational Visual Media
DOI 10.1007/s41095-016-0059-z Vol. 2, No. 4, December 2016, 379–388
Research Article

LSTM-in-LSTM for generating long descriptions of images

Jun Song[1], Siliang Tang[1], Jun Xiao[1], Fei Wu[1] (✉), and Zhongfei (Mark) Zhang[2]

Abstract In this paper, we propose an approach for generating rich *fine-grained* textual descriptions of images. In particular, we use an LSTM-in-LSTM (long short-term memory) architecture, which consists of an inner LSTM and an outer LSTM. The inner LSTM effectively encodes the long-range *implicit* contextual interaction between visual cues (i.e., the spatially-concurrent visual objects), while the outer LSTM generally captures the *explicit* multi-modal relationship between sentences and images (i.e., the correspondence of sentences and images). This architecture is capable of producing a long description by predicting one word at every time step conditioned on the previously generated word, a hidden vector (via the outer LSTM), and a context vector of fine-grained visual cues (via the inner LSTM). Our model outperforms state-of-the-art methods on several benchmark datasets (Flickr8k, Flickr30k, MSCOCO) when used to generate *long* rich fine-grained descriptions of given images in terms of four different metrics (BLEU, CIDEr, ROUGE-L, and METEOR).

Keywords long short-term memory (LSTM); image description generation; computer vision; neural network

1 Introduction

Automatically describing the content of an image by means of text (description generation) is a fundamental task in artificial intelligence, with many applications. For example, generating descriptions of images may help visually impaired people better understand the content of images and retrieve images using descriptive texts. The challenge of description generation lies in appropriately developing a model that can effectively represent the visual cues in images and describe them in the domain of natural language at the same time.

There have been significant advances in description generation recently. Some efforts rely on manually-predefined visual concepts and sentence templates [1–3]. However, an effective image description model should be free of hard coded templates and categories. Other efforts treat the image description task as a multi-modal retrieval problem (e.g., *image–query–text*) [4–7]. Such methods obtain a descriptive sentence of each image by retrieving similarly described images from a large database and then modifying these retrieved descriptions based on the query image. Such methods lack the ability to generate descriptions of unseen images.

Motivated by recent successes in computer vision and natural language processing, current image description generation approaches generate more reasonable descriptive sentences of given images [8–10] based on an approach of *word-by-word* generation via recurrent neural networks (RNN) (e.g., using long short-term memory (LSTM)) since these approaches store context information in a recurrent layer. Most description generation research only utilizes the image being described to the RNN at the beginning [10]. By looking at the image only once during word-by-word generation, the precision and recall of the predicted noun words (i.e., visual objects in images) decrease rapidly with their position of

1 College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China. E-mail: J. Song, songjun54cm@zju.edu.cn; S. Tang, siliang@cs.zju.edu.cn; J. Xiao, junx@cs.zju.edu.cn; F. Wu, wufei@cs.zju.edu.cn (✉).

2 Department of Computer Science, Watson School of Engineering and Applied Sciences, Binghamton University, Binghamton, NY, USA. E-mail: zhongfei@cs.binghamton.edu.

Manuscript received: 2016-07-25; accepted: 2016-08-19

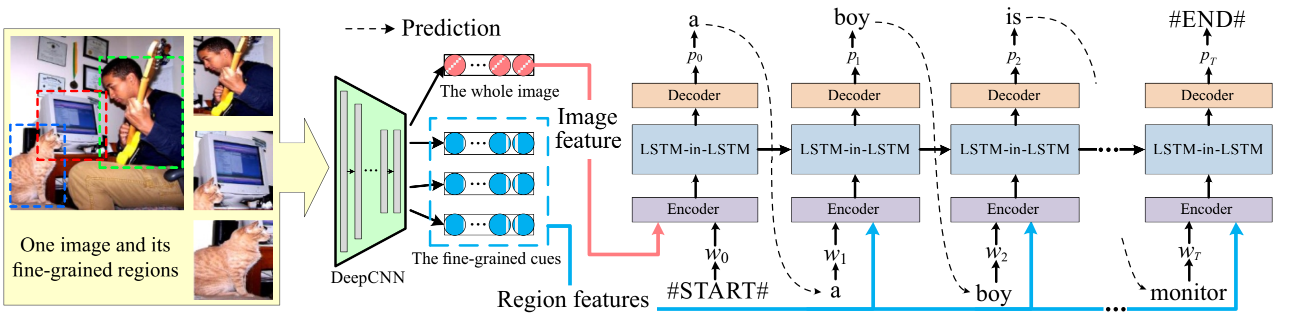


Fig. 1 Overview of our approach. The DeepCNN model projects the pixels of an image and its fine-grained regions into a 4096-dimensional feature. The encoder layer encodes the textual words, the whole image, and the visual objects as vectors. The prediction layer outputs one hidden vector at each step which is then used to predict the next word in the decoder layer. While training, the t^{th} word in the sentence is fed into the model to predict the next word (solid lines). While testing, the word predicted at the previous step $(t-1)$ is fed into the model at step t.

occurrence in a sentence (as shown in Fig. 5), since these approaches merely preserve global semantics at the beginning and disregard the fine-grained interactions between visual cues which could be useful if we wish to generate richer, more descriptive captions.

From the point of view of the mutual utilization of visual and textual contexts during each step of word-by-word generation, image description generation methods may in general be categorized into two classes. The first class repeatedly takes advantage of the whole image at each time step of the output word sequence [9]. Such methods may identify the most interesting salient objects the words refer to; however, they may still ignore the fine-detail objects.

The second class explicitly learns the correspondences between visual objects (detected as object-like or regions of attention) and the matching words at each step of generation, and then generates the next word according to both the correspondences and the LSTM hidden vector [11, 12]. Such methods may neglect *long-range* interactions between visual cues (e.g., the spatially-concurrent visual objects).

In this paper, we develop a new neural network structure called LSTM-in-LSTM (long short-term memory) which can generate semantically rich and descriptive sentences for given images. The LSTM-in-LSTM consists of an inner LSTM (encoding the implicit long-range interactions between visual cues) and an outer LSTM (capturing the explicit multi-modal correspondences between images and sentences). This architecture is capable of producing a description by predicting one word at each time step conditioned on the previously generated word, a hidden vector (via the outer LSTM), and the context vector of fine-grained visual cues (via the inner LSTM).

Compared with existing methods, the proposed LSTM-in-LSTM architecture, as illustrated in Fig. 1, is particularly appropriate for generating rich fine-grained *long* descriptions with appealing diversity, owing to its modeling of long-range interactions between visual cues.

2 Related work

2.1 Natural language models

Over the last few years, natural language models based on neural networks have been widely used in the natural language processing domain. Artificial neural networks have been employed to learn a distributed representation for words which better captures the semantics of words [13]. Recursive neural networks have been used to encode a natural language sentence as a vector [7]. Palangi et al. [14] use a recurrent neural network (RNN) with *long short-term memory* (LSTM) to sequentially take each word in a sentence, and encode it as a semantic vector. A recurrent neural network encoder–decoder architecture has been proposed to encode a source language sentence, and then decode it into a target language [15].

2.2 Deep model for computer vision

Methods based on deep neural networks have been adopted by a large number of computer vision applications. *Deep convolutional neural networks* (DeepCNN) have achieved excellent performance in image classification tasks (e.g., AlexNet [16], VggNet [17]). Object detection systems based on a well trained DeepCNN outperform previous works (RCNN [18], SPPNet [19]). Girshick [20] proposed Fast-RCNN which is much faster than RCNN and SPPNet for object detection during both training and testing.

2.3 Image descriptions

There are two main categories of methods for automatically describing an image: retrieval based methods and generation based methods. Many works try to describe an image by retrieving a relevant sentence from a database. They learn the co-embedding of images and sentences in a common vector space and then descriptions are retrieved which lie close to the image in the embedding space [4, 5, 7]. Karpathy et al. [21] argue that by using a correspondence model that is based on a combination of image regions and phrases of sentences, the performance of retrieval based image description methods can be boosted. Generation based methods often use fixed templates or generative grammars [22]. Other generation methods more closely related to our method learn the probability distribution of the next word in a sentence based on all previously generated words [8–10].

3 Method

Our model comprises three layers: the encoder layer, the prediction layer, and the decoder layer. In the encoder layer, the words in sentences are encoded into different word vectors (one vector per word). For whole images and visual objects (detected as object-like regions), a deep convolutional neural network is used to encode them into 4096-dimensional visual vectors. The prediction layer outputs a single hidden vector which is then used to predict the next word in the decoder layer. The overview of our approach is illustrated in Fig. 1.

3.1 Encoder layer

First, we encode the words in sentences, the whole image, and the visual objects in the image as vectors. Given training data denoted as (S, I), which is a pair of a sentence S and its length (in words) T, and image I. The words in the sentence S are $w_1, w_2, \cdots, w_T$. We first denote each word as a one-hot representation $\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_T$. This representation is a binary representation which has the same dimension as the vocabulary size and only one non-zero element. After that, the one-hot representation is transformed into an h-dimensional vector as follows:

$$\boldsymbol{\omega}_t = \boldsymbol{W}_s \boldsymbol{w}_t \tag{1}$$

$\boldsymbol{W}_s$ is a matrix of size $h \times V$, where V is the size of the vocabulary. $\boldsymbol{W}_s$ is randomly initialized and learned during the model training.

For images, we use Fast-RCNN [20] to detect the visual objects in the image. Fast-RCNN is a fast framework for object detection based on a deep convolutional neural network. This framework is trained using a multi-task loss function in a single training stage, which not only simplifies learning but also improves the detection accuracy.

A threshold τ is set to select the *valid* visual objects from all objects detected by Fast-RCNN. Visual objects with a detection score higher than τ are considered as *valid* visual objects; the rest are discarded. The number of the *valid* objects may be different in each image.

For each image I and each visual object r, we first obtain their 4096-dimensional VGGNet16 [17] fc7 features. Then these features are encoded as h-dimensional vectors as follows:

$$\boldsymbol{v}_I = \boldsymbol{W}_e CNN_{VGGNet16}(I) + \boldsymbol{b}_e \tag{2}$$

$$\boldsymbol{r} = \boldsymbol{W}_r CNN_{VGGNet16}(r) + \boldsymbol{b}_r \tag{3}$$

$\boldsymbol{v}_I$ is the vector of image I and $\boldsymbol{r}$ is the vector of visual object r. The $CNN_{VGGNet16}(\cdot)$ function projects the pixels into a 4096-dimensional VGGNet16 [17] fc7 feature. $\boldsymbol{W}_e$ and $\boldsymbol{W}_r$ are matrices with dimension $h \times 4096$; $\boldsymbol{b}_e$ and $\boldsymbol{b}_r$ are bias vectors with dimension h. $\boldsymbol{W}_e$, $\boldsymbol{W}_r$, $\boldsymbol{b}_e$, and $\boldsymbol{b}_r$ are parameters learned during training.

3.2 Prediction layer

The prediction layer consists of two LSTMs, namely the outer LSTM and the inner LSTM. We call this architecture LSTM-in-LSTM.

3.2.1 Basic LSTM unit

In order to predict each word in a sentence, the recurrent net needs to store information over an extended time interval. Here we briefly introduce the basic LSTM approach [23] which has had great success in machine translation [24] and sequence generation [25].

As shown in Fig. 2, a single memory cell c is surrounded by three gates controlling whether to input new data (*input gate* i), whether to forget history (*forget gate* f), and whether to produce the current value (*output gate* o) at each time t. The memory cell in LSTM encodes information at every time step concerning what inputs have been observed prior to this step. The value of each gate is calculated according to the word vector $\boldsymbol{\omega}_t$ at step t and the predicted hidden vector $\boldsymbol{m}_{t-1}$ at step $t-1$. The definitions of the memory cell and each gate are as follows:

$$\left\{\begin{aligned} \boldsymbol{x}_t &= [\boldsymbol{\omega}_t; \boldsymbol{m}_{t-1}] \\ \boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i \cdot \boldsymbol{x}_t) \\ \boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f \cdot \boldsymbol{x}_t) \\ \boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \cdot \boldsymbol{x}_t) \\ \boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \phi(\boldsymbol{W}_c \cdot \boldsymbol{x}_t) \\ \boldsymbol{m}_t &= \boldsymbol{o}_t \odot \boldsymbol{c}_t \end{aligned}\right. \tag{4}$$

where $\odot$ represents the element-wise product. σ and ϕ are nonlinearlity mapping functions. In our experiments, we set σ as a sigmoid function and ϕ as hyperbolic tangent. $\boldsymbol{m}_t$ is the output of the LSTM at step t. $\boldsymbol{W}_i$, $\boldsymbol{W}_f$, $\boldsymbol{W}_o$, and $\boldsymbol{W}_c$ are parameter matrices learned during training.

3.2.2 LSTM-in-LSTM unit

As previously discussed, we attempt to employ both the explicit multi-modal correspondence of sentences and images, and the implicit long-range interactions of fine-grained visual cues, during the prediction of each word. The proposed LSTM-in-LSTM has two layers of LSTM networks, namely the outer LSTM and the inner LSTM.

See Fig. 3. The outer LSTM is a basic LSTM unit. At each step t, the outer LSTM takes a word vector $\boldsymbol{\omega}_t$ (the t^{th} word vector of the sentence in training, or the word vector of the previously predicted word in prediction), the last predicted hidden vector $\boldsymbol{m}_{t-1}$, and the context output vector of the inner LSTM $\boldsymbol{m}_t^{\text{inner}}$ as the input. In the outer LSTM, the vector $\boldsymbol{x}_t$ is defined as follows:

$$\boldsymbol{x}_t = [\boldsymbol{\omega}_t; \boldsymbol{m}_{t-1}; \boldsymbol{m}_t^{\text{inner}}] \tag{5}$$

$\boldsymbol{x}_t$ is employed to obtain the t step output of the LSTM-in-LSTM $\boldsymbol{m}_t$.

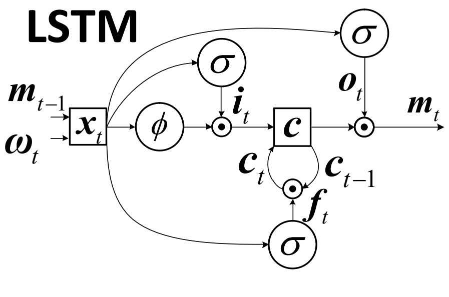


Fig. 2 The basic LSTM method.

The inner LSTM is composed of stacked LSTM units. In essence, the gates of the inner LSTM learn to adaptively look up significant visual object-like regions, and encode the implicit interactions between visual cues at each step. For the k^{th} basic LSTM in the inner LSTM, the input is the k^{th} object vector $\boldsymbol{r}_k$ and the output vector of the previous basic LSTM ($\boldsymbol{m}_{t-1}$ for the first LSTM unit), as follows:

$$\boldsymbol{x}_k^{\text{inner}} = [\boldsymbol{m}_{k-1}^{\text{inner}}; \boldsymbol{r}_k], \quad \boldsymbol{m}_0^{\text{inner}} = \boldsymbol{m}_{t-1} \tag{6}$$

Note that the parameters of the outer LSTM (e.g., $\boldsymbol{W}_i$, $\boldsymbol{W}_f$, $\boldsymbol{W}_o$, and $\boldsymbol{W}_c$) differ from those of the inner LSTM ($\boldsymbol{W}_i^{\text{inner}}$, $\boldsymbol{W}_f^{\text{inner}}$, $\boldsymbol{W}_o^{\text{inner}}$, and $\boldsymbol{W}_c^{\text{inner}}$); however all basic LSTM units in the inner LSTM share the same parameters.

For the inner LSTM, each basic LSTM unit takes one visual object vector as an input, so the number of basic LSTM units in the inner LSTM equals the number of *valid* visual objects.

3.3 Training the model

We use a probabilistic mechanism to generate the description of each image. The training objective is to minimize the log-likelihood of the *perplexity* of each sentence in the training set using an L_2 regularization term, as shown in Eq. (7):

$$O(\theta) = \\ \arg\min_{\theta}\left(\frac{1}{\sum_{i=1}^{N} T_i}\sum_{i=1}^{N}\log \mathcal{PPL}(S_i|I_i,\theta) + \frac{\lambda}{2}||\theta||_2^2\right) \tag{7}$$

θ denotes all training parameters in our model, N is the size of the training set, i indicates the index of each training sample, and I_i and S_i denote the image and the sentence for the i^{th} training sample. T_i denotes the length (in words) of sentence S_i; λ is the weighting parameter for standard L_2 regularization of θ.

The *perplexity* of a sentence is calculated as the negative log-likelihood of its words according to its associated image, as follows:

$$\mathcal{PPL}(S_i|I_i,\theta) = -\sum_{t=1}^{T_i}\log_2 P(w_t^{(i)}|w_{1:t-1}^{(i)}, I_i, \theta) \tag{8}$$

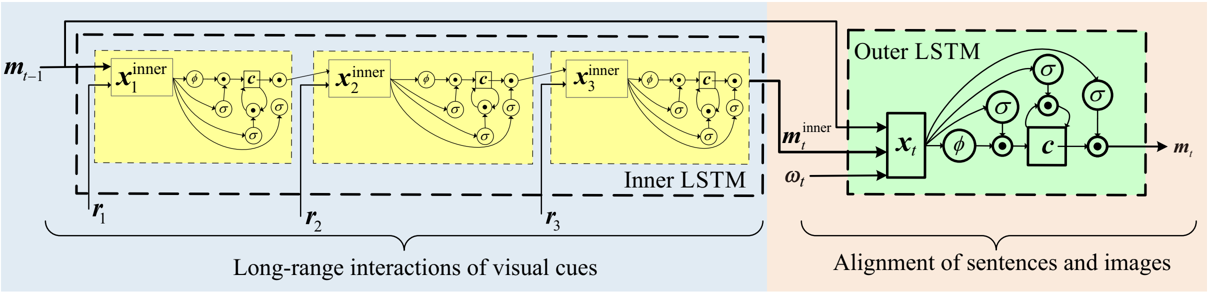


Fig. 3 LSTM-in-LSTM structure. For simplicity, we show three visual object vectors $\boldsymbol{r}_1$, $\boldsymbol{r}_2$, and $\boldsymbol{r}_3$, so there are 3 LSTM units in the inner LSTM. The 3 visual objects are sequentially fed into the inner LSTM in a descending order according to their Fast-RNN detection scores. The parameters of the outer LSTM and the inner LSTM differ, but each LSTM unit in the inner LSTM shares the same parameters.

Here the probability of each word is computed based on the words in its context and the corresponding image. $w_t^{(i)}$ denotes the t^{th} word in the i^{th} sentence and $w_{1:t-1}^{(i)}$ denotes the words before the t^{th} word in the i^{th} sentence. Therefore, minimizing the *perplexity* is equivalent to maximizing the log-likelihood. Stochastic gradient descent is used to learn the parameters of our model.

Algorithm 1 summarises the training procedure for our model. outerLSTM(·) denotes the forward pass of the outer LSTM and innerLSTM(·) denotes the forward pass of the inner LSTM. We insert a start token *#START#* at the beginning of each sentence and an end token *#END#* at its end. Thus the subscript t expands from 0 (*#START#*) to $T + 1$ (*#END#*). In the first step ($t = 0$), the word vector of the start token *#START#* $\boldsymbol{\omega}_0$ and the vector of the i^{th} image ($\boldsymbol{v}_{I^i}$) are fed into the outer LSTM to obtain the first predicted hidden vector $\boldsymbol{m}_0$.

Algorithm 1: Algorithm for training our model

```
Input: A batch $\mathcal{B}$ of training data, as image and sentence pairs.
for all pair $(S_i, I_i) \in \mathcal{B}$ do
    /* Encoder layer */
    Encode each word in sentence $I_i$ into word vectors $\boldsymbol{\omega}_t$ $(t = 0, \cdots, T_i)$.
    Detect visual objects and learn the vector of objects $\boldsymbol{r}_k$ $(k = 1, \cdots, K)$ and the image $\boldsymbol{v}_{I^i}$.
    /* Prediction layer */
    $\boldsymbol{m}_0 = \text{outerLSTM}(\boldsymbol{\omega}_0, \boldsymbol{0}, \boldsymbol{v}_{I^i})$
    for all $t \leftarrow 1$ to $T_i$ do
        $\boldsymbol{m}_t^{\text{inner}} = \text{innerLSTM}(\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_K)$
        $\boldsymbol{m}_t = \text{outerLSTM}(\boldsymbol{\omega}_t, \boldsymbol{m}_{t-1}, \boldsymbol{m}_t^{\text{inner}})$
    end for
    /* Decoder layer */
    for all $t \leftarrow 0$ to $T_i$ do
        $\boldsymbol{p}_t = \text{Softmax}(\boldsymbol{W}_d \boldsymbol{m}_t + \boldsymbol{b}_d)$
    end for
    Calculate and accumulate the gradients.
end for
Calculate the update values $\nabla\theta$
/* Update the parameters */
$\theta = \theta - \nabla\theta$
Output: The parameters $\theta$ of the model.
```

3.4 Sentence generation

Given an image, its descriptive sentence is generated in a word-by-word manner according to the predicted probability distribution at each step, until the end token *#END#* or some maximum length L is reached. We insert a start token *#START#* at the beginning of each sentence and an end token *#END#* at its end. Thus the subscript t goes from 0 (*#START#*) to $T+1$ (*#END#*). In the first step ($t = 0$), the word vector of the start token *#START#* $\boldsymbol{\omega}_0$ and the vector of i^{th} image (e.g., $\boldsymbol{v}_{I^i}$) are fed into the outer LSTM to get the first predicted hidden vector $\boldsymbol{m}_0$. We use *BeamSearch* to iteratively select the set of κ best sentences up to step t as candidates to generate sentences at step $t + 1$, and keep only the resulting best κ of them. Algorithm 2 summarises the process used to generate one sentence.

4 Experiments

4.1 Comparison methods

Since we are interested in word-by-word image-caption generation which utilizes mutual visual and textual information during each prediction step, we compare our work to three types of algorithms as follows:

- **NIC model** [10] and **Neural-Talk** [8]: NIC and Neural-Talk models only utilize whole-image

Algorithm 2: Generating one sentence in our model

```
Input: The input image $I$.
Detect visual objects and learn the vectors $\boldsymbol{r}_k$ ($k = 1, \cdots, K$) and $\boldsymbol{v}_I$.
$\boldsymbol{\omega}_0$ is the word vector of #START#.
$\boldsymbol{m}_0 = \text{outerLSTM}(\boldsymbol{\omega}_0, \boldsymbol{0}, \boldsymbol{v}_I)$
$t = 1$, $w_t$ is the word with the highest probability.
while $w_t$ is not #END# and $t \leqslant L$ do
    $\boldsymbol{m}_t^{\text{inner}} = \text{innerLSTM}(\boldsymbol{r}_1, \cdots, \boldsymbol{r}_K)$
    $\boldsymbol{m}_t = \text{outerLSTM}(\boldsymbol{\omega}_t, \boldsymbol{m}_{t-1}, \boldsymbol{m}_t^{\text{inner}})$
    $\boldsymbol{p}_t = \text{Softmax}(\boldsymbol{W}_d \boldsymbol{m}_t + \boldsymbol{b}_d)$
    $t = t + 1$
    $w_t$ is the word with the highest probability.
end while
Output: The sentence with words in sequence: $w_1, \cdots, w_T$.
```

information at the beginning during description prediction.

- **m-RNN** [9]: the m-RNN model employs whole-image information at each prediction step.
- **attention model** [11]: this attention model uses fine-grained visual cues (regions of attention) during each prediction step.

4.2 Datasets

Three different benchmark datasets were used in the experiments; Table 1 shows the size of each dataset.

- **Flickr8k**: the Flickr8k [5] dataset comprises 8000 images from Flickr showing persons and animals. Each image has 5 descriptive sentences.
- **Flickr30k**: the Flickr30k [26] comprises 30,000 images from Flickr showing daily activities, events, and scenes. Each image has 5 descriptive sentences.
- **MSCOCO**: the Microsoft COCO [27] dataset comprises more than 120,000 images. Each image has 5 descriptive sentences.

4.3 Experimental setup

In order to perform a fair comparison, we used the same VGGNet16 fc7 feature as the visual feature for all models. For the Flickr8k and Flickr30k datasets, the dimension of the hidden vectors was $h = 512$. For MSCOCO, $h = 600$. In our experiments, we used the threshold $\tau = 0.5$ to select valid visual objects in each image.

Table 1 Sizes of the three benchmark datasets, and the numbers of images used for training, validation, and testing

Dataset	Size		
	Training	Validation	Testing
Flickr8k	6000	1000	1000
Flickr30k	28000	1000	1000
MSCOCO	82783	40504	5000

4.4 Results

Our experiments compared the methods in three ways: (i) a qualitative analysis of long description generation performance in terms of four metrics, (ii) the predictive ability for rich fine-grained semantics in *long* descriptive sentences, and (iii) the ability to predict SVO (*subject–verb–object*) triplets.

4.4.1 Generation of long descriptions

Many metrics have been used in the image description literature. The most commonly used metrics are BLEU [28] and ROUGE [29]. BLEU is a precision-based measure and ROUGE is a recall-related measure. BLEU and ROUGE scores can be computed automatically from a number of ground truth sentences, and have been used to evaluate a number of sentence generation systems [2, 5, 30]. In this paper we use BLEU-N, ROUGE-L, CIDEr [31], and METEOR [32] to evaluate the effectiveness of our model. We used the open-source project coco-caption software[1] to calculate those metrics.

When generating descriptions, accurate generation of the sentences which consist of many words (i.e., long sentences) is difficult, as it is likely that long sentences deliver rich fine-grained semantics. We argue that the LSTM-in-LSTM architecture is capable of predicting long sentence descriptions since it implicitly learns the contextual interactions between visual cues. Thus, we divide the test data into two parts: images with *long* sentence descriptions and images with *short* sentence descriptions. Descriptions of images in the test dataset are considered to be long if they have more than 8 words (which is the average length of the sentences in the MSCOCO test dataset); the remaining images have short descriptions.

Table 2 reports the image-captioning performance of the images with long and short descriptions. ***B-N*** gives the BLEU-N metric. The performance of our model is comparable to that of the state-of-the-art methods on short descriptions. However, the performance of our approach is remarkably better than that for other models for long descriptions.

[1] coco-caption: https://github.com/tylin/coco-caption.

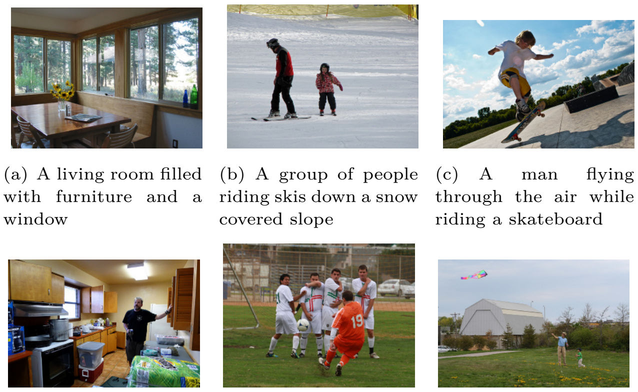
Table 2 Performance for image-captioning on Flickr8k, Flickr30k, and MSCOCO on *long* and *short* descriptions. The best results shown in boldface

Model	B-1	B-2	B-3	B-4	CIDEr	ROUGE-L	METEOR
	Flickr8k (short descriptions / long descriptions)						
NIC model	66.5 / 53.4	48.7 / 35.1	33.3 / 22.3	21.2 / 14.7	52.1 / 31.7	48.1 / 40.7	21.0 / 17.6
Neural-Talk	63.0 / 54.8	43.3 / 34.9	28.5 / 21.5	17.9 / 13.7	38.3 / 22.2	42.0 / 38.8	17.0 / 15.0
m-RNN	**68.2** / 53.4	**48.4** / 32.1	**31.9** / 19.9	19.7 / 12.6	46.1 / 29.3	43.6 / 38.8	17.9 / 16.4
Our model	64.1 / **58.1**	45.4 / **39.0**	30.8 / **26.2**	**19.9** / **18.2**	**48.4** / **37.2**	**44.2** / **43.3**	**18.6** / **18.4**
	Flickr30k (short descriptions / long descriptions)						
NIC model	62.5 / 57.6	40.4 / 36.7	26.3 / 22.9	16.5 / 14.6	25.9 / 22.2	37.3 / 38.7	14.7 / 15.1
Neural-Talk	60.9 / 57.4	41.1 / 36.0	28.6 / 22.1	18.0 / 14.1	24.3 / 14.9	39.6 / 38.5	14.7 / 14.1
m-RNN	65.8 / 58.4	45.7 / 37.7	30.4 / 24.7	18.8 / 16.6	34.3 / 28.4	39.5 / 40.1	15.6 / 16.0
Our model	**66.7** / **61.4**	**46.9** / **40.8**	**31.3** / **27.1**	**19.7** / **18.2**	**39.9** / **29.6**	**41.0** / **41.5**	**16.3** / **16.7**
	MSCOCO (short descriptions / long descriptions)						
NIC model	68.3 / 61.7	49.4 / 43.7	35.0 / 31.5	24.9 / 23.3	68.5 / 68.8	48.1 / 46.3	20.4 / 21.3
Neural-Talk	67.6 / 57.0	48.7 / 40.2	34.5 / 29.0	24.6 / 21.0	60.8 / 55.0	47.2 / 42.4	19.0 / 19.5
m-RNN	**70.0** / 63.0	**52.1** / 45.5	**37.5** / 33.1	26.7 / 24.1	74.4 / 70.9	48.9 / 46.2	**21.3** / 20.8
Our model	69.8 / **66.3**	51.5 / **48.8**	37.3 / **35.9**	**27.0** / **26.9**	**76.5** / **80.9**	**49.2** / **49.1**	20.9 / **23.0**

Compared with the second best methods, our long descriptions of the MSCOCO data show 5.2%, 7.3%, 8.5%, 11.6%, 14.1%, 6.0%, and 8.0% average performance improvements for B-1, B-2, B-3, B-4, CIDEr, ROUGE-L, and METEOR metrics, respectively. Other methods which utilize the visual cues at each step also achieve a better performance than methods only using the visual cues at the beginning step; this observation demonstrates that appropriate utilization of visual information helps boost the performance of image-captioning with rich diverse semantics. We show some examples generated by our model for the MSCOCO dataset in Fig. 4.

4.4.2 Fine-grained semantic interaction

During image captioning, the caption is predicted word-by-word in grammatical interaction order. It is interesting to show the prediction performance of the nouns (i.e., the corresponding grounded visual objects) in order (deminstrating how the next noun word is generated). Figure 5 illustrates the average prediction performance of the first 5 noun words in sentences in terms of recall and precision for the Flick8k dataset.



(a) A living room filled with furniture and a window

(b) A group of people riding skis down a snow covered slope

(c) A man flying through the air while riding a skateboard

(d) A man standing in a kitchen preparing food

(e) A group of young men playing a game of soccer

(f) A group of people flying kites in a field

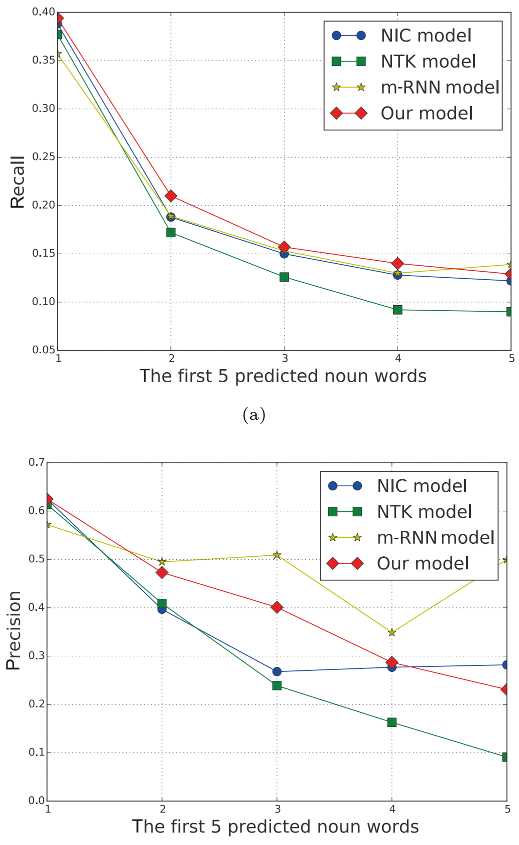
Fig. 4 Long descriptions of images generated by our model.

As can be seen in Fig. 5(a), our model (red line with diamond) shows better performance than the other models due to taking into account long-range interactions between visual objects at each prediction step in our model.

Figure 5(b) shows that our model does not perform better than m-RNN. In m-RNN, the whole image is used at each step and therefore mRNN has a tendency to predict noun words for a large region several times. For the test images in the Flick8k dataset, the occurrence rate of one noun word appearing more than once in a sentence is 0.076. The rates of the predicted noun words occurring more than once in a sentence are 0.245 (m-RNN), 0.015 (Neural-Talk), and 0.039 (our model). This demonstrates that our model is capable of generating more diverse rich fine-grained descriptions.

4.4.3 SVO triplet prediction

We next evaluate the performance of our model in terms of predicting SVO (*subject–verb–object*) triplets. First, we found all SVO triplets in the descriptive sentences in the Flickr8k and Flickr30k test data respectively, using the Stanford Parser [33].



(a)

(b)

Fig. 5 Recall–precision curves in terms of the first 5 predicted noun words from NIC model, Neural-Talk (NTK) model, m-RNN model, and our model.

For example, given the sentence "a small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it", we get the following SVO triplets: (girl, in, grass), (rainbow, on, grass), (girl, play, fingerpaint), (girl, play, rainbow). Then we remove the *object* of each triplet, and feed the visual content (the whole image and the visual objects), the *subject* and the *verb* into each method, and evaluate how well it can predict the removed *object*.

Table 3 compares the ability of different models to predict the removed *object*. R@K (Recall at K) measures whether the correct result is ranked ahead of others. We use R@K ($K = 1, 5, 10, 15, 20$) to compute the fraction of times where the correct result is found among the top K ranked items. A higher R@K means a better performance.

Table 3 Triplet prediction performance. The best results are shown in boldface

Model	**R@1**	**R@5**	**R@10**	**R@15**	**R@20**
	Flickr8k				
Neural-Talk	0.50	1.80	3.00	5.01	6.11
NIC model	0.80	4.91	7.52	9.02	11.52
m-RNN	0.60	5.31	8.82	11.02	13.03
Our model	**1.20**	**6.11**	**9.42**	**12.63**	**14.73**
	Flickr30k				
Neural-Talk	0.30	1.50	2.50	3.30	4.60
NIC model	0.40	1.60	3.30	4.40	5.50
m-RNN	0.40	2.60	4.80	6.60	8.20
Our model	**0.40**	**3.00**	**6.20**	**8.50**	**10.60**

5 Limitations and further work

The major limitation of our model lies in the time taken to train our model. Compared to other models which ignoring contextual interaction between visual cues, our model spends more time for object detection and encoding the long-range implicit contextual interactions. Our model can generate rich fine-grained textual descriptions of each image; it could be further extended to generate much more detailed descriptions of visual objects in each image and much more accurate descriptions of the interactions between visual objects.

6 Conclusions

This paper proposed an LSTM-in-LSTM architecture for image captioning. The proposed model not only encodes long-range *implicit* contextual interactions between visual cues (spatially occurrences of visual objects), but also captures the *explicit* hidden relations between sentences and images (correspondence of sentences and images). The proposed method shows significant improvements over state-of-the-art methods, especially for long sentence descriptions.

Acknowledgements

This work was supported in part by the National Basic Research Program of China (No. 2012CB316400), National Natural Science Foundation of China (Nos. 61472353 and 61572431), China Knowledge Centre for Engineering Sciences and Technology, the Fundamental Research Funds for the Central Universities and 2015 Qianjiang Talents Program of Zhejiang Province. Z. Zhang was supported in part by the US NSF (No. CCF-1017828) and Zhejiang Provincial Engineering Center on Media Data Cloud Processing and Analysis.

References

[1] Farhadi, A.; Hejrati, M.; Sadeghi, M. A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; Forsyth, D. Every picture tells a story: Generating sentences from images. In: *Computer Vision—ECCV 2010.* Daniilidis, K.; Maragos, P.; Paragios, N. Eds. Springer Berlin Heidelberg, 15–29, 2010.

[2] Kulkarni, G.; Premraj, V.; Ordonez, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A. C.; Berg, T. L. BabyTalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 35, No. 12, 2891–2903, 2013.

[3] Li, S.; Kulkarni, G.; Berg, T. L.; Berg, A. C.; Choi, Y. Composing simple image descriptions using web-scale n-grams. In: Proceedings of the 15th Conference on Computational Natural Language Learning, 220–228, 2011.

[4] Gong, Y.; Wang, L.; Hodosh, M.; Hockenmaier, J.; Lazebnik, S. Improving image-sentence embeddings using large weakly annotated photo collections. In: *Computer Vision—ECCV 2014.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer International Publishing, 529–545, 2014.

[5] Hodosh, M.; Young, P.; Hockenmaier, J. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research* Vol. 47, 853–899, 2013.

[6] Ordonez, V.; Kulkarni, G.; Berg, T. L. Im2text: Describing images using 1 million captioned photographs. In: Proceedings of Advances in Neural Information Processing Systems, 1143–1151, 2011.

[7] Socher, R.; Karpathy, A.; Le, Q. V.; Manning, C. D.; Ng, A. Y. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics* Vol. 2, 207–218, 2014.

[8] Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 3128–3137, 2015.

[9] Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Huang, Z.; Yuille, A. Deep captioning with multimodal recurrent neural networks (m-RNN). *arXiv preprint* arXiv:1412.6632, 2014.

[10] Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3156–3164, 2015.

[11] Jin, J.; Fu, K.; Cui, R.; Sha, F.; Zhang, C. Aligning where to see and what to tell: Image caption with region-based attention and scene factorization. *arXiv preprint* arXiv:1506.06272, 2015.

[12] Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. S.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In: Proceedings of the 32nd International Conference on Machine Learning, 2048–2057, 2015.

[13] Bengio, Y.; Schwenk, H.; Senécal, J.-S.; Morin, F.; Gauvain, J.-L. Neural probabilistic language models. In: *Innovations in Machine Learning.* Holmes, D. E.; Jain, L. C. Eds. Springer Berlin Heidelberg, 137–186, 2006.

[14] Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Deep sentence embedding using the long short term memory network: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* Vol. 24, No. 4, 694–707, 2016.

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

[16] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 1097–1105, 2012.

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

[18] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 580–587, 2014.

[19] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *Computer Vision—ECCV 2014.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer International Publishing, 346–361, 2014.

[20] Girshick, R. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448, 2015.

[21] Karpathy, A.; Joulin, A.; Li, F. F. F. Deep fragment embeddings for bidirectional image sentence mapping. In: Proceedings of Advances in Neural Information Processing Systems, 1889–1897, 2014.

[22] Elliott, D.; Keller, F. Image description using visual dependency representations. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 1292–1302, 2013.

[23] Sutton, R. S.; Barto, A. G. *Reinforcement Learning: An Introduction.* The MIT Press, 1998.

[24] Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 3104–3112, 2014.

[25] Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint* arXiv:1308.0850, 2013.

[26] Young, P.; Lai, A.; Hodosh, M.; Hockenmaier, J. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* Vol. 2, 67–78, 2014.

[27] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision—ECCV 2014*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer International Publishing, 740–755, 2014.

[28] Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J. BLEU: A method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 311–318, 2002.

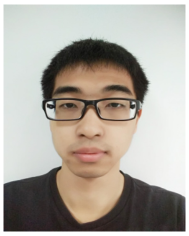
[29] Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, Vol. 8, 2004.

[30] Kuznetsova, P.; Ordonez, V.; Berg, A. C.; Berg, T. L.; Choi, Y. Collective generation of natural image descriptions. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, Vol. 1, 359–368, 2012.

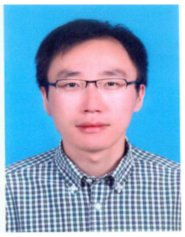
[31] Vedantam, R.; Zitnick, C. L.; Parikh, D. CIDEr: Consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4566–4575, 2015.

[32] Denkowski, M.; Lavie, A. Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the 9th Workshop on Statistical Machine Translation, 2014.

[33] De Marneffe, M.-C.; Manning, C. D. The Stanford typed dependencies representation. In: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, 1–8, 2008.



Jun Song received his B.Sc. degree from Tianjin University, China, in 2013. He is currently a Ph.D. candidate in computer science in the Digital Media Computing and Design Lab of Zhejiang University. His research interests include machine learning, cross-media information retrieval and understanding.



Siliang Tang received his B.Sc. degree from Zhejiang University, Hangzhou, China, and Ph.D. degree from the National University of Ireland, Maynooth, Co. Kildare, Ireland. He is currently a lecturer in the College of Computer Science, Zhejiang University. His current research interests include multimedia analysis, text mining, and statistical learning.

Jun Xiao received his B.Sc. and Ph.D. degrees in computer science from Zhejiang University in 2002 and 2007, respectively. Currently he is an associate professor in the College of Computer Science, Zhejiang University. His research interests include character animation and digital entertainment technology.

Fei Wu received his B.Sc. degree from Lanzhou University, China, in 1996, M.Sc. degree from the University of Macau, China, in 1999, and Ph.D. degree from Zhejiang University, Hangzhou, China, in 2002, all in computer science. He is currently a full professor in the College of Computer Science and Technology, Zhejiang University. His current research interests include multimedia retrieval, sparse representation, and machine learning.

Zhongfei (Mark) Zhang received his B.Sc. (Cum Laude) degree in electronics engineering, M.Sc. degree in information science, both from Zhejiang University, and Ph.D. degree in computer science from the University of Massachusetts at Amhers, USA. He is currently a full professor of computer science in the State University of New York (SUNY) at Binghamton, USA, where he directs the Multimedia Research Laboratory.