Image Captioning with Visual-Semantic LSTM

Nannan Li, Zhenzhong Chen*
School of Remote Sensing and Information Engineering,
Wuhan University, Wuhan, P.R. China
{live, zzchen} @whu.edu.cn

Abstract

In this paper, a novel image captioning approach is proposed to describe the content of images. Inspired by the visual processing of our cognitive system, we propose a visual-semantic LSTM model to locate the attention objects with their low-level features in the visual cell, and then successively extract high-level semantic features in the semantic cell. In addition, a state perturbation term is introduced to the word sampling strategy in the REINFORCE based method to explore proper vocabularies in the training process. Experimental results on MS COCO and Flickr30K validate the effectiveness of our approach when compared to the state-of-the-art methods.

1 Introduction

Automatically generating descriptions of a given image is a prominent research problem in computer vision [Xu *et al.*, 2015; Fang *et al.*, 2015]. It aims to translate visual information into semantic information based on scene understanding and natural language processing. Recently, great progress has been made in image captioning, especially by constructing a CNN-LSTM framework [Mao *et al.*, 2015]. In this framework, the CNN outputs of visual features or semantic attributes are first encoded using the LSTM cell, and then decoded into the corresponding word in the caption.

In image captioning methods based on visual features, typically, the low-level visual features are exploited to produce an attention map that highlights different objects relavant to each word in the caption [Chen *et al.*, 2017; Lu *et al.*, 2017; Liu *et al.*, 2017]. These approaches can find *where* the objects are by predicting the attended objects at each time step, but lack information of the objects' current states such as *holding* and *sitting*. On the other hand, in image captioning approaches based on semantic features, the high-level attributes are utilized to describe the objects and their states in the image [Yao *et al.*, 2017; Wu *et al.*, 2016], such as *group* and *stand*. These methods can obtain *what* attributes are in the image, but the objects described in the attributes cannot be localized.

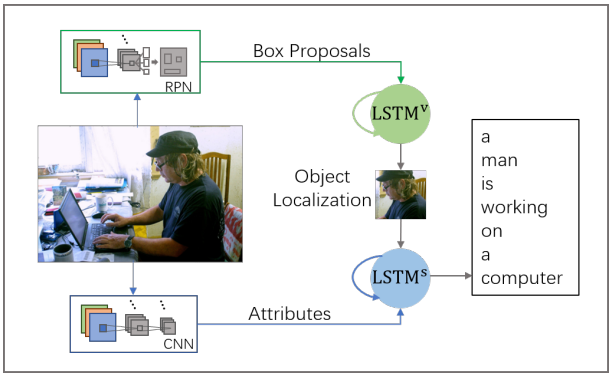

Figure 1: An overview of the proposed VS-LSTM model. First the low-level visual features (i.e., box proposals) and the high-level semantic features (i.e., attributes) of the image are extracted by the Region Proposal Network (RPN) and CNN model, respectively. Then in the LSTM model, the visual cell LSTM^v utilizes the visual features to localize the objects in the image, whilst the semantic cell LSTM^s further integrates the localized objects with their attributes to generate corresponding word.

The above two kinds of approaches either use the low-level visual features to localize objects [Xu *et al.*, 2015; Lu *et al.*, 2017], or utilize the high-level semantic features to describe objects' attributes [Wang *et al.*, 2017b; Wu *et al.*, 2016], whilst the inner connections of these two types of features are not utilized. Inspired by the visual processing of our cognitive system, we propose a visual-semantic LSTM model that incorporates the low-level visual information and the high-level semantic descriptions, considering both *where* the objects are on an object level and *what* their attributes are on a semantic level simultaneously when generating the corresponding word. The proposed model automatically localizes and describes objects with the visual-semantic cells. Please refer to Figure 1 for an overview of our algorithm. The contributions of our work are summarized as follows:

- A novel visual-semantic LSTM based model named VS-LSTM is proposed. The objects in the image are first localized in the visual cell and then described in the

*Corresponding author. This work was supported in part by the National Key R&D Program of China under Grant 2017YFB1002202 and the National Natural Science Foundation of China under Grant 61771348 and 61471273.

semantic cell by successively processing the low-level and high-level features in the caption generation process, which enables VS-LSTM to automatically recognize the objects with their corresponding states when generating each word.

- A sampling strategy with state perturbation term is introduced to encourage exploration of proper vocabularies in the reinforcement learning process, which can balance the training of the frequent words and less frequent words.

2 Related Work and Discussions

2.1 Attention/Attribute Based Method

Recently, various attention mechanisms have been introduced to the CNN-LSTM framework in image captioning. A soft and hard attention mechanism is proposed by [Xu *et al.*, 2015] to change gaze on salient objects when generating each word in the sentence. [Lu *et al.*, 2017] considers that non-visual words require less information from the image and introduce an adaptive attention model. Instead of using the uniformly-divided grids of the outputs of the CNN model as the attention units, [Anderson *et al.*, 2017] utilizes the object proposals of object detection results as the basic attention units and apply attention mechanism on these proposals.

Several methods have been proposed to utilize attributes as the high-level concepts in the caption generation process. [Wu *et al.*, 2016] exploits the detected attributes as the high level features and feed them into an LSTM model to generate captions. [You *et al.*, 2016] utilizes image attributes as an external guide to decide when the attention should be activated. [Yao *et al.*, 2017] explores different ways of feeding image features and attributes into a RNN network. [Wang *et al.*, 2017b] exploits attributes to generate the skeleton sentence and attribute phrases separately.

2.2 REINFORCE Based Method

Reinforcement learning has been exploited as a training method to deal with the out-of-context problem [Choi *et al.*, 2008]. [Rennie *et al.*, 2017] trains the model directly on non-differentiable metrics by using test-time reward as the baseline in the objective function. The implicit optimization towards the target metric improves the results. [Ren *et al.*, 2017] presents a policy network and a value network using an actor-critic reinforcement learning model.

REINFORCE based method faces the dilemma of exploration and exploitation [Sutton and Barto, 1998]. In practice, the model generates the next word depending on the probability of the vocabulary distribution predicted by the model itself, thus the frequent words of the ground truth captions are always preferred.

2.3 Discussions

Models based on attention can predict *where* to attend to with low-level visual features, but lack high-level descriptions of the attended areas. Methods based on high-level attributes can only find *what* objects are in the image, without their spatial relationships. Therefore, we propose to use the low-level visual features to locate the objects first, and then utilize the high-level semantic features to describe the localized objects. In this way, the objects can be recognized with corresponding properties, and detailed captions can be generated. In addition, based on the REINFORCE method, to balance the explorations between the frequent words and the less frequent words, we introduce a sampling strategy that utilizes both the internal vocabulary distribution of the model and the external reward to sample the next word..

The most similar work to ours are those who combine image features with attributes [You *et al.*, 2016; Yao *et al.*, 2017; Wang *et al.*, 2017b]. In these approaches, image features and attributes are viewed as two types of representations of the image, either combined in one LSTM cell or separated in two networks. Different from their methods, we propose that the low-level image features and the high-level image attributes should be processed successively to locate and recognize the objects in the image in a unified network. In our framework, the image features and image attributes are fed into two connected LSTM cells successively, and then decoded into the corresponding word.

3 Visual-Semantic Model

The overall architecture is shown in Figure 1. We first describe the low-level visual features and the high-level semantic features in Section 3.1, and then introduce the proposed LSTM in Section 3.2. The objective function will be explained in Section 3.3.

3.1 Visual and Semantic Features

In the general CNN-LSTM framework [Mao *et al.*, 2015], the outputs of the CNN model are taken as the visual features and fed into the LSTM model. In the classical image classification models [Simonyan and Zisserman, 2015], the output featrues are divided by uniform grids. Each grid may contain information of more than one object. Whereas in the object detection networks [Ren *et al.*, 2015], each output bounding box defines the border of an instance. Considering that it is easier to accurately localize the objects by taking objects as the basic units in our model, we choose the output proposals of Faster R-CNN [Ren *et al.*, 2015] for the visual feature extraction. We also conduct an ablation experiment (see Section 4.2) to address the benefits of using proposals as the visual features in our model.

Let $R = (R_1, R_2, ..., R_k)$ be the top-k detected proposals, ROI-pooling is applied to each proposal so that the output feature vector has the same dimension D. We define the concatenated feature vectors v_i as the low-level visual features V^{v}.

$$V^{\mathrm{v}} = (v_1, v_2, ..., v_k) \tag{1}$$

The visual features contain several object proposals of an image, whereas the semantic features (i.e, image attributes) can describe motions and properties, including nouns, verbs and adjectives. Following [Fang *et al.*, 2015], we use the NOR model as the objective function to compute $p_i^{w_k}$, which is the probability that image i contains word (i.e, attribute)

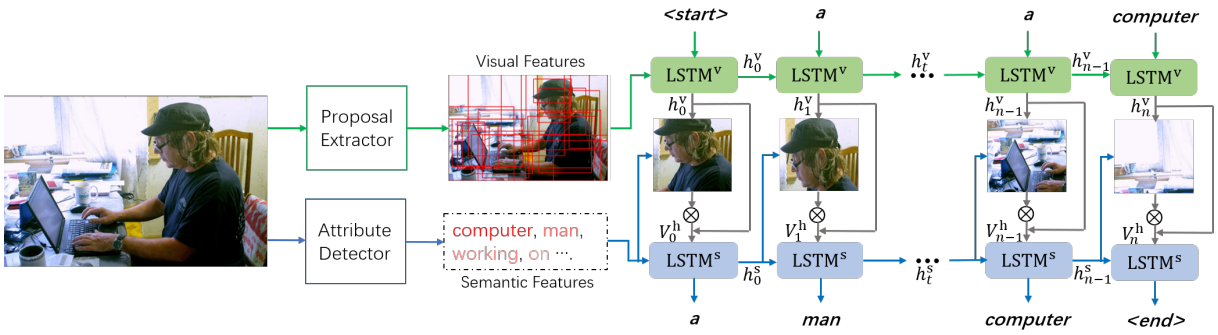

Figure 2: Illustration of the proposed visual-semantic LSTM. The visual features and the semantic features are first extracted by the proposal extractor and the attribute detector separately. The visual cell LSTM$^{\text{v}}$ encodes the overall visual features to a vector representation h_t^{v}, which is combined with the previous generated information h_{t-1}^{s} (upward blue arrow) to find relevant objects. Through an attention gate ($\otimes$), the objects are localized and then serve as the high-level visual features V_t^{h} to be integrated with the high-level semantic features (i.e, attributes) in the semantic cell LSTM$^{\text{s}}$. Thus the next word is determined by both the localized objects and their descriptions.

w_k.

$$p_i^{w_k} = 1 - \prod_{j \in b(i)} (1 - p_{ij}^{w_k}) \tag{2}$$

where $p_{ij}^{w_k}$ is the probability that region j in image i contains word w_k, and $b(i)$ denotes all regions of image i. $p_i^{w_k}$ is 1 if word w_k appears in the ground truth captions of image i, otherwise 0. Let m be the number of attributes, the probability distribution vector of the ith image is defined as the representation of the semantic features A.

$$A = (p_i^{w_0}, p_i^{w_1} ... p_i^{w_{m-1}}) \tag{3}$$

Visual features V^{v} and semantic features A are then fed into the proposed LSTM to be integrated and decoded into the caption.

3.2 Visual-Semantic LSTM

The design of our visual-semantic LSTM is shown in Figure 2. The visual cell LSTM$^{\text{v}}$ encodes the low-level visual features V^{v} while the semantic cell proceeds with the high-level semantic features A. The hidden states of the LSTM cells at time -1 are initialized with the averaged visual features $\bar{V}^{\text{v}}$.

$$h_{-1}^{\text{v/s}} = \tanh(W_{-1}^{\text{v/s}} \bar{V}^{\text{v}}) \tag{4}$$

where $h_{-1}^{\text{v/s}}$ represents the initial hidden state of the visual cell and the semantic cell, respectively, and $W_{-1}^{\text{v/s}}$ is the learned weight.

Visual Cell. In the visual cell, the averaged visual features $\bar{V}^{\text{v}}$ and the word embedding x_t at time t are fed at each time step to inform the network of the visual content.

$$h_t^{\text{v}} = \text{LSTM}^{\text{v}}(W_1^{\text{v}} x_t + W_2^{\text{v}} \bar{V}^{\text{v}}) \tag{5}$$

where h_t^{v} is the hidden state of the visual cell at time t. LSTM$^{\text{v}}$ is the the LSTM function to compute the hidden state in the visual cell and W_1^{v}, W_2^{v} are learned weights.

Object Localization. The encoded visual feature $h_t^{\text{v}}$is utilized to find the relevant object proposals of word x_t using a soft attention mechanism. The attention value of each proposal at time t is computed by the visual cell outputs h_t^{v}, the semantic cell outputs h_{t-1}^{s}, and the visual features V^{v}.

$$a_{t,i} = \text{softmax}_i(W_1^{\text{a}} \tanh(W_2^{\text{a}} h_t^{\text{v}} + W_3^{\text{a}} h_{t-1}^{\text{s}}) + W_4^{\text{a}} V^{\text{v}}) \tag{6}$$

$$c_t = \sum_i v_i a_{t,i} \tag{7}$$

where $a_{t,i}$ is the attention value of the ith proposal at time t, and W_1^{a}, W_2^{a}, W_3^{a}, W_4^{a} are learned weights. The context vector c_t of the given image is the weighted sum of visual feature vector v_i and its corresponding attention value $a_{t,i}$.

To localize the attended object proposals in the image, an attention gate [Wang *et al.*, 2017a] is added based on the current context vector c_t and the encoded visual features h_t^{v}.

$$g_t = \text{sigmoid}([W_1^{\text{g}} h_t^{\text{v}}, W_2^{\text{g}} c_t]) \tag{8}$$

$$V_t^{\text{h}} = g_t \otimes [h_t^{\text{v}}, c_t] \tag{9}$$

where g_t is the additional attention gate and $\otimes$ is the element-wise multiplication operation. W_1^{g} and W_2^{g} are learned weights. V_t^{h} is regarded as the high-level visual feature after object localization.

Semantic Cell. In the semantic cell, high-level information including the localized visual features and the semantic features are further processed and integrated to interpret the image content.

$$h_t^{\text{s}} = \text{LSTM}^{\text{s}}(W_1^{\text{s}} V_t^{\text{h}} + W_2^{\text{s}} A) \tag{10}$$

where V_t^{h} indicates the high-level visual feature and A is the high-level semantic feature. LSTM$^{\text{s}}$ is the LSTM function to compute the hidden state h_t^{s} in the semantic cell with learned weights W_1^{s} and W_2^{s}. The output of the semantic cell h_t^{s} obtains the object location from the visual cell, and the object description from the detected attributes.

The output of the semantic cell is then decoded into the next word x_t. By integrating the low-level visual features V^{v} and the high-level semantic features A in the visual-semantic LSTM, the final features contain the full information of the objects' locations in the image and their corresponding properties. Thus, the next word x_t is computed by a softmax classifier.

$$x_t = \text{softmax}(W^{\text{h}} h_t^{\text{s}}) \tag{11}$$

where h_t^{s} is the output of the semantic cell and W^{h} is the weight to be learned.

3.3 Objective Function

The model parameters are learned by maximizing the probability of the generated caption given the query image. Let w_t denote each word in the caption S, the loss function is the sum of the negative log-likelihood of each word given the visual features V and semantic features A.

$$L = -\sum_{t=1}^{n} \log P(w_t | w_{0...t-1}, (V^{\mathrm{v}}, A)) \tag{12}$$

where n is the length of the sentence, $P(w_t|w_{0...t-1}, (V^{\mathrm{v}}, A))$ is the probability of word w_t given previous words $w_{0...t-1}$ and the visual features V^{v} and semantic features A .

Training the model depending on the ground truth captions as in Equation (12) causes the out-of-context problem [Choi *et al.*, 2008]. This means that the given captions only cover limited content of the image, so objects beyond the sentence are not explored. To ease the problem, we use the evaluation metric as the reward function to train the model as in [Rennie *et al.*, 2017]:

$$L_r = -(r(w_t) - b(w_t)) \log(P(w_t|w_{0..t-1}, (V^{\mathrm{v}}, A))) \tag{13}$$

where $r(w_t)$ is the reward of the sampled word w_t, and $b(w_t)$ is its baseline. In practice, Monte Carlo return is used to compute $r(w_t)$ and the model parameters are updated after generating a complete sentence. To encourage appropriate exploration of the less frequent words in the ground truth captions, we introduce a new sampling strategy [Aman *et al.*, 2018] for the reinforcement learning process.

$$w_t \leftarrow \underset{w_t}{\operatorname{argmin}} \{r(w_t) + \gamma ||h_t - h_t'||_2\} \tag{14}$$

where h_t and h_t' are the hidden states of the true next word w_t and the candidate next word w_t', respectively. $||\cdot||_2$ represents euclidean distance. γ is a constant. In the sampling strategy, the candidate next word w_t' is first sampled according to the given distribution of the vocabulary produced by the model, then the true next word w_t is drawn according to its reward $r(w_t)$ and the distance of the hidden states between w_t and w_t'. With appropriate γ, the sampled word has lower reward but similar state with the candidate word. Thus proper vocabularies can be explored.

4 Experiments

To validate the effectiveness of our model, we conduct experiments on Flickr30K [Young *et al.*, 2014] and MS COCO [Lin *et al.*, 2014] datasets which have 31,783 and 123,287 annotated images, respectively. In both datasets, each image has at least 5 human annotated captions as reference. We use the public available splits [Karpathy and Fei-Fei, 2015] which has 5000 randomly selected images for validation and test. Our vocabulary size is fixed to 10,000 for both datasets including special start sign <BOS> and end sign <EOS>. We report our results with the widely used evaluation metrics: BLUE-1,2,3,4, METEOR and CIDEr, as provided by MS COCO.

Method	B-1	B-2	B-3	B-4	METEOR	CIDEr
VS-LSTM(*w/o* LSTM$^{\mathrm{v}}$)	74.3	57.4	43.2	33.3	26.5	105.1
VS-LSTM(*w/o* LSTM$^{\mathrm{s}}$)	75.1	58.2	43.9	33.5	26.5	105.8
VS-LSTM(Box Proposals)	**76.3**	**59.4**	**44.8**	**34.3**	**26.9**	**110.2**
VS-LSTM(Uniform Grids)	74.5	58.0	43.8	33.1	26.2	105.5
VS-LSTM(RL)	78.4	62.3	47.3	35.8	27.1	119.5
VS-LSTM(RL,$\gamma = 0.1$)	77.4	62.0	46.9	35.4	26.9	116.0
VS-LSTM(RL,$\gamma = 0.3$)	78.1	62.5	47.3	35.2	27.1	118.2
VS-LSTM(RL,$\gamma = 0.5$)	**78.9**	**63.4**	**48.1**	**36.3**	**27.3**	**120.8**
VS-LSTM(RL,$\gamma = 0.7$)	78.4	62.7	47.3	35.8	27.0	119.4
VS-LSTM(RL,$\gamma = 0.9$)	78.2	62.3	47.4	35.9	27.1	119.8

Table 1: Results of the ablation experiments on MS COCO test split. B-n stands for BLEU-n metric. RL indicates reinforcement learning

4.1 Implementation Details

In the CNN mode, the attribute detector is finetuned from VGG16 [Simonyan and Zisserman, 2015] pretrained on ImageNet by changing the last fully-connected layer into a multiple instance loss layer, and trained on MS COCO with the top-1000 frequent words as the attributes. Some common functional words such as *is, to, a* are excluded from the attributes. The visual features are acquired by finetuning ResNet-101 [He *et al.*, 2016] on PASCAL VOC 2012 for the Faster R-CNN model. k is $4,364,359/123,287 = 35$ for MS COCO and $1,179,149/31,783 = 37$ for Flickr 30K, which is the average number of detected proposals of each image in the corresponding dataset. We did not finetune the above features in the training of the LSTM model.

In the LSTM model, the number of hidden nodes of the LSTM is set to 512, with word embedding size of 512. The reward function in reinforcement learning is set to be the CIDEr score. The robust parameter γ of the REINFORCE sampling strategy is set to 0.5 from experimental results. In training, we use the Adam optimizer with learning rate decay and set initial learning rate of 5×10^{-4}. We use 0.5 dropouts of the output and feed back 5% of sampled words every 4 epochs until reaching a 25% feeding back rate [Bengio *et al.*, 2015]. A batch normalization layer [Ioffe and Szegedy, 2015] is added to the beginning of the image encoder to accelerate training with mini-batch size of 50. Additionally, for faster convergence from random initial state, we adopt the orthogonal initializer instead of the random Gaussian initializer.

4.2 Results

In this subsection, we first analyze the impact of each part in our model by conducting ablation experiments, then present some visualized results. We also compare our approach with the state-of-the-art methods to show its outperformance.

Ablation Experiments

We conduct several ablation experiments to see the importance of each part in our model in Table 1. To study how the low-level visual features and the introduced sampling strategy influence the results separately, we divide the experiments into two parts: with reinforcement learning (RL) and without RL.

Firstly, with box proposals as the visual features (**VS-LSTM(Box Proposals)**), the visual cell (**VS-LSTM(*w/o* LSTM$^{\mathrm{v}}$)**) and the semantic cell (**VS-LSTM(*w/o* LSTM$^{\mathrm{s}}$)**) are removed, respectively, as the baseline models in Table

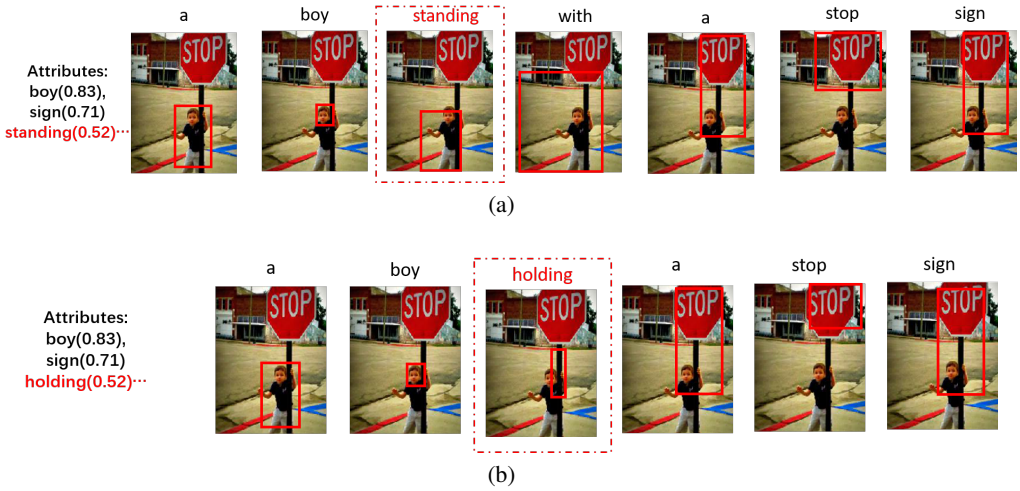

Figure 3: A visualized example showing that the localized objects are consistent with the detected semantic features. The digit in the brackets beside the attribute indicates its probability. After replacing the attribute *standing* in figure (a) with *holding* in figure (b) while keeping others unchanged, the corresponding box proposal moves from the boy's body to his hand.

Figure 4: Generated examples of our model. **VS-LSTM(RL,γ** is the proposed model and **GT** represents the ground truth captions. Here γ is set to 0.5. The first two pictures present successful examples and the last two pictures show the failed ones.

1.The results prove that **VS-LSTM(Box Proposals)** acquires clear improvements by processing low-level features in the visual cell and high-level features in the semantic cell successively. Besides, since the output uniform grids of the CNN model can also be regarded as the low-level visual features, we conduct an ablation experiment with the output uniform grids as the visual features on MS COCO. **VS-LSTM(Uniform Grids)** is our model with the 14×14 output feature map of ResNet-101 as the low-level visual features. From the results in Table 1, **VS-LSTM(Box Proposals)** with the object proposals as the visual features performs better than **VS-LSTM(Uniform Grids)**. This can be explained because the box proposals which segment objects from the image make it easier to localize the corresponding objects in the visual cell, whereas the output uniform grids blur the boundaries between objects and thus cause misrecognition of the localized objects.

Sencondly, ablation experiments about how the introduced sampling strategy of RL influences the results are shown in Table 1. We use box proposals as the visual features in the following experiments since they prove to have better performance in our model. **VS-LSTM(RL)** is our model with the same sampling strategy in [Rennie *et al.*, 2017] and **VS-LSTM(RL, γ)** is our model by importing the state perturbation term into the sampling strategy with different γ values. We can tell that the perturbation term with $\gamma = 0.5$ is optimal to further improve the RL results. This is because small γ diverges from the optimal solution whereas large γ reduces exploration of the vocabularies. By sampling a word which has lower reward but similar state with appropriate γ, the model can expand the exploration of the whole vocabularies as well as balance the training between the frequent words and the less frequent words.

Visualized Results

To see which objects the model focuses on when generating each word, we visualize the results in Figure 3. The bounding box with the highest attention value is selected as the most relevant proposal of the corresponding word. We can tell that the selected proposals by the visual cell are quite consistent with the detected attributes. For example, after we replace the attribute *standing* in Figure 3(a) with *holding* in Figure 3(b), the corresponding proposal moves its position from the boy's

Method	MS COCO						Flickr30K					
	B-1	B-2	B-3	B-4	METEOR	CIDEr	B-1	B-2	B-3	B-4	METEOR	CIDEr
Hard-ATT[Xu *et al.*, 2015]	71.8	50.4	35.7	25.0	23.0	-	66.9	43.9	24.0	15.7	15.3	-
Semantic-ATT^2 [You *et al.*, 2016]	70.9	53.7	40.2	30.4	24.3	-	64.7	46.0	32.4	23.0	-	-
CNN$_L$+RHN [Gu *et al.*, 2016]	72.3	55.3	41.3	30.6	25.2	98.9	73.8	56.3	41.9	30.7	20.6	-
SCA-CNN [Chen *et al.*, 2017]	71.9	54.8	41.1	31.1	25.0	95.2	66.2	46.8	32.5	22.3	19.5	44.7
Ada-ATT [Lu *et al.*, 2017]	74.2	58.0	43.9	33.2	26.6	108.5	67.7	49.4	35.4	25.1	20.4	53.1
ATT-kCC [Mun *et al.*, 2017]	74.9	58.1	43.7	32.6	25.7	102.4	-	-	-	-	-	-
LSTM-A$_5$ [Yao *et al.*, 2017]	73.4	56.7	43.0	32.6	25.4	100.2	-	-	-	-	-	-
Att-CNN+LSTM [Wu *et al.*, 2016]	74	56	42	32	26	94	73.0	55.0	40.0	28.0	-	-
SCN-LSTM [Gan *et al.*, 2017]	74.1	57.8	44.4	34.1	26.1	104.1	74.7	55.2	40.3	28.8	22.3	-
MAT [Liu *et al.*, 2017]	73.1	56.7	42.9	32.3	25.8	105.8	-	-	-	-	-	-
Skel-Attr-LSTM [Wang *et al.*, 2017b]	74.2	57.7	44.0	33.6	26.8	107.3	-	-	-	-	-	-
$*$ 4 Att2in [Rennie *et al.*, 2017]	-	-	-	34.8	26.9	115.2	-	-	-	-	-	-
Ours-VS-LSTM(Box Proposals)	76.3	59.4	44.8	34.3	26.9	110.2	74.1	56.6	42.5	30.1	22.5	62.7
$*$ Ours-VS-LSTM(RL,$\gamma = 0.5$)	**78.9**	**63.4**	**48.1**	**36.3**	**27.3**	**120.8**	**75.5**	**57.1**	**42.9**	**31.7**	**22.9**	**71.5**

Table 2: Results on MS COCO and Flickr30K test split in comparison with other methods. B-n stands for BLEU-n metric. **-** represents for unknown result. Methods marked with $*$ adopt reinforcement learning for CIDEr optimization.

Method	c5							c40						
	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr
Hard-ATT	70.5	52.8	38.3	27.7	24.1	51.6	86.5	88.1	77.9	65.8	53.7	32.2	65.4	89.3
Semantic-ATT^2	73.1	56.5	42.4	31.6	25.0	53.5	94.3	92	81.5	70.9	59.9	33.5	68.2	95.8
SCA-CNN	71.2	54.2	40.4	30.2	24.4	52.4	91.2	89.4	80.2	69.1	57.9	33.1	67.4	92.1
Ada-ATT	74.8	58.4	44.4	33.6	26.4	55.0	104.2	92.0	84.5	74.4	63.7	35.9	70.5	105.9
ATT-kCC	74.3	57.5	43.1	32.1	25.5	53.9	98.7	91.5	83.2	72.2	60.7	34.1	68.6	100.1
LSTM-A$_3$	78.7	62.7	47.6	35.6	**27.0**	56.4	116.0	93.7	86.7	76.5	65.2	35.4	70.5	118.0
Att-CNN+LSTM	73	56	41	31	25	53	92	89	80	69	58	33	67	93
SCN-LSTM	74.0	57.5	43.6	33.1	25.7	54.3	100.3	91.7	83.9	73.9	63.1	34.8	69.6	101.3
MAT	73.4	56.8	42.7	32.0	25.8	54.0	102.9	91.1	83.1	72.7	61.7	34.8	69.1	106.4
4 Att2in	78.1	61.9	47.0	35.2	**27.0**	56.3	114.7	93.1	86.0	75.9	64.5	**35.5**	**70.7**	116.7
Ours-VS-LSTM(RL,$\gamma = 0.5$)	**78.8**	**62.8**	**47.9**	**35.9**	**27.0**	**56.5**	**116.6**	**94.6**	**87.5**	**77.3**	**66.3**	35.3	70.3	**119.5**

Table 3: Results on the online MS COCO test server. All metrics are reported using c5 and c40 references.

body to his hand. This indicates that by localizing and describing objects successively in an unified network, the generated attention can be more specific as well as accurate.

Some examples of the generated captions are shown in Figure 4. The images are selected from the 5000 images in the public available test split [Karpathy and Fei-Fei, 2015] on MS COCO. The first two pictures present successful generated captions while the last two pictures show the typical failed examples. By localizing and describing objects successively in the visual-semantic LSTM, our model can find primary objects in the image and describe them with detailed attributes, such as *red and white plane* in the first picture. However, since the detected proposals and attributes may not be accurate, some objects are misrecognized and left out in the generated captions. For example, the model misrecognizes the *wagon* as *truck* in the third picture, and misses the *cat* in the fourth picture.

Comparison with Other Methods

In Table 2, we choose our model **VS-LSTM(RL,** $\gamma = 0.5$**)**) that has the best performance in the ablation experiments to compare with the recent state-of-the-art results. Since performing RL for CIDEr optimization can bring a boost in performance [Rennie *et al.*, 2017], for fair comparison, we present our results without RL (**VS-LSTM(Box Proposals)**) and with RL (**VS-LSTM(RL,** $\gamma = 0.5$**)**). Methods marked with $*$ adopt RL for CIDEr optimization. We can see that the proposed model outperforms the other results on most metrics with and without RL.

In addition, we also report our results on the online MS COCO test server in Table **??**. Both our model **VS-LSTM(RL,** $\gamma = 0.5$**)** and the model **4Att2in** adopt RL for CIDEr optimization, but our gain drops to 116.9 CIDEr on the test server. Our **VS-LSTM(RL,** $\gamma = 0.5$**)** is a single model whereas **4Att2in** uses ensemble models. In addition, we set γ value at an interval of 0.2, and thus $\gamma = 0.5$ may not be optimal. Still, the proposed **VS-LSTM(RL,** $\gamma = 0.5$**)** outperforms the compared methods on most metrics.

5 Conclusion

In this paper, A novel visual-semantic LSTM based model named VS-LSTM is proposed. By first localizing the objects in the visual cell with low-level features, and then describing them in the semantic cell with high-level features, VS-LSTM automatically recognizes the objects with their corresponding states when generating each word. In addition, in the reinforcement learning process, by importing a state perturbation term, the proposed model can explore proper vocabularies and balance the training between the frequent words and the less frequent words. Experimental results on MS COCO and Flickr30K prove the effectiveness of our model with respect to existing methods.

References

[Aman *et al.*, 2018] Sinha Aman, Namkoong Hongseok, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *ICLR*, 2018.

[Anderson *et al.*, 2017] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *arXiv preprint arXiv:1707.07998*, 2017.

[Bengio *et al.*, 2015] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*, 2015.

[Chen *et al.*, 2017] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017.

[Choi *et al.*, 2008] Myung Jin Choi, Antonio Torralba, and Alan S. Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 2008.

[Fang *et al.*, 2015] Hao Fang, John C. Platt, C. Lawrence Zitnick, Geoffrey Zweig, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollar, and Jianfeng Gao. From captions to visual concepts and back. In *CVPR*, 2015.

[Gan *et al.*, 2017] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *CVPR*, 2017.

[Gu *et al.*, 2016] Jiuxiang Gu, Gang Wang, Jianfei Cai, and Tsuhan Chen. An empirical study of language CNN for image captioning. In *ICCV*, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

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

[Lin *et al.*, 2014] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[Liu *et al.*, 2017] Chang Liu, Fuchun Sun, Changhu Wang, Feng Wang, and Alan Yuille. Mat: A multimodal attentive translator for image captioning. In *IJCAI*, 2017.

[Lu *et al.*, 2017] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

[Mao *et al.*, 2015] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-RNN). In *ICLR*, 2015.

[Mun *et al.*, 2017] Jonghwan Mun, Minsu Cho, and Bohyung Han. Text-guided attention model for image captioning. In *AAAI*, 2017.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[Ren *et al.*, 2017] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*, 2017.

[Rennie *et al.*, 2017] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Sutton and Barto, 1998] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[Wang *et al.*, 2017a] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *ACL*, 2017.

[Wang *et al.*, 2017b] Yufei Wang, Zhe Lin, Xiaohui Shen, Scott Cohen, and Garrison W. Cottrell. Skeleton key: Image captioning by skeleton-attribute decomposition. In *CVPR*, 2017.

[Wu *et al.*, 2016] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. What value do explicit high level concepts have in vision to language problems? In *CVPR*, 2016.

[Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[Yao *et al.*, 2017] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

[You *et al.*, 2016] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[Young *et al.*, 2014] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *ACL*, 2014.