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 relevant 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.

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

![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 utilizes the visual features to localize the objects in the image, whilst the semantic cell LSTM further integrates the localized objects with their attributes to generate corresponding word.](image)
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. [You 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 features 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 \( V^v \).

\[
V^v = (v_1, v_2, ..., v_k)
\]

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 \( p_i^{w^v} \), 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 $V$ encodes the overall visual features to a vector representation $h^v_t$, which is combined with the previous generated information $h^v_{t-1}$ (upward blue arrow) to find relevant objects. Through an attention gate ($\odot$), the objects are localized and then serve as the high-level visual features $V^h$ to be integrated with the high-level semantic features (i.e., attributes) in the semantic cell LSTM $A$. Thus the next word is determined by both the localized objects and their descriptions.

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

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 $i$. $p_{ij}^{w_k} = 1$ 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 $i$th image is defined as the representation of the semantic features $A$.

$$A = (p_1^{w_0}, p_1^{w_1} \ldots p_1^{w_{m-1}}) \quad (3)
$$

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 $V$ 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 $t-1$ are initialized with the averaged visual features $\bar{V}^v$.

$$h^v_{t-1} = \tanh(W^v_1 \bar{V}^v) \quad (4)
$$

where $h^v_{t-1}$ represents the initial hidden state of the visual cell and the semantic cell, respectively, and $W^v_1$ is the learned weight.

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

$$h^v_t = \text{LSTM}^v(W^v_1 x_t + W^v_2 \bar{V}^v) \quad (5)
$$

where $h^v_t$ is the hidden state of the visual cell at time $t$. LSTM$^v$ is the the LSTM function to compute the hidden state in the visual cell and $W^v_1$, $W^v_2$ are learned weights.

**Object Localization.** The encoded visual feature $h^v_t$ 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 $h^v_t$, the semantic cell outputs $h^s_{t-1}$, and the visual features $V^v$. The encoded visual feature $h^v_t$ is regarded as the high-level visual feature after object localization.

$$a_{t,i} = \text{softmax}(W^v_1 h^v_t + W^v_3 h^s_{t-1} + W^v_4 V^v) \quad (6)
$$

$$c_t = \sum_i v_i a_{t,i} \quad (7)
$$

where $a_{t,i}$ is the attention value of the $i$th proposal at time $t$, and $W^v_1$, $W^v_2$, $W^v_3$, $W^v_4$ 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 $h^v_t$.

$$g_t = \text{sigmoid}(W^v_2 h^v_t, W^v_3 c_t) \quad (8)
$$

$$V^h_t = g_t \odot [h^v_t, c_t] \quad (9)
$$

where $g_t$ is the additional attention gate and $\odot$ is the element-wise multiplication operation. $W^v_2$ and $W^v_3$ are learned weights. $V^h_t$ 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.

$$h^s_t = \text{LSTM}^s(W^s_1 h^v_t + W^s_2 A) \quad (10)
$$

where $h^s_t$ indicates the high-level visual feature and $A$ is the high-level semantic feature. LSTM$^s$ is the LSTM function to compute the hidden state $h^s_t$ in the semantic cell with learned weights $W^s_1$ and $W^s_2$. The output of the semantic cell $h^s_t$ 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.

$$x_t = \text{softmax}(W^s h^s_t) \quad (11)$$
where $h^s_t$ 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 $w_t \in S$ 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 $A$.

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

where $n$ is the length of the sentence, $P(w_t|w_{0...t-1}, (V^v, A))$ 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]:

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

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.

$$w_t \leftarrow \arg \min_{w_t} \{r(w_t) + \gamma ||h_t - h_t'||_2\}$$

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. $\gamma$ 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 $\gamma$, 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.

### 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 $\gamma$ 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 $5 \times 10^{-4}$. 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)) and the semantic cell (VS-LSTM(w/o LSTM)) are removed, respectively, as the baseline models in Table 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.

Secondly, 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 γ = 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.

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. Red and white plane flying in the sky.
2. A man sitting in a chair with a woman tying a tie.
3. A man putting a tie on a man as he sits at a table.
4. A man standing next to a truck.
5. A dog laying on the back of a boat.
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, γ = 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, γ = 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 2. Both our model VS-LSTM(RL, γ = 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, γ = 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 γ = 0.5 may not be optimal. Still, the proposed VS-LSTM(RL, γ = 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


