Image Captioning via Hierarchical Attention Mechanism and Policy Gradient Optimization

Shiyang Yan[a,*], Yuan Xie[b,c,e], Fangyu Wu[d,f], Jeremy S. Smith[d], Wenjin Lu[f], Bailing Zhang[b,c]

[a] *School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, United Kingdom*
[b] *The Institute of Advanced Artificial Intelligence in Nanjing, Nanjing, China*
[c] *Horizon Robotics, Beijing, China*
[d] *Electrical Engineering and Electronic, University of Liverpool, Liverpool, United Kingdom*
[e] *Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[f] *Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China*

Abstract

Automatically generating the descriptions of an image, i.e., image captioning, is an important and fundamental topic in artificial intelligence, which bridges the gap between computer vision and natural language processing. Based on the successful deep learning models, especially the CNN model and Long Short Term Memories (LSTMs) with attention mechanism, we propose a hierarchical attention model by utilizing both of the global CNN features and the local object features for more effective feature representation and reasoning in image captioning. The generative adversarial network (GAN), together with a reinforcement learning (RL) algorithm, is applied to solve the exposure bias problem in RNN-based supervised training for language problems. In addition, through the automatic measurement of the consistency between the generated caption and the image content by the discriminator in the GAN framework and

[*] Corresponding author
Email addresses: elyotyan@gmail.com (Shiyang Yan), yuan.xie@ia.ac.cn (Yuan Xie), Fangyu.Wu@xjtlu.edu.cn (Fangyu Wu), J.S.Smith@liverpool.ac.uk (Jeremy S. Smith), Wenjin.Lu@xjtlu.edu.cn (Wenjin Lu), bailing.zhang@horizon.ai (Bailing Zhang)

RL optimization, we make the finally generated sentences more accurate and natural. Comprehensive experiments show the improved performance of the hierarchical attention mechanism and the effectiveness of our RL-based optimization method. Our model achieves state-of-the-art results on several important metrics in the MSCOCO dataset, using only greedy inference.

Keywords: Image captioning, Hierarchical attention mechanism, Generative adversarial network, Reinforcement learning, Policy gradient

1. Introduction

Naturalistic description of an image is one of the primary goals of computer vision, which has recently received much attention in the field of artificial intelligence recently. It is a high-level task and much more complicated than some fundamental recognition tasks, e.g., image classification [1] [2] [3] [4], image retrieval [5] [6] [7], object detection and recognition [8] [9] [10]. This requires the system to comprehensively understand the content of an image and bridge the gap between the image and the natural language. Automatically generating image descriptions is useful in multimedia retrieval, and image understanding.

Some pioneering research has been carried out in generating image descriptions [11] [12]. However, as pointed out in [13], most of these models often rely on hard-coded visual concepts and sentence templates, which limits their generalization capability. Recently, with the rapid development of deep learning in image recognition and natural language processing, the current trend of image captioning approaches [14] is to follow the encoder-decoder framework, which shares the similarity with that in neural machine translation [15]. Most of these approaches represented the image as a single feature vector from the top layer of a pre-trained convolutional neural network (CNN) and cascaded recurrent neural network (RNN) to generate languages.

In fact, the tasks like image captioning and machine translation can be considered as a structured output problem where the task is to map the input to an output that possesses its own structure, as stated in [16]. An inherent challenge in these tasks is the structure of the output is closely related to the structure of the input. Hence, a key problem in these tasks is alignment [16]. Take neural machine translation for example, [17] trained a neural model to softly align the output to the input for machine translation. Subsequent research [18] applied the visual attention model to address this problem in image captioning, with much improvement. The visual attention mechanism is to dynamically select the relevant receptive fields in the CNN features to facilitate the image description generation, which, in other words, is to align the output words to spatial regions of the source image. In this paper, we also employ the visual attention mechanism for image captioning.

Nevertheless, natural language often consists of very meticulous descriptions, which correspond to the fine-grained objects of an image. As pointed out by [19], there are certain limitations of the most existing neural model-based schemes due to the mere use of the global feature representation in the image level. Some of the fine-grained objects might not to be recognized by only relying on the global image features. In this paper, we propose to use a pre-trained image detection model, i.e., Faster RCNN [10], to retrieve the fine-grained image features from the top detected objects. These fine-grained object features, are able to provide complementary information for the global image representation, which will be proved in the experiments. In terms of the model structure, the object features are also processed by a visual attention mechanism, and are added to the original model to form a hierarchical feature representation and hence it is able to generate more meticulous descriptions.

In addition to the improvement of the image feature representation, we also consider to improve the current language model, which is widely used in neural machine translation and image captioning. An issue with most of the previous language model is the training framework, namely, the RNN using Maximum Likelihood Estimation (MLE) to generate image descriptions. As pointed out in [20], the MLE approaches suffer from the so-called exposure bias in the inference stage: the model generates a sequence iteratively and predicts the next token based on the previously predicted ones that may never be observed in the training data. In image description generation, the MLE also suffers from a problem that the generated languages do not correlate well with a human assessment of quality [21].

Instead of only relying on the MLE, an alternative scheme is the generative adversarial network (GAN) [22]. GAN was first proposed to generate realistic images. The GAN learns generative models without explicitly defining a loss function from the target distribution. Instead, GAN introduces a discriminator network which tries to differentiate real samples from generated samples. The whole network is trained using an adversarial training strategy. One can subsequently build a discriminator to judge how realistic are the samples generated by the description generator. The role of the caption generator, in this model, is similar with that of the the generator in the conditional GAN [23], which is conditioned on the image features.

However, language generation is a discrete process. Directly providing the discrete samples as inputs to the discriminator does not allow the gradients to be back propagated through them. The reinforcement learning (RL) [24] framework provides a solution to estimate the gradients of the discontinuous units. The RL framework, when dealing with sequence generation, has the problem of lacking the intermediate reward, as discussed in [25]. The reward value can only be obtained when the whole sequence is generated. This is not suitable since what we want is the long-term reward of each intermediately generated token, so the whole sequence better optimized.

In the proposed scheme, the discriminator takes into account not only the differences between the generated captions and the reference captions but also the consistencies between captions and image features. Through the evaluation of the discriminator, the networks can better compensate for some unrealistic captions which might be generated under the MLE training. However, to deal with the discreteness of language, we treat the image captioning generator as an agent of RL. The feedbacks from the discriminator are considered as the rewards for the generator. To update the parameters of the image description generator in this framework, we consider the generator as a stochastic parameterized policy. We train the policy network using Policy Gradient [26], which naturally solve the differential difficulties in conventional GAN. Also, to solve the problem of lacking intermediate rewards, we borrow the idea from the famous "AlphaGo" program [27] in which a Monte Carlo roll-out strategy is applied to sample the expected long-term reward for an intermediate move. If we consider the sequence token generation as the the action to be taken in RL, we can apply a similar Monte Carlo roll-out strategy to obtain the intermediate rewards. [25] has successfully applied the Monte Carlo roll-out in sequence generation. In this paper, we use a similar sampling method to deal with intermediate rewards during the process of caption generation.

To summarize, our contribution in this paper is threefold:

- We propose a hierarchical attention mechanism to reason on the global features and the local object features for image captioning.
- The policy gradient algorithm combined with the GAN is proposed for the training and optimization of the language model, with improvements over MLE training scheme.
- Through comprehensive experiments, we validate the proposed algorithm and comparable results with current state-of-the-art methods are achieved on the MSCOCO dataset.

2. Related Work

2.1. Deep Model-based Image Captioning

Promoted by the recent success of deep learning network in image recognition tasks and machine translation, the research on generating image description or image captioning has made remarkable progress [28] [13] [12] [29] [14] [30]. As mentioned above, most of the previously proposed approaches consider the image description generation as a translation process, mainly by borrowing the idea of the encoder-decoder framework [31] from neural machine translation [15]. Generally, this paradigm considers a deep CNN model as the image encoder, which maps the image into a static feature representation, and a RNN as a decoder to decode this static representations to an image description. The whole framework is trained using supervised learning under MLE. The generated description should be grammatically correct and match the content of the image.

Specifically, Karpathy et al. [13] proposed an alignment model through a multi-modal embedding layer. This model is able to align parts of a description with the corresponding regions of the image, which attracts significant attention. Jia et al. [29] proposed a variation of LSTM, called gLSTM, for the image captioning task to mainly tackle the problem of losing track of the image content. This model includes the semantic information along with the whole image as inputs to generate captions. Donahue et al. [30] applied both of the convolutional layers and recurrent layers to form a Long-term Recurrent Convolutional Network (LRCN) for visual recognition and description.

Bahdanau et al. [17] pointed out that a potential problem in this approach is that the model should compress all the necessary information of a source sentence into a fixed-length representation. This may make it difficult for the neural network to cope with long sentences. The static feature representation in the encoder-decoder framework, for both of machine translation and image captioning, cannot automatically retrieve relevant information from the source and thus at last influence the final performance. In neural machine translation, Bahdanau et al. [17] proposed a kind of soft attention mechanism for machine translation, which enables the decoder to automatically focus on the relevant parts of the source sentence. In computer vision, the attention mechanism has long been the focus of much research [17] [32] [33] since human perception does not tend to process a whole scene in its entirety at once but applies some mechanisms to selectively focus on the information needed. A comprehensive study for hard attention bound with reinforcement learning and soft attention for the task of image captioning was published by Xu et al. [18].

Yao et al. [34] tackled the video captioning task through capturing global temporal structures among video frames with a temporal attention mechanism, which makes the model dynamically focus on the key frames that are more relevant with the predicted word. Attention Models (ATT) developed by You et al. [35] first extracted semantic concept proposals and fused them with RNNs into hidden states and outputs. This method used K-NN, multi-label ranking to extract semantic concepts or attributes and fused these concepts into one vector using an attention mechanism. Similarly, Yao et al. [36] embedded attributes with image features into a RNN with various methods to boost the image captioning performance. Recently, Chen et al. [37] proposed to combine the spatial attention and the channel-wise attention mechanism for image captioning, with improved results. Alternatively, Li et al. [19] proposed a global-local attention mechanism to include local features extracted from the top detected objects from a pre-trained object detector. Inspired by [19], we also include the local features from top detected objects. However, we build a hierarchical model whilst they treated local and global features equivalently.

2.2. Policy Gradient Optimization for Image Captioning

Another approach to boost the performance of language tasks is to compensate the so-called exposure bias problem in RNN-based MLE learning. As pointed out in [38], RNNs are trained by MLE, which essentially minimized the KL-divergence between the distribution of target sequences and the distribution defined by the model. This KL-divergence objective tends to favour a model that overestimates its smoothness, which can lead to unrealistic samples [39].

In order to tackle the problems and generate more realistic image descriptions, some researches directly use evaluation metrics such as BLEU [40], METEOR [41] and ROUGE [42] as the reward signal and build the model under the RL framework. For instance, Ranzato et al. [43] is the first research using the policy gradient algorithm in a RNN-based sequence model, in which a REINFORCE-based approach was used to calculate the sentence-level reward and a Monte-Carlo technique was employed for training. Liu et al. [44] studied several linear combinations of the evaluation metrics and proposed to use a linear combination of SPICE [45] and CIDEr [46] as the reward signal and apply a policy gradient algorithm to optimize the model, with improved results. This research used a Monte-Carlo roll-out strategy to obtain the intermediate reward during the process of description generation. More recently, Bahdanau et al. [47], instead of sentence-level reward in the training, applied the token-level reward in temporal difference training for sequence generation.

As discussed previously, the GAN [22] estimates a difference measure using a binary classifier, called a discriminator, to discriminate between the target samples and generated samples. GANs rely on back-propagating these difference estimates through the generated samples to train the generator to minimize these differences. Hence, the whole network in GAN is trained in an adversarial way. The GAN was originally proposed to generate naturalist images [22] [23] [48] [49]. Directly applying a GAN for the language problem is impossible since sequences are composed of discrete elements in many application areas such as machine translation and image captioning.

A possible solution to tackle the discreteness problem of language is to use the Gumbel-Softmax approximation [50] [51]. For instance, Shetty et al. [52] use a GAN to generate more realistic and accurate image descriptions with the aid of Gumbel-Softmax to deal with the discontinuousness issue in language processing. Another more general solution is to borrow an idea from the RL framework, in which the feedback from the discriminator is considered as the reward for the language generator. Dai et al. [21] built a model based on conditional GAN to generate diverse and naturalistic image descriptions and paragraphs, which utilizes a policy gradient for optimization. Yu et al. [25] proposed a model called SeqGAN, which unified the GAN framework and RL learning problem, this has recently received much attention [53] [54]. They propose a three steps training strategy, which includes the pre-training the generator, pre-training the discriminator and the final adversarial training. In this paper, inspired by the SeqGAN, we propose to use a discriminator to judge the fitness of the generated image descriptions with reference to the image content and apply the policy gradient optimization technique [26] to train the model. Unlike the original SeqGAN, our discriminator not only cares about the differences between the target language and model-generated language but also considers the coherence of the language with the image content.

3. Approach

In this section, we describe the proposed method based on two parts: the hierarchical attention mechanism and the policy gradient optimization algorithm.

3.1. Hierarchical Attention Mechanism

The hierarchical attention mechanism consists of two parts: a spatial attention mechanism which corresponds to global CNN features and a local attention mechanism which corresponds to object features.

The spatial attention mechanism is based on the model in [18]. Specifically, the model comprises of an encoder and a decoder. We use a convolutional neural network pre-trained on the ImageNet dataset [55] in order to extract a set of

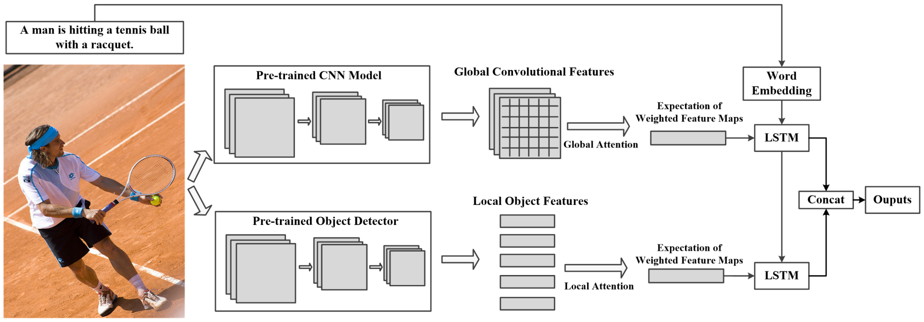

A man is hitting a tennis ball with a racquet.
Pre-trained CNN Model
Global Convolutional Features
Word Embedding
Expectation of Weighted Feature Maps
Global Attention
LSTM
Concat
Ouputs
Local Object Features
Pre-trained Object Detector
Expectation of Weighted Feature Maps
Local Attention
LSTM

Figure 1: The hierarchical attention model structure: The CNN encoder and the object detector extracts the global and local features, respectively. These two types of features are forwarded to the LSTM models with the global and the local attention mechanisms. The outputs from the two LSTM models are concatenated and decoded to words.

convolutional features. These features, denoted as $a = \{a_1, ..., a_L\}$, correspond to certain portions of the 2-D image. We extract convolutional features instead of fully connected ones in order to build a spatial attention mechanism since convolutional features have a spatial layout.

The Long-short Term Memory (LSTM) network, originally proposed by Hochreiter and Schmidhuber in [56], is applied as the language decoder because of its superior performance in natural language processing.

$$c_t, h_t = LSTM(z_{t-1}, c_{t-1}, h_{t-1}) \tag{1}$$

In Equation 1, c_t and h_t are the memory cells and hidden states of the LSTM, respectively. z_t is the context vector, which can be processed by the soft attention mechanism and is able to capture visual information associated with a certain input location. The soft attention mechanism has to automatically allocate adaptive weights for the image locations to facilitate the task at hand.

$$a = Average\_Pooling(a_i), i = 1...L \tag{2a}$$

$$e_{ti} = MLP(a, h_{t-1}) \tag{2b}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{L} exp(e_{tk})} \tag{2c}$$

$$z_t = \sum_{i=1}^{L} \alpha_{ti} a_i \tag{2d}$$

where $a_i \in \{a_1, ..., a_L\}$. The first and second equations of Equation 2 map the image features from each location, along with information from the hidden state, into an adaptive weight, which indicates the importance of each image location for the recognition. Then, we normalize the adaptive weights into a probability value in the range of 0 and 1 using the Softmax function. Once these weights (summed to 1) are computed, we element-wisely multiply the weights vector α_t with image feature vector a and sum them to the context vector z_t, which can be expressed as in the last equation of Equation 2. This can be seen as the expectation of weighted features maps. Then the context vector z_t is forwarded to the LSTM network to generate captions, as described in Equation 1. This soft attention mechanism is able to adaptively select the relevant visual parts of the given image features and thus facilitate the recognition.

The local attention mechanism is formulated using object features and another LSTM model. We use a pre-trained object detector to retrieve the top N detected object features, which are denoted as $d = \{d_1, ..., d_N\}$. We then use another LSTM model with soft attention to allocate adaptive weights to each of these features.

$$z_t^d = Concat(\sum_{i=1}^{N} \alpha_{ti}^d d_i, h_{t-1}) \tag{3}$$

Equation 3 demonstrates that the context vector for local attention model catching information from both the local features and the global attention mechanism, where *Concat* indicates the concatenation operation of the features. This context vector is then forwarded to a second LSTM model.

The two LSTM models, denoted as $LSTM^G$ for the global features and $LSTM^L$ for the local features are jointly trained to map the hierarchical feature representation with language. $LSTM^L$ is at a higher level, which can be used to decode the hidden states for the final outputs. However, the gradient vanishing problem cannot be avoided if we only use the hidden states from $LSTM^L$ to decode information. Inspired by [3] in which a shortcut in network connections is applied to solve the gradient vanishing problem, we concatenate the hidden states from $LSTM^G$ and $LSTM^L$ to decode and map the hidden states to language vectors, which can be seen in Equation 4.

$$h_t^{output} = Concat(h_t, h_t^d) \tag{4a}$$

$$logits = W_p h_t^{output} \tag{4b}$$

$$P(s_t|I, s_0, s_1, s_2, ..., s_{t-1}) = Softmax(logits) \tag{4c}$$

In MLE training, if the length of a sentence is T, the loss function can be formulated as in Equation 5, which is the sum of the log likelihood of each word.

$$Loss = \sum_{i=0}^{T} log(p(s_t|I, s_0, s_1, s_2, ..., s_i)) \tag{5}$$

3.2. Policy Gradient Optimization

In addition to only using the MLE to train the image caption generator, to alleviate the previously discussed exposure bias problem in RNN-based MLE

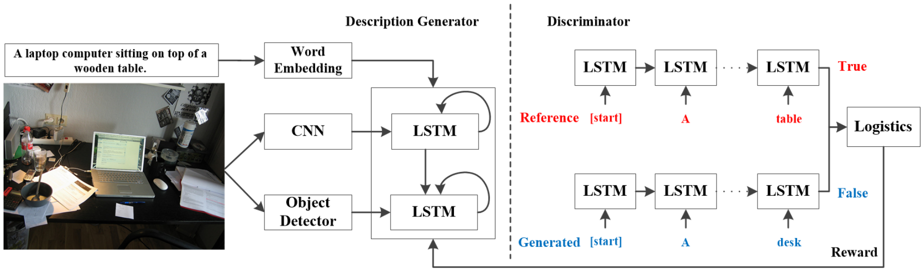


Figure 2: Policy Gradient optimization with a discriminator to evaluate the similarity between the generated sentence and the reference sentence.

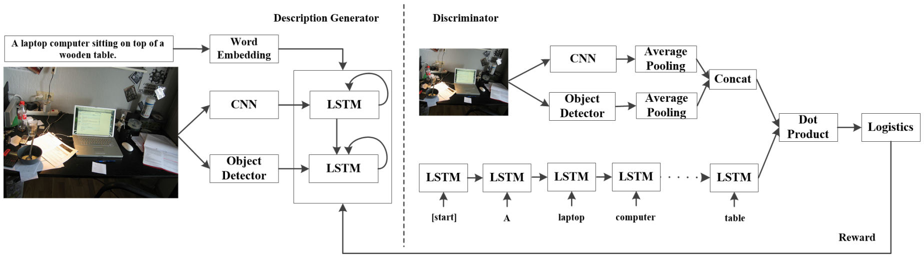


Figure 3: Policy Gradient optimization with a discriminator to evaluate the coherence between the generated sentence and the image contents.

training as discussed previously, we also apply a policy gradient optimization algorithm in the RL framework to increase the quality of the generated descriptions.

We feed both of the generated descriptions and the reference descriptions to the discriminator. The level of coherence of the descriptions and image content is calculated by the dot product, which is forwarded to the discriminator, as described in Fig. 3. This operation is to consider the coherence between certain captions (sequences) and corresponding image features, which is able to make the generated captions more realistic and naturalistic. The reference sequences are labeled as true whilst the generated sequences are labeled as false during the training of the discriminator. The model is also a LSTM network with

Softmax Cross Entropy loss. Hence, the discriminator outputs the probabilities of a sample being true. These probabilities, are then considered as the reward signal in the RL framework, to be utilized in the Policy Gradient algorithm for updating the parameters of the image caption generator.

Following [26], the objective of the policy network $G_\theta(y_t|y_{1:t-1})$ (the image caption generator), is to generate a sequence from the start state S_0 to maximize its expected long-term reward as described by Equation 6:

$$J(\theta) = E[R_T|s_0, \theta] = \sum_{y_1 \in Y} G_\theta(y_1|s_0) \cdot Q_{D_\theta}^{G_\theta}(s_0, y_1) \tag{6}$$

where R_T is the reward for a complete sequence. $Q_{D_\theta}^{G_\theta}(s, y)$ is the action-value function of a language sequence, which is defined as the expected accumulative reward starting from state s, taking a certain action, and then following policy G_θ.

The action-value function is estimated using the REINFORCE algorithm [57] and considers the probability of being real generated by the discriminator as a reward, which can be defined as in Equation 7.

$$Q_{D_\theta}^{G_\theta}(a = y_T, s = Y_{1:T-1}) = D_\theta(Y_{1:T}) \tag{7}$$

As can be seen in Equation 7, the discriminator only provides a reward for a complete sequence. We should not only care about the reward for a complete tokens but also the long-term reward for the future time-steps since the long-term reward is what we actually want. Similar to the game of Go [27] in which the agent sometimes give up an immediate interest but cares about the final victory, we apply a similar Monte Carlo roll-out strategy for an intermediate state, i.e., an unfinished sequence. We represent an N-time Monte Carlo search as in Equation 8.

$$\begin{aligned} &Y^1_{t+1:T}, ..., Y^n_{t+1:T}, ..., Y^N_{t+1:T} = MC^{G_\theta}(Y_{1:t}; N) \\ &MC = \sim Multinomial(logits) \end{aligned} \tag{8}$$

where $Y_{1:t}$ is the generated sequence tokens and $Y^n_{t+1:T}$ is the Monte Carlo sampled based on a roll-out policy, which, in our case, is set as the same as the image caption generator for convenience. In reality, we can use any policy to perform the roll-out operation. *logits* is the output of the LSTM decoder. MC is defined as a sampling procedure from a Multinomial distribution.

If there is no intermediate reward, the Monte Carlo roll-out strategy can sample the future possible tokens N times and average these rewards to achieve the goal of reward estimation, which is described in Equation 9.

$$Q^{G_\theta}_{D_\theta}(a = y_t, s = Y_{1:t-1}) = \begin{cases} \frac{1}{N}\sum_{n=1}^{N} D_\theta(Y^n_{1:T}), Y^n_{1:T} \in MC^{G_\theta}(Y_{1:t}; N), & for\ t < T \\ D_\theta(Y_{1:T}), & for\ t = T \end{cases} \tag{9}$$

The Monte Carlo roll-out strategy can be better visualized in Fig. 4.

Once the reward value from the discriminator is obtained, it is ready to update the generator. The goal is to maximize the average reward starting from the initial state as defined in Equation 10.

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} V_\theta(s_0|X_i, Y_i) \tag{10}$$

where N is the number of samples used for training. We can use the Policy Gradient theorem from [26] and write the gradient of the objective function

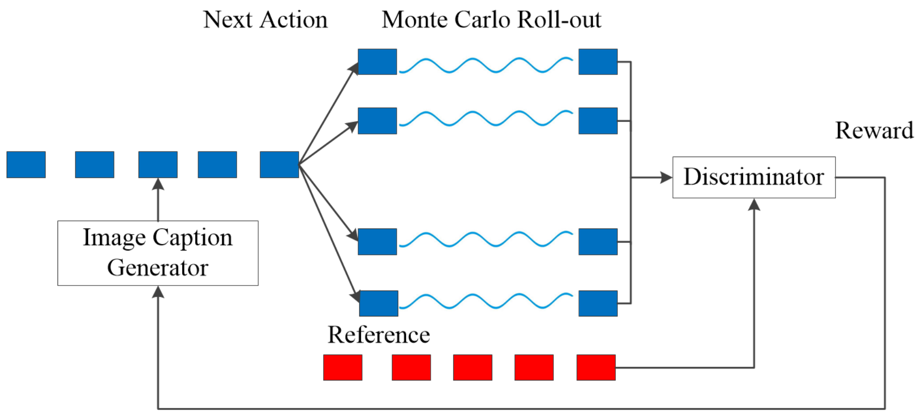


Figure 4: Monte Carlo roll-out: We use Monte Carlo sampling to sample tokens in the future time steps and average them to obtain the intermediate rewards so as to optimize the token generated at each time step.

(reward signal) as in Equation 11.

$$\bigtriangledown_\theta J(\theta) = E_{Y_{1:t-1}\sim G_\theta}[\sum_{y_t\in Y} \bigtriangledown G_\theta(y_t|Y_{1:t-1})\cdot Q^{G_\theta}_{D_\theta}(Y_{1:t-1}, y_t)] \tag{11}$$

Since the expectation can be approximated by sampling, we can now update the parameters of the image caption generator using Equation 12.

$$\theta \leftarrow \theta + \alpha_h \bigtriangledown_\theta J(\theta) \tag{12}$$

In practice, we can use advanced gradient algorithms such as RMSprop [58] and Adam [59] in training the caption generator.

The image caption generator and discriminator are adversarially trained in the framework of GAN [22]. In GAN [23], the discriminator can pass the gradient directly to the generator. Due to the discreteness of the sequence generation, we apply RL to estimate the gradient of the generator in our model.

Specifically, the training strategy is described in Algorithm 1. We initially pre-train the image caption generator using MLE. In practice, this is equivalent to the Cross Entropy loss [60]. Hence, we can set the pre-training step the same as in [18]. The trained model is used to generate some captions which are set as fake samples, which, along with the reference captions, are fed into the discriminator for training. Similarly, the discriminator is also pre-trained for certain steps. The next steps are the adversarial training steps, in which the image caption generator and discriminator are trained alternatively until convergence of the networks.

In addition to the sentence comparison scheme introduced previously, and shown in Fig. 2, we also employ a scheme to evaluate the coherence between the generated captions and the image content. Specifically, both of the global features and local object features are processed by average pooling in order to obtain fixed-size feature representation, denoted as V_i. The captions, similar to the sentence comparison scheme, are also encoded into a fixed-size vector, using a LSTM model, denoted as V_w. The two vectors V_i and V_w are then dot producted and forwarded to logistic function to obtain the reward for RL training, which can be seen in Fig. 3.

4. Experimental Validation

4.1. Dataset Introduction

We conduct our experiments using the MSCOCO dataset [61]. To be consistent with the previous researches, we use the MSCOCO 2014 released version, which includes 123,000 images. The dataset contains 82,783 images in the training set, 40,504 images in the validation set and 40,775 images in the test set. As the ground-truth for the MSCOCO test set is not available, the validation set is

Algorithm 1 Image Caption Generation by Adversarial Training and Reinforcement Learning

```
Input: Image Caption Generator $G_\theta$; Discriminator $D_\theta$.
  Pre-training $G_\theta$ using MLE by 10 epoches.
  Generating negative samples using pre-trained $G_\theta$ to train $D_\theta$.
  Pre-training $D_\theta$ by 2500 steps.
  repeat
    for update-generator for 1 step do
      Generate a sequence $Y_{1:T} = (y_1, .., y_T)$.
      for $t = 1$ to $T$ do
        Compute the intermediate reward $Q(t)$ by Monte Carlo roll-out.
      end for
      Update the parameters $\theta$ using Policy Gradient.
    end for
    for update-discriminator for 1 step or 5 steps do
      Training discriminator $D_\theta$ using reference sequence (True) and generated
      sequence (Fake) using current generator.
    end for
  until Convergence
```

Table 1: Parameter Settings of the Hierarchical Image Encoder

Global Image Decoder	Global Image Features Dimension	Faster RCNN model	Local Image Features Dimension
Residual-152	49×2048	VGG16	30×4096

Table 2: Parameter Settings of the Language Decoder

Word Embedding Dimension	$LSTM^G$ Dimension	$LSTM^L$ Dimension	Maximum Sequence Length (Training)	Maximum Sequence Length (Inference)
512	512	512	20	30

further splited into a validation subset for model selection and a test subset for local experiments. This is the “Karpathy” split [13]. It utilizes the whole 82,783 training set images for training, and selects 5,000 images for validation and 5,000 images for testing from the official validation set. The standard evaluation protocol contains BLEU [40], METEOR [41], CIDEr [46] and ROUGE-L [42].

BLEU is the most popular metric for the performance evaluation in machine translation. The metric is only based on the n-gram statistics. The BLEU-1, BLEU-2, BLEU-3 and BLEU-4 measure the performance of the 1, 2, 3, 4-gram, respectively. METEOR is based on the harmonic mean of unigram precision and recall, and seeks correlation at the corpus level. CIDEr can be used to evaluate the generated sentences with human consensus. ROUGE-L measures the common maximum-length subsequence for the target sentence and the generated sentence.

4.2. Implementation Details

The whole pipeline of the algorithm and implementation procedure are pre-

Table 3: Parameter Settings of Training

MLE Pre-training	Batch Size	Learning Rate of MLE	Optimizer	Discriminator Pre-training	Learning Rate of Policy Gradient
10 epochs	32	0.001	Adam	2500 iterations	0.0001

Algorithm 2 The whole pipeline of the proposed method

```
Pre-train the Faster R-CNN on the MSCOCO dataset.
Extract features via Residual-152 and the pre-trained Faster R-CNN.
Language pre-processing.
MLE pre-training.
Perform Algorithm 1.
```

sented in Algorithm 2. For all the images in the COCO dataset, we obtain global convolutional features (from the layer "res5c") using a pre-trained Residual-152 network [3] on the platform of Caffe [62], with a dimensionality of 49×2048. We also retrieve local object features using a Faster RCNN [10] object detection network pre-trained on the MSCOCO dataset. Specifically, we obtain the top K detected object features from the layer of "FC6" layer of the VGG16 model [2] used in Faster RCNN, with dimensionality of $K \times 4096$. We build the hierarchical attention mechanism and policy gradient optimization on the TensorFlow platform [63].

4.2.1. Training the Faster RCNN on the MSCOCO dataset

In order to obtain better local object features, we train the Faster RCNN model on MSCOCO object detection dataset. The model is first pre-trained on the ImageNet object detection dataset [55]. The MSCOCO object detection dataset shares the same images with the image caption task. Consequently, we keep the same splits with the image caption dataset for training. The training process on the MSCOCO dataset is almost the same with the pre-training on ImageNet. The initial learning rate is set to 0.001. The momentum of the stochastic gradient descent is set to 0.9 and the weight decay is set to 0.0005.

4.2.2. Language Pre-processing

To pre-process the language, the special symbols such as '.', ',', '(', ')' and '-' are replaced with blank spaces whilst '&' is replaced with 'and'. Since we set the maximum length of the descriptions as 20 words, we delete the caption references from the original dataset which are longer than 20. For the vocabulary establishment, following the open-source code of [13], we include words that occurs more than 5 times in the vocabulary. We map the symbol 'NULL' to 0, 'START' to 1 and 'END' to 2.

4.2.3. Training Details of the Model

Following the open-source code of [13], at training time, we set the maximum length of the input sequence to 20 words. During the testing time, alternatively, we set maximum length of a generated symbols as 30 words. During the training of the proposed model, we add a trainable word embedding layer from Google's TensorFlow platform [63]. All the experiments are conducted on a server embedded with NVIDIA TITAN X GPU and installed with the Ubuntu 14.04 operating system. We summarise the parameters of the model and training in Table 1, Table 2 and Table 3. Our code is publicly available at [1].

4.3. Results

4.3.1. Quantitative Evaluation

In this section, a comprehensive quantitative evaluation is conducted using different experimental settings on the MSCOCO dataset.

[1] https://github.com/Shiyang-Yan/image-captioning-with-hierarchical-attention-and-policy-gradient-optimisation

Comparison between the global attention, the local attention and the hierarchical attention model. We first obtain the results using only the global attention model, which is similar to the soft attention model in [18]. Since we use advanced CNN features from the Residual-152 model, the results of BLEU, METEOR, CIDEr and ROUGE-L are all satisfactory, and are listed in Table 4. Then only the local attention model using the detected object features from a Faster RCNN detector is tested, with results which are much lower than those for the global attention model as listed in Table 4. One of the possible reasons is that the Faster RCNN only uses the VGG16 model, which is not as powerful as the Residual-152 network. Another reason is that the local object features, despite the capability to provide complementary information to the global attention model, can sometimes miss many important features. Finally, we test our proposed hierarchical attention model under MLE training, which utilizes both of the global and local attention for image captioning. The results improve the baseline significantly, which can be seen in Table 4. Specifically, all of the seven evaluation metrics are improved using our hierarchical attention model.

The determination of the number of top detected objects. To determine the best number k for the top detected objects in the local attention model, we perform an ablation study. We extract the 10, 20 and 30 top detected object features and test them using the hierarchical attention model. The results can be seen in Table 5. With the increase of the number k from 10 to 30, the performance increases accordingly. Although the maximum length of our generated sentences is set as 30, not every word represents an object. Also, intuitively, there are a maximum 30 objects within an image. Hence, in the following experiments, we use the 30 top detected object features for the local attention model.

Table 4: Comparison of image captioning using different attention mechanism results on the MSCOCO dataset

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
Soft Attention [18]	70.7	49.2	34.4	24.3	23.90	-	-
Global Attention	70.121	50.304	35.434	25.111	23.658	84.701	54.308
Local Attention	64.059	42.359	28.089	19.033	20.203	56.898	49.861
Hierarchical Attention	**72.611**	**52.769**	**37.802**	**27.243**	**24.731**	**88.140**	**56.048**

Table 5: Comparison of image captioning results on the MSCOCO dataset with different numbers of objects

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
Hierarchical Attention with 10 Objects for Local Attention	70.601	50.423	36.643	25.389	24.633	87.316	55.241
Hierarchical Attention with 20 Objects for Local Attention	72.159	52.498	37.552	26.918	24.725	**88.639**	55.825
Hierarchical Attention with 30 Objects for Local Attention	**72.611**	**52.769**	**37.802**	**27.243**	**24.731**	88.140	**56.048**

Table 6: Comparison of image captioning results on the MSCOCO dataset with different settings for policy gradient (PG) optimization

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
MLE training only	**72.611**	52.769	37.802	27.243	24.731	88.140	56.048
PG with 2500 steps for pre-training D followed by 1 D and 1 G step	72.450	**52.845**	**38.141**	27.551	24.543	87.416	**55.876**
PG with 2500 steps for pre-training D followed by 5 D and 1 G step	72.104	52.739	38.122	**27.602**	**24.928**	**89.072**	56.063

The performance of Policy Gradient with reward only from language comparison. Next we start the reinforcement learning steps. We first train the discriminator which only compares the similarity between the reference sentence and the generated sentence. Specifically, we follow the model defined in Fig. 2. The discriminator is first trained in 2500 steps, which we find sufficient for the discriminator to converge. The loss curve of the image caption generator is shown in Fig. 5. After 2500 steps pre-training the discriminator, the loss of the image caption generator starts to decline, which validates that the policy gradient starts to work. Then we further train the generator and discriminator

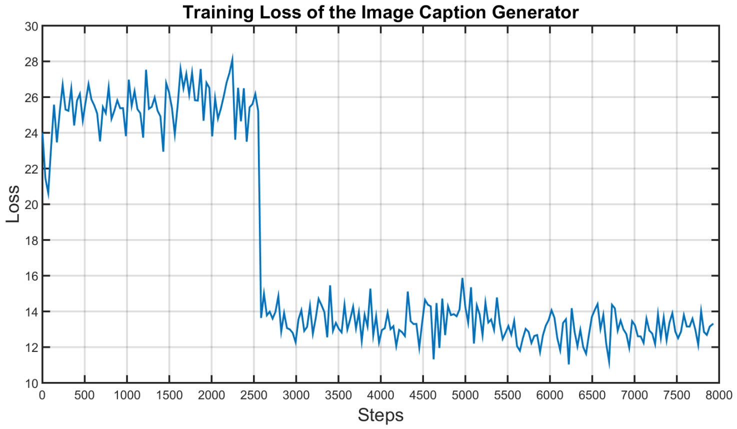


Figure 5: The loss curve of the image caption generator during reinforcement learning steps: before 2500 iterations, we pre-train the discriminator. Starting from the 2500 iterations, we start the adversarial training of the generator and discriminator. The loss value starts to decrease starting from 2500 iterations as the parameters of the generator begins to be updated.

adversarially for another 1 epoch, and report the results in Table 6. We also experimented with two different settings in the adversarial training steps. The first setting is to train 1 step for the discriminator, followed by another step for the generator. Another setting is to train the discriminator for 5 steps, followed by 1 step training for the generator. We find the final results of the two setting are similar, which all slightly improve the MLE training baseline. The reason for the improvement is because the reinforcement learning solves the exposure bias problem during MLE training. However, this scheme lacks the measurement of the similarity between the generated descriptions and the image contents, which prevents the image caption generator from generating more naturalistic and diverse descriptions.

The performance of Policy Gradient with reward from the measurement of coherence between language and image content. To train the image caption generator

Table 7: Comparison of image captioning results on the MSCOCO dataset for policy gradient (PG) optimization with discriminator for evaluation of the coherence between language and image content.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
MLE training only	72.611	52.769	37.802	27.243	24.731	88.140	56.048
Global Attention	70.121	50.304	35.434	25.111	23.658	84.701	54.308
PG with similarity of global features (1 D and 1 G step)	72.250	52.290	37.099	26.331	23.815	84.516	55.238
PG with similarity of global features (5 D and 1 G step)	72.234	52.120	36.887	26.065	23.957	84.224	55.244
PG with similarity of global-local features (1 D and 1 G step)	**73.036**	**53.688**	**39.069**	**28.551**	**25.324**	**92.449**	**56.539**

to generate more naturalistic and diverse descriptions, we further test the model defined in Fig. 2. First we only extract the global features and perform average pooling, resulting with a feature dimension of 2048. We then use the dot product to measure these image features and language embedding features by a discriminator, which can be considered as the reward within the reinforcement learning framework. The experimental results from this model can be seen in Table 7.

However, the results from all of the seven metrics are even lower than the MLE training baseline. One possible reason, is the measurement of discriminator which only uses the global features, which is not consistent with the hierarchical attention model in the generator side. As can be seen from the Table 7, the results from this model are similar to that of global attention model, since the reward signal from the discriminator tends to force the generator to produce sentences that only matches the global features.

We further build a model exactly like in the one defined in Fig. 3. This model includes both of the global image features and the local object features, and thus guarantees that the discriminator and the generator are utilizing the same information source. The final results can be seen in Table 7, which outperform

Table 8: Comparison of image captioning results on the MSCOCO dataset with previous methods.

Methods	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr	ROUGE-L
Google NIC [14]	66.6	46.1	32.9	24.6	-	-	-
m-RNN [28]	67	49	35	25	-	-	-
BRNN [13]	64.2	45.1	30.4	20.3	-	-	-
MSR/CMU [64]	-	-	-	19.0	20.4	-	-
Spatial Attention [18]	71.8	50.4	35.7	25.0	23.0	-	-
gLSTM [29]	67.0	49.1	35.8	26.4	22.7	81.3	-
GLA [19]	56.8	37.2	23.2	14.6	16.6	36.2	41.9
MIXER [43]	-	-	-	29.0	-	-	-
Conv Image Caption [65]	71.1	53.8	39.4	28.7	24.4	91.2	52.2
SCA-CNN-ResNet [37]	71.9	**54.8**	**41.1**	**31.1**	25.0	-	-
Semantic Attention [35]	70.9	53.7	40.2	30.4	24.3	-	-
DCC [66]	64.4	-	-	-	21.0	-	-
RL with G-GAN [21]	-	-	30.5	29.7	22.4	79.5	47.5
RL with Embedding Reward [67]	71.3	53.9	40.3	30.4	25.1	93.7	52.5
Self-Critical (CIDEr) [68]	-	-	-	31.9	**25.5**	**106.3**	54.3
Ours	**73.036**	53.688	39.069	28.551	25.324	92.449	**56.539**

all of other experimental settings.

To prove the effectiveness of the proposed method, we compare our final results on the "Karpathy" test split with previously published results, which is shown in Table 8. We list most of the published results on the "Karpathy" split, which are grouped into three categories. The first category corresponds to various methods without external information and reinforcement learning. The best of them (SCA-CNN-ResNet) is the spatial and channel-wise attention model [37] in which both the spatial and channel-wise attention mechanisms are utilized for image captioning. The methods in the second group use extra information during the training of the model. For instance, Semantic Attention [35] utilizes rich extra data from social media to train the visual attribute predictor.

Deep Compositional Captioning (DCC) [66] generates extra data to prove its unique transfer capability. The third group corresponds to the reinforcement learning technique. RL with G-GAN [21] applies conditional GAN and policy gradient to generate image descriptions. Although their results on the evaluation metrics are not improved, they prove that the generated captions are more diverse and naturalistic. Embedding Reward [67] applies a policy network to generate captions and a value network to evaluate the reward. Additionally, they also apply advanced inference method called lookahead inference and beam search during testing. They achieve the current state-of-the-art results on the “Karpathy” split. Although we do not use any external knowledge and any advanced inference technique (including beam search, we use greedy search in all of our experiments), we achieve similar results to the current state-of-the-art methods (Embedding Reward [67], SCA-CNN-ResNet [37] and self-critical [68]), with state-of-the-art results on two important metrics: BLEU-1 and ROUGE-L and lead other methods significantly.

4.3.2. Qualitative Evaluation

In addition to the quantitative evaluation using the standard metrics, we qualitatively evaluate the proposed model by visualization. Firstly, we plot some global attention maps corresponding to each generated words as shown in Fig. 6. It is obvious in the figure that the attentive regions normally correspond with the semantic meaning of the generated word in each time step. Then we choose some examples to visualize the local attention weights on the detected objects, which are shown in Fig. 7. We only retrieve the top 10 detected objects and corresponding attentive weights obtained from the local attention mechanism because of limited space in the figure. The detector can detect

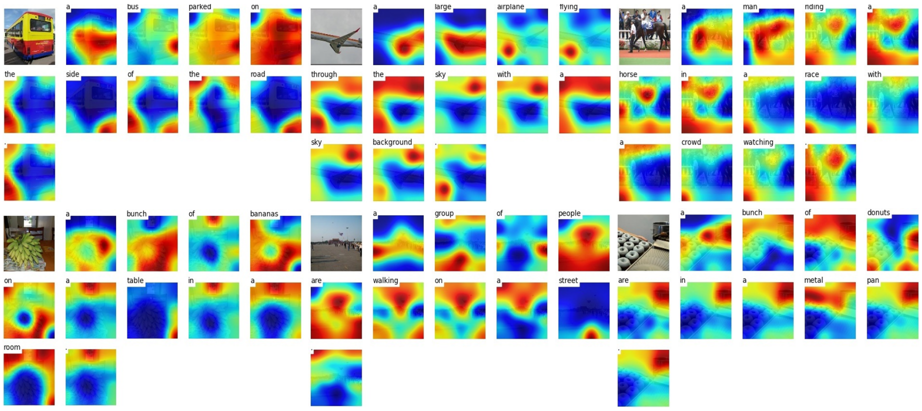


Figure 6: Visualization of the global attention maps and generated captions. The red color indicates the importance of each region of the image.

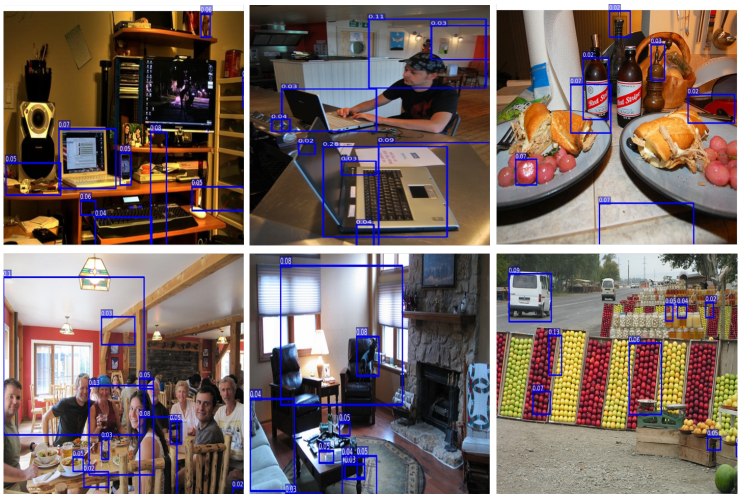
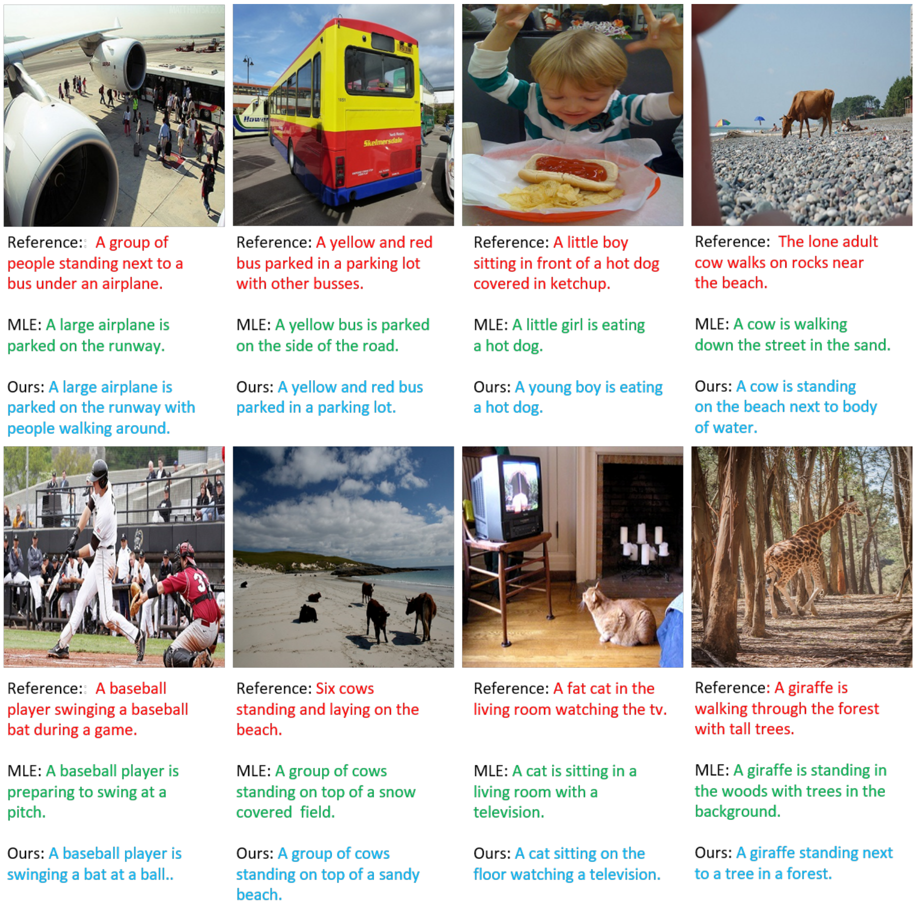


Figure 7: Visualization of the attentive weights on the top 10 detected objects, the blue boxes indicate the detected objects whilst the labels show the attentive weights of the local attention model.



Reference: A group of people standing next to a bus under an airplane.

MLE: A large airplane is parked on the runway.

Ours: A large airplane is parked on the runway with people walking around.

Reference: A yellow and red bus parked in a parking lot with other busses.

MLE: A yellow bus is parked on the side of the road.

Ours: A yellow and red bus parked in a parking lot.

Reference: A little boy sitting in front of a hot dog covered in ketchup.

MLE: A little girl is eating a hot dog.

Ours: A young boy is eating a hot dog.

Reference: The lone adult cow walks on rocks near the beach.

MLE: A cow is walking down the street in the sand.

Ours: A cow is standing on the beach next to body of water.

Reference: A baseball player swinging a baseball bat during a game.

MLE: A baseball player is preparing to swing at a pitch.

Ours: A baseball player is swinging a bat at a ball..

Reference: Six cows standing and laying on the beach.

MLE: A group of cows standing on top of a snow covered field.

Ours: A group of cows standing on top of a sandy beach.

Reference: A fat cat in the living room watching the tv.

MLE: A cat is sitting in a living room with a television.

Ours: A cat sitting on the floor watching a television.

Reference: A giraffe is walking through the forest with tall trees.

MLE: A giraffe is standing in the woods with trees in the background.

Ours: A giraffe standing next to a tree in a forest.

Figure 8: The visualisation of some captioning results.

some fine-grained objects, which provide complementary information for the global attention mechanism. At last, we show some of the generated sentences using different methods. Specifically, we show the ground-truth sentences, descriptions generated by the MLE training-based model and by the proposed model as shown in Fig. 8. The text in blue are the sentences generated by the proposed model, which are more accurate and naturalist than the MLE-based model, which are shown in green. Specially, the proposed model show superior performance in finding the fine-grained properties of the image since the RL model automatically measure the coherence of the sentences and the image content.

5. Conclusion

This paper targets the image captioning task, which is a fundamental problem in artificial intelligence. Based on the recent successes of deep learning, especially the CNN feature representation and the LSTM with attention model, the paper proposes the use of a hierarchical attention mechanism, considering not only the global image features but also detected object features, with improved results. A significant improvement over the current RNN-based MLE training has also been demonstrated. Specifically, a GAN framework with RL optimization for the image captioning task is proposed to generate more accurate and high-quality captions. The discriminator is to evaluate the coherence and consistency between the generated sentences and image content, thus providing the rewards for optimization. The whole model follows a three-step training strategy. Experiments analysis confirms the merits of the framework and key contributors the improved performance. Comparable results with current state-of-the-art methods are achieved using only greedy inference, which proves the effectiveness of the training procedure.

References

References

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.

[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: ICLR, 2015.

[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016.

[4] S. Tang, Y. T. Zheng, Y. Wang, T. S. Chua, Sparse ensemble learning for concept detection, IEEE Transactions on Multimedia 14 (1) (2012) 43–54.

[5] C. Kang, S. Xiang, S. Liao, C. Xu, C. Pan, Learning consistent feature representation for cross-modal multimedia retrieval, IEEE Transactions on Multimedia 17 (3) (2015) 370–381.

[6] S. Bu, Z. Liu, J. Han, J. Wu, R. Ji, Learning high-level feature by deep belief networks for 3-d model retrieval and recognition, IEEE Transactions on Multimedia 16 (8) (2014) 2154–2167.

[7] P. Liu, J. M. Guo, C. Y. Wu, D. Cai, Fusion of deep learning and compressed domain features for content-based image retrieval, IEEE Transactions on Image Processing 26 (12) (2017) 5706–5717.

[8] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: CVPR, 2014.

[9] R. Girshick, Fast r-cnn, in: ICCV, 2015.

[10] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: NIPS, 2015.

[11] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, T. L. Berg, Babytalk: Understanding and generating simple image descriptions, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (12) (2013) 2891–2903.

[12] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al., From captions to visual concepts and back, in: CVPR, 2015.

[13] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: CVPR, 2015.

[14] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: CVPR, 2015.

[15] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014.

[16] K. Cho, A. Courville, Y. Bengio, Describing multimedia content using attention-based encoder-decoder networks, IEEE Transactions on Multimedia 17 (11) (2015) 1875–1886.

[17] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: ICLR, 2015.

[18] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: ICML, 2015.

[19] L. Li, S. Tang, Y. Zhang, L. Deng, Q. Tian, Gla: Global-local attention for image description, IEEE Transactions on Multimedia PP (99) (2017) 1–1.

[20] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer, Scheduled sampling for sequence prediction with recurrent neural networks, in: NIPS, 2015.

[21] B. Dai, S. Fidler, R. Urtasun, D. Lin, Towards diverse and natural image descriptions via a conditional gan, in: CVPR, 2017.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: NIPS, 2014.

[23] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.

[24] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, Vol. 1, MIT press Cambridge, 1998.

[25] L. Yu, W. Zhang, J. Wang, Y. Yu, Seqgan: Sequence generative adversarial nets with policy gradient., in: AAAI, 2017.

[26] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: NIPS, 2000.

[27] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al.,

Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.

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

[29] X. Jia, E. Gavves, B. Fernando, T. Tuytelaars, Guiding long-short term memory for image caption generation, arXiv preprint arXiv:1509.04942.

[30] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: CVPR, 2015.

[31] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078.

[32] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: NIPS, 2014.

[33] J. Ba, V. Mnih, K. Kavukcuoglu, Multiple object recognition with visual attention, arXiv preprint arXiv:1412.7755.

[34] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, in: ICCV, 2015.

[35] Q. You, H. Jin, Z. Wang, C. Fang, J. Luo, Image captioning with semantic attention, in: CVPR, 2016.

[36] T. Yao, Y. Pan, Y. Li, Z. Qiu, T. Mei, Boosting image captioning with attributes, in: ICCV, 2017.

[37] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, T.-S. Chua, Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning, in: CVPR, 2017.

[38] A. Goyal, N. R. Ke, A. Lamb, R. D. Hjelm, C. Pal, J. Pineau, Y. Bengio, Actual: Actor-critic under adversarial learning, arXiv preprint arXiv:1711.04755.

[39] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160.

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

[41] A. Lavie, A. Agarwal, Meteor: An automatic metric for mt evaluation with improved correlation with human judgments, in: EMNLP Workshop on Statistical Machine Translation, 2005.

[42] C.-Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: NAACL, 2003.

[43] M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence level training with recurrent neural networks, in: ICLR, 2016.

[44] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, K. Murphy, Improved image captioning via policy gradient optimization of spider, in: ICCV, 2017.

[45] P. Anderson, B. Fernando, M. Johnson, S. Gould, Spice: Semantic propositional image caption evaluation, in: ECCV, 2016.

[46] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: Consensus-based image description evaluation, in: CVPR, 2015.

[47] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, Y. Bengio, An actor-critic algorithm for sequence prediction, in: ICLR, 2017.

[48] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: NIPS, 2016.

[49] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv preprint arXiv:1701.07875.

[50] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint arXiv:1611.01144.

[51] C. J. Maddison, A. Mnih, Y. W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, arXiv preprint arXiv:1611.00712.

[52] R. Shetty, M. Rohrbach, L. A. Hendricks, M. Fritz, B. Schiele, Speaking the same language: Matching machine to human captions by adversarial training, in: ICCV, 2017.

[53] M. J. Kusner, J. M. Hernández-Lobato, Gans for sequences of discrete elements with the gumbel-softmax distribution, arXiv preprint arXiv:1611.04051.

[54] L. Wu, Y. Xia, L. Zhao, F. Tian, T. Qin, J. Lai, T.-Y. Liu, Adversarial neural machine translation, arXiv preprint arXiv:1704.06933.

[55] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual

recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.

[56] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[57] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning 8 (3-4) (1992) 229–256.

[58] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning 4 (2) (2012) 26–31.

[59] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015.

[60] P.-T. De Boer, D. P. Kroese, S. Mannor, R. Y. Rubinstein, A tutorial on the cross-entropy method, Annals of operations research 134 (1) (2005) 19–67.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: ECCV, 2014.

[62] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACMMM, 2014.

[63] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale

machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467.

[64] X. Chen, C. L. Zitnick, Mind's eye: A recurrent visual representation for image caption generation, in: CVPR, 2015.

[65] J. Aneja, A. Deshpande, A. G. Schwing, Convolutional image captioning, in: CVPR, 2018.

[66] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, T. Darrell, Deep compositional captioning: Describing novel object categories without paired training data, in: CVPR, 2016.

[67] Z. Ren, X. Wang, N. Zhang, X. Lv, L.-J. Li, Deep reinforcement learning-based image captioning with embedding reward, in: CVPR, 2017.

[68] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, Self-critical sequence training for image captioning, in: CVPR, 2017.