

Improving Image Captioning with Conditional Generative Adversarial Nets

Chen Chen, Shuai Mu, Wanpeng Xiao, Zexiong Ye, Liesi Wu, Qi Ju

Tencent AI Lab, Shenzhen, China 518000

{beckhamchen, harrymou, wanpengxiao, joeyye, henrylwu, damonju}@tencent.com

Abstract

In this paper, we propose a novel conditional-generative-adversarial-nets-based image captioning framework as an extension of traditional reinforcement-learning (RL)-based encoder-decoder architecture. To deal with the inconsistent evaluation problem among different objective language metrics, we are motivated to design some “discriminator” networks to automatically and progressively determine whether generated caption is human described or machine generated. Two kinds of discriminator architectures (CNN and RNN-based structures) are introduced since each has its own advantages. The proposed algorithm is generic so that it can enhance any existing RL-based image captioning framework and we show that the conventional RL training method is just a special case of our approach. Empirically, we show consistent improvements over all language evaluation metrics for different state-of-the-art image captioning models. In addition, the well-trained discriminators can also be viewed as objective image captioning evaluators.

1 Introduction

Generating natural language descriptions of given images, known as image captioning, has attracted great academic and industrial interest since it can be widely used in image-text cross-searching, early childhood education and eye-handicapped people assistance. Compared with other computer vision tasks, *e.g.* image classification, object detection and semantic segmentation, image captioning is a more challenging and comprehensive task — as it requires a fine-grain understanding of image objects as well as their attributes and relationships. Therefore, image captioning can be viewed as an interdisciplinary research domain of computer vision and natural language processing (NLP).

Inspired by the successful application of the encoder-decoder paradigm in neural machine translation (NTM) with RNNs, some pioneer works (Mao et al. 2014; Vinyals et al. 2015) creatively proposed to replace an RNN with a CNN to encode image features. Since then, the CNN-RNN structure has become a standard configuration of image captioning algorithms. Most prior works used maximum likelihood estimation (MLE) for training. However, as pointed out in (Ranzato et al. 2015), this approach suffers from error accumula-

tion, namely a bias exposure problem (Bengio et al. 2015), which creates a mismatch between training and testing — since at test-time, the model uses the previously generated words from the model distribution to predict the next word while directly uses ground-truth during training.

In order to address the exposure bias issue, many works incorporated reinforcement learning (RL) into the training stage to directly optimize language metrics such as BLEU (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005), ROUGE (Lin 2004), CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) and SPICE (Anderson et al. 2016). A policy-gradient (Sutton et al. 2000)-based RL method was first employed in image captioning, such as REINFORCE with a baseline (Rennie et al. 2017) and Monte-Carlo rollouts (Liu et al. 2017). Then, the so-called actor-critic RL algorithm was applied in (Ren et al. 2017; Zhang et al. 2017), which trains a second “critic” network to estimate the expected future reward during the intermediate state when generating each word given the policy of an actor network.

Recently, the most popular generative model—generative adversarial nets (GANs) (Goodfellow et al. 2014)—has achieved great success in computer vision tasks. But unlike the deterministic continuous mapping from random vectors to the image domain, many NLP tasks are discrete domain generation issues. Inspired by the idea of reinforcement learning, a SeqGAN algorithm (Yu et al. 2017) was proposed to bypass the generator differentiation problem by directly performing the gradient policy update and successfully applied the result to text and music generation. In (Dai et al. 2017) and (Shetty et al. 2017), the authors conducted the first study that explores the use of conditional GANs (Mirza and Osindero 2014) or adversarial training in combination with an approximate Gumbel sampler in generating image descriptions. However, what these papers concern about most is naturalness and diversity of descriptions while sacrificing the fidelity, which results in much lower language metrics scores compared with other image captioning algorithms.

In order to achieve high evaluation scores over objective automatic metrics, many prior works chose to directly optimize one metric (BLEU, CIDEr, SPICE, *etc.*) or combination of them. However, optimizing one metric cannot ensure consistent improvements over all metrics. Therefore, in or-

der to simultaneously improve all language evaluation metrics and generate more human-like descriptions, we are motivated to design a “discriminator” network to judge whether the input sentence is human described or machine generated based on the idea of conditional GANs (Mirza and Osindero 2014). In this paper, we propose to employ an adversarial training method to alternatively improve the generator (caption generation network) and the discriminator (sentence evaluation network).

The main contributions of our proposed image captioning framework are listed as follows:

- Our proposed algorithm is generic so that it can enhance any existing RL-based image captioning framework and we show consistent improvements over all evaluation metrics by experiments.
- The well-trained discriminators can also be viewed as evaluators for image captioning learned by neural network instead of the traditional handcrafted evaluation metrics.
- Based on the up-down (Anderson et al. 2018) image captioning model, the improved results trained by our approach are uploaded to MSCOCO online test server and achieve the state-of-the-art performance.

2 Image Captioning Via Reinforcement Learning

As described in the introduction, the traditional RNN model training method is MLE. That is, the model parameters θ of the caption generator are trained to maximize

$$\begin{aligned} J_G(\theta) &= \frac{1}{N} \sum_{j=1}^N \log G_\theta(\mathbf{x}^j | \mathbf{I}^j) \\ &= \frac{1}{N} \sum_{j=1}^N \sum_{t=1}^{T_j} \log G_\theta(\mathbf{x}_t^j | \mathbf{x}_{1:t-1}^j, \mathbf{I}^j), \end{aligned} \quad (1)$$

where \mathbf{I}^j is the j -th image, $\mathbf{x}^j = (\mathbf{x}_1^j, \mathbf{x}_2^j, \dots, \mathbf{x}_{T_j}^j)$ is the ground truth caption of the j -th image, T_j is the caption length of the j -th image, N is the total number of training examples, and $G_\theta(\cdot)$ is the probability of generated words given an image or previous words, parameterized by θ (or we can directly call G_θ the generator).

By using the RL terminologies as described in (Sutton and Barto 1998), in an encoder-decoder image captioning paradigm, the decoder can be viewed as an “agent” that interacts with an external “environment” (input words and image features extracted by the encoder). The “policy” is the caption generator G_θ , that results in an “action” that is the prediction of the next word. After taking each “action”, the “agent” updates its internal “state” (weights of decoder, attention models, *etc.*). Upon generating the end-of-sequence (EOS) token, the “agent” returns a “reward”, denoted by r , that is, for instance, a language evaluation metric score (CIDEr, BLEU, SPICE, *etc.*) calculated by comparing generated sentences and the corresponding ground truth. So, the goal of RL training is to maximize the final expected reward of the generator:

$$L_G(\theta) = \mathbb{E}_{\mathbf{x}^s \sim G_\theta} [r(\mathbf{x}^s)], \quad (2)$$

where $\mathbf{x}^s = (\mathbf{x}_1^s, \mathbf{x}_2^s, \dots, \mathbf{x}_T^s)$ is a sample sequence from generator G_θ . In practice, the expected value is approximated by a Monte-Carlo sample (Sutton and Barto 1998):

$$L_G(\theta) \approx r(\mathbf{x}^s), \quad \mathbf{x}^s \sim G_\theta. \quad (3)$$

Typically, the gradient $\nabla_\theta L_G(\theta)$ can be calculated by a policy gradient approach such as REINFORCE (Williams 1992) algorithm with a baseline function b to effectively reduce the variance of the gradient estimate:

$$\nabla_\theta L_G(\theta) \approx \sum_{t=1}^{T_s} (r(\mathbf{x}_{1:t}^s) - b) \nabla_\theta \log G_\theta(\mathbf{x}_t^s | \mathbf{x}_{1:t-1}^s). \quad (4)$$

The baseline b can be an arbitrary function, as long as it does not depend on the “action” \mathbf{x}^s . A self-critical sequence training (SCST) (Rennie et al. 2017) method employs the reward $r(\mathbf{x}^g)$ obtained by the current model under the greedy decoding algorithm used at test time as the baseline function. Then, the gradient function can be written as

$$\nabla_\theta L_G(\theta) = \sum_{t=1}^{T_s} (r(\mathbf{x}^s) - r(\mathbf{x}^g)) \nabla_\theta \log G_\theta(\mathbf{x}_t^s | \mathbf{x}_{1:t-1}^s). \quad (5)$$

Since the reward of the sample sequence $\mathbf{x}_{1:T_s}^s$ and the greedy-decoded sequence $\mathbf{x}_{1:T}^g$ are the same during each time step t , we omit the subscripts of \mathbf{x}^s and \mathbf{x}^g in Eq. (5).

3 Proposed Conditional Generative Adversarial Training Method

3.1 Overall Framework

As described in the introduction, the most commonly used image captioning model is the so-called encoder-decoder framework. Typically, a CNN is employed as the encoder and an RNN is utilized as the decoder. Together with the attention mechanism and reinforcement learning method, a general caption generation framework is shown in the left part of Fig. 1, denoted as the generator. Inspired by the well-known generative adversarial nets (Goodfellow et al. 2014), a discriminator can be embedded into the image captioning framework to further improve the performance of the generator, which is the initial design spirit of our proposed adversarial training method. Notice that our proposed framework is generic and can enhance any existing RL-based encoder-decoder model so the attention mechanism may be unnecessary. In addition, the CNN encoder can be pre-trained on other datasets so that its weights can be fixed during decoder training (see the dashed lines and squares in Fig. 1).

As shown in Fig. 1, after generating a sentence $\tilde{\mathbf{x}}_{1:T}$, two modules will compute two scores based on different criteria: a discriminator D_ϕ with parameters ϕ will produce a probability $p \in [0, 1]$ that indicates a given sentence is human generated rather than machine generated, and a language evaluator module will calculate an objective score s based on some predefined evaluation metrics Q such as BLEU, CIDEr and SPICE. Notice that the discriminator will be improved together with the generator alternatively during training while the language evaluator module is a predefined

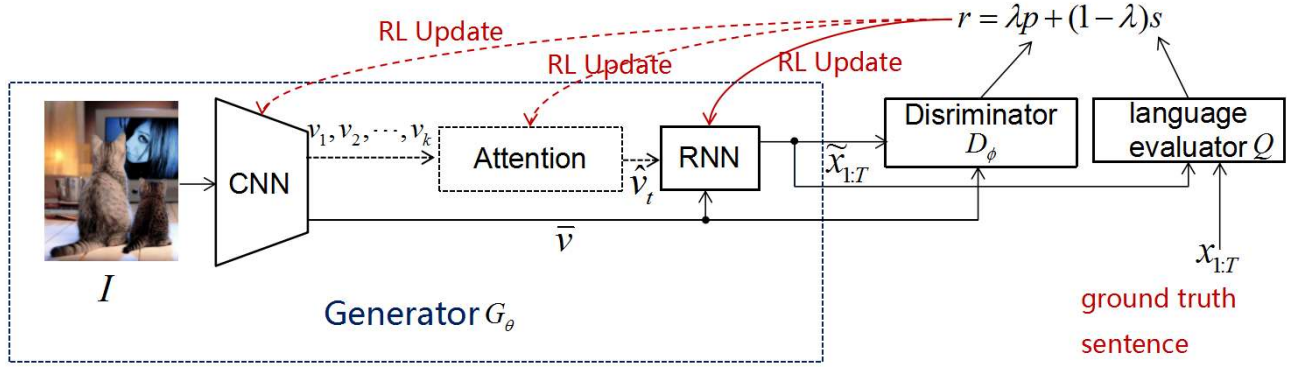


Figure 1: The overall framework of our proposed generative adversarial training method. The generator contains a CNN encoder, a RNN decoder and an unnecessary attention mechanism. The discriminator cooperated with the language evaluator provides feedback reward to update generator parameters by the reinforcement learning method (notice that CNN weights can be pre-trained and fixed).

Table 1: Kernel sizes and number of CNN discriminators

(window size, kernel numbers)
(1,100), (2,200), (3,200), (4,200), (5,200), (6,100), (7,100), (8,100), (9,100), (10,100), (15,160), (16,160)

function and is strictly fixed during training. Therefore, the two modules are cooperated together to obtain a criterion that balances the fidelity (achieves a high score under objective evaluation metrics) and naturalness (a human-like language style).

Finally, the obtained reward for reinforcement learning after generating a full sentence \tilde{x} given image I and ground-truth sentence x is calculated as

$$r(\tilde{x}|I, x) = \lambda \cdot p + (1-\lambda) \cdot s = \lambda \cdot D_\phi(\tilde{x}|I) + (1-\lambda) \cdot Q(\tilde{x}|x), \quad (6)$$

where λ is a hyper-parameter between 0 and 1.

In the following subsections, we will introduce two kinds of discriminators—CNN-based and RNN-based structures—in detail.

3.2 CNN-based Discriminator Model

Recently, many CNN-based algorithms have shown great effectiveness in complicated NLP tasks such as sentence classification (Kim 2014) and text classification (Zhang and LeCun 2015). Therefore, in this subsection, we first present the conditional CNN as our discriminator for real or fake sentence classification.

Following the discriminator design of SeqGAN (Yu et al. 2017), as illustrated in Fig. 2(a), first, we should build a feature map that consists of image features and sentence features, such as

$$\varepsilon = \bar{v} \oplus E \cdot x_1 \oplus E \cdot x_2 \oplus \dots \oplus E \cdot x_T, \quad (7)$$

where $\bar{v} = \text{CNN}(I)$ is the d -dimensional image feature pre-processed by a CNN for input image I , $E \in \mathbb{R}^{d \times U}$ is the

embedding matrix to map a U -dimensional one-hot word vector x_i ($i = \{1, 2, \dots, T\}$) into a d -dimensional token embedding, and \oplus is the horizontal concatenation operation to build the matrix $\varepsilon \in \mathbb{R}^{d \times (T+1)}$. In order to extract

Algorithm 1 Image Captioning Via Generative Adversarial Training Method

Require: caption generator G_θ ; discriminator D_ϕ ; language evaluator Q , e.g. CIDEr-D; training set $\mathbb{S}_r = \{(I, x_{1:T})\}$ and $\mathbb{S}_w = \{(I, \tilde{x}_{1:T})\}$.

Ensure: optimal parameters θ, ϕ .

- 1: Initial G_θ and D_ϕ randomly.
- 2: Pre-train G_θ on \mathbb{S}_r by MLE.
- 3: Generate some fake samples based on G_θ to form $\mathbb{S}_f = \{(I, \tilde{x}_{1:T})\}$.
- 4: Pre-train D_ϕ on $\mathbb{S}_r \cup \mathbb{S}_f \cup \mathbb{S}_w$ by Eq. (12).
- 5: **repeat**
- 6: **for** g-steps=1 : g **do**
- 7: Generate a mini-batch of image-sentence pairs $\{(I, \tilde{x}_{1:T})\}$ by G_θ .
- 8: Calculate p based on Eqs. (7)-(9) or Eqs. (10)-(11).
- 9: Calculate s based on Q .
- 10: Calculate reward r according to Eq. (6).
- 11: Update generator parameters θ by SCST method via Eq. (5).
- 12: **end for**
- 13: **for** d-steps=1 : d **do**
- 14: Generate negative image-sentence pairs $\{(I, \tilde{x}_{1:T})\}$ by G_θ , together with negative samples $\{(I, \tilde{x}_{1:T})\} \subseteq \mathbb{S}_w$ and positive samples $\{(I, x_{1:T})\} \subseteq \mathbb{S}_r$.
- 15: Update discriminator parameters ϕ via Eq. (12).
- 16: **end for**
- 17: **until** generator and discriminator converge

different features, we apply m group convolution kernels with different window sizes $d \times l_i$ ($i = \{1, 2, \dots, m\}$), each of which consists of n_i ($i = \{1, 2, \dots, m\}$) ker-

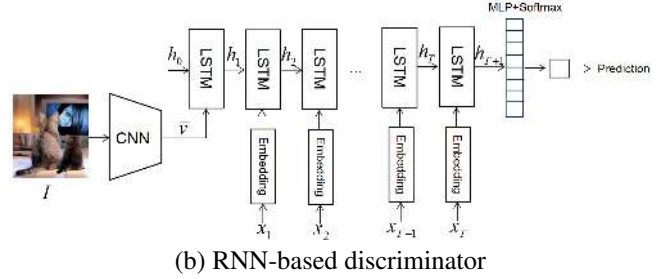
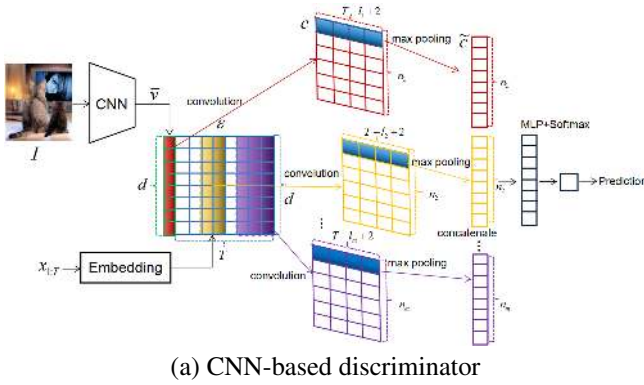


Figure 2: CNN and RNN-based discriminator architectures. Best viewed in colour.

Table 2: λ selection

fixed parameters: $g=1$; $d=1$; Metric=CIDEr-D					
λ	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
0	0.363	0.277	0.569	1.201	0.214
0.3	0.383	0.286	0.586	1.232	0.221
0.5	0.368	0.285	0.581	1.215	0.220
0.7	0.353	0.280	0.565	1.169	0.215
1	0.341	0.268	0.555	1.116	0.205

nels. The detailed design of kernel window sizes and number is presented in Table 1 (slightly different from (Yu et al. 2017)). Without loss of generality, a kernel $w \in \mathbb{R}^{d \times l}$ with window size $d \times l$ applied to the concatenated feature map of the input image and sentence will produce a feature map $c = [c_1, c_2, \dots, c_{T-l+2}]$. Concretely, a specific feature is calculated as $c_i = \text{ReLU}(w * \varepsilon_{i:i+l-1} + b)$, where $i = \{1, 2, \dots, T-l+2\}$, $*$ is the convolution operation, b is a bias term, and $\text{ReLU}(\cdot)$ is the Rectified Linear Unit.

Then we apply a max-over-time pooling operation over the feature map $\tilde{c} = \max\{c\}$ and concatenate all the pooled features from different kernels to form a feature vector $\tilde{c} \in \mathbb{R}^n$ ($n = \sum_{i=1}^m n_i$). Following the instruction of (Yu et al. 2017), to enhance the performance, we also add the highway architecture (Srivastava, Greff, and Schmidhuber 2015) before the final fully connected layer:

$$\begin{aligned} \tau &= \sigma(W_T \cdot \tilde{c} + b_T) \\ H &= \text{ReLU}(W_H \cdot \tilde{c} + b_H) \\ \tilde{C} &= \tau \odot H + (1 - \tau) \odot \tilde{c}, \end{aligned} \quad (8)$$

where $W_T, W_H \in \mathbb{R}^{n \times n}$ and $b_T, b_H \in \mathbb{R}^n$ are highway layer weights and bias, respectively, σ is the sigmoid function and \odot is the piece-wise multiplication operation.

Finally, a fully connected layer and sigmoid transformation are applied to \tilde{C} to get the probability that a given sentence is real under a given image:

$$p = \sigma(W_o \cdot \tilde{C} + b_o), \quad (9)$$

where $W_o \in \mathbb{R}^{2 \times n}$ and $b_o \in \mathbb{R}^2$ are output layer weights and bias, respectively.

3.3 RNN-based Discriminator Model

Since the most commonly used sequence modeling network is an RNN, in this subsection, we present the RNN-based discriminator architecture that consists of the standard LSTM structure, a fully connected linear layer and a softmax output layer.

For the first time step, the image feature vector \bar{v} is fed into the LSTM as an input with the randomly initialized hidden state $h_0 \in \mathbb{R}^d$. Then, for the following time steps, the input vectors will be changed to token embeddings $E \cdot x_t$ ($t = \{1, 2, \dots, T\}$). The mathematical expressions are shown as follows:

$$h_{t+1} = \begin{cases} \text{LSTM}(\bar{v}, h_t) & t = 0 \\ \text{LSTM}(E \cdot x_t, h_t) & t = 1, 2, \dots, T \end{cases} \quad (10)$$

Then, after a fully connected layer and softmax layer, the probability that a given sentence is real under a given image can be calculated as:

$$p = \sigma(W_R \cdot h_{T+1} + b_R), \quad (11)$$

where $W_R \in \mathbb{R}^{2 \times n}$ and $b_R \in \mathbb{R}^2$ are linear layer weights and bias, respectively.

3.4 Algorithm

In order to incorporate the conditional GAN idea into the reinforcement learning method, the necessary back-propagated signal is the final reward r , as depicted in Fig. 1. Since the language evaluation score s is calculated by standard metric criterions (CIDEr, SPICE, etc.), the most important issue is the computation of discriminator probability output p . One straightforward way to train a conditional GAN is to train the discriminator to judge pairs (image, sentence) as real or fake. This type of conditioning is naive in the sense that the discriminator has no explicit notion of whether real training sentences match the extracted image features.

Typically, the discriminator observes three kinds of input pairs (one kind of positive samples and two kinds of negative samples): ground-truth sentences with matched images (real pairs $(I, x_{1:T})$), generated sentences with matched images (fake pairs $(I, \tilde{x}_{1:T})$) and ground-truth sentences with

Table 3: Metric selection

fixed parameters: $g=1; d=1; \lambda=0.3$					
Metric	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
CIDEr	0.381	0.280	0.580	1.248	0.213
CIDEr-D	0.383	0.286	0.586	1.232	0.221
BLEU-4	0.383	0.279	0.574	1.182	0.209
ROUGE-L	0.368	0.283	0.585	1.195	0.217
METEOR	0.377	0.287	0.576	1.180	0.214

Table 4: Step size combination selection

fixed parameters: $\lambda=0.3$; Metric=CIDEr-D					
Step Sizes	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
$g=1; d=5$	0.378	0.284	0.582	1.209	0.220
$g=1; d=1$	0.383	0.286	0.586	1.232	0.221
$g=5; d=1$	0.383	0.285	0.585	1.231	0.220
$g=10; d=1$	0.381	0.284	0.583	1.228	0.220

mismatched images (wrong pairs $(\mathbf{I}, \hat{\mathbf{x}}_{1:T})$). For the convenience of mathematical expression, we denote three sets as $\mathbb{S}_r, \mathbb{S}_f, \mathbb{S}_w$, which respectively consist of the three kinds of samples mentioned above.

Therefore, we slightly modify the traditional GAN training loss function (Goodfellow et al. 2014) to separate the two error sources. The model parameters ϕ of caption discriminator are trained to maximize

$$\begin{aligned}
 L_D(\phi) = & \mathbb{E}_{(\mathbf{I}, \mathbf{x}_{1:T}) \in \mathbb{S}_r} [\log D_\phi(\mathbf{I}, \mathbf{x}_{1:T})] \\
 & + 0.5 \cdot \mathbb{E}_{(\mathbf{I}, \tilde{\mathbf{x}}_{1:T}) \in \mathbb{S}_f} [\log(1 - D_\phi(\mathbf{I}, \tilde{\mathbf{x}}_{1:T}))] \\
 & + 0.5 \cdot \mathbb{E}_{(\mathbf{I}, \hat{\mathbf{x}}_{1:T}) \in \mathbb{S}_w} [\log(1 - D_\phi(\mathbf{I}, \hat{\mathbf{x}}_{1:T}))].
 \end{aligned} \tag{12}$$

Algorithm 1 describes the image captioning algorithm via the generative adversarial training method in detail. Notice that our proposed algorithm needs pre-training for both the generator and discriminator first. Then, we fine-tune the generator and discriminator alternatively based on the standard GAN training process.

4 Experiments

4.1 Dataset

The most widely used image captioning training and evaluation dataset is the MSCOCO dataset (Lin et al. 2014) which contains 82,783, 40,504, and 40,775 images with 5 captions each for training, validation, and test, respectively. For offline evaluation, following the Karpathy splits from (Karpathy and Fei-Fei 2015), we use a set of 5K images for validation, 5K images for test and 113,287 images for training. We adopt five widely used automatic evaluation metrics: BLEU, ROUGE-L, METEOR, CIDEr and SPICE, to objectively evaluate the performance of different algorithms. For online evaluation on the MSCOCO evaluation server, we add the 5K validation set and 5K testing set into the training set to form a larger training set.

All the sentences in the training set are truncated to ensure the longest length of any sentence is 16 characters. We follow standard practice and perform some text pre-processing, converting all sentences to lower case, tokenizing on white space, and filtering words that do not occur at least 5 times, resulting in a model vocabulary of 9,487 words.

4.2 Implementation Details

The LSTM hidden dimension for the RNN-based discriminator is 512. The dimension of image CNN feature and word embedding for both CNN-based and RNN-based discriminators is fixed to 2048. We initialize the discriminator via pre-training the model for 10 epochs by minimizing the cross entropy loss in Eq. (12) using the ADAM (Kingma and Ba 2014) optimizer with a batch size of 16, an initial learning rate of 1×10^{-3} and momentum of 0.9 and 0.999. Similarly, the generator is also pre-trained by MLE for 25 epochs. We use a beam search with a beam size of 5 when validating and testing. Notice that our proposed generative-adversarial-nets-based image captioning framework is generic so that any existing encoder-decoder model can be employed as the generator. In our experiments, the Resnet101 (He et al. 2016) and bottom-up mechanism (Anderson et al. 2018) based on fastar R-CNN are chosen as encoders, the top-down attention model (Anderson et al. 2018), att2in and att2all model (Rennie et al. 2017) are chosen as decoders to conduct controlled experiments. In addition, the SCST (Rennie et al. 2017) reinforcement learning method is adopted by all experiments. Therefore, all the generator and reinforcement learning hyper-parameters are identical with those of original referenced papers.

Finally, the parameters that are unique and need to be determined in our algorithm setting are the balance factor λ in Eq. (6), g -steps g and d -steps d during adversarial training in Algorithm 1, and the language evaluator Q . All the above

Table 5: Performance comparisons on MSCOCO Karpathy test set. The baseline algorithms are using resnet101 or bottom-up mechanism as the image feature extractor and SCST as the training method. Results of algorithms denoted by * are provided by original papers and the remaining experimental results are implemented by us for comparison. “None” means RL training method without discriminator. “CNN-GAN” and “RNN-GAN” mean training with our proposed approach by CNN-based and RNN-based discriminator, respectively. “Ensemble” indicates an ensemble of 4 CNN-GAN and 4 RNN-GAN models with different initializations. All values are reported in percentage (%).

Generator	Discriminator	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	CNN-D	RNN-D
resnet101+att2in (Rennie et al. 2017)	none*	-	-	-	33.3	26.3	55.3	111.4	-	-	-
	CNN-GAN	78.1	61.3	46.4	34.4	26.6	56.1	112.3	20.3	47.9	45.8
	RNN-GAN	78.0	61.4	46.3	34.3	26.5	56.0	112.2	20.4	46.0	48.1
	ensemble	78.5	61.8	47.1	35.0	27.1	56.6	114.8	20.5	48.0	48.2
bottom-up+att2in (Rennie et al. 2017)	none	79.0	62.1	48.2	35.5	27.0	56.3	117.0	20.9	45.6	44.5
	CNN-GAN	80.1	63.8	49.0	37.0	27.9	57.7	118.0	21.4	51.2	49.7
	RNN-GAN	80.0	63.9	49.1	36.8	27.8	57.6	118.1	21.3	49.5	51.9
	ensemble	80.5	64.8	50.0	37.9	28.4	58.2	119.5	21.5	51.4	51.5
resnet101+att2all (Rennie et al. 2017)	none*	-	-	-	34.2	26.7	55.7	114.0	-	-	-
	CNN-GAN	78.4	62.6	47.6	35.4	27.4	56.8	115.2	20.6	49.0	47.2
	RNN-GAN	78.3	62.5	47.6	35.2	27.3	56.9	115.1	20.6	47.1	48.8
	ensemble	79.0	62.8	48.2	35.8	27.7	57.6	117.8	20.9	49.5	49.1
bottom-up+att2all (Rennie et al. 2017)	none	79.6	63.5	49.1	36.1	27.8	56.7	119.8	21.2	46.3	45.9
	CNN-GAN	80.7	64.7	50.1	38.0	28.4	58.4	122.1	21.9	53.5	50.8
	RNN-GAN	80.6	64.8	50.0	38.1	28.3	58.3	122.0	21.8	50.6	53.2
	ensemble	81.1	65.7	50.8	39.0	28.6	58.7	124.1	22.0	53.7	53.5
resnet101+top-down (Anderson et al. 2018)	none*	76.6	-	-	34.0	26.5	54.9	111.1	20.2	-	-
	CNN-GAN	78.5	62.7	48.0	35.6	27.3	56.7	113.0	20.6	49.5	47.6
	RNN-GAN	78.4	62.7	48.0	35.5	27.2	56.6	112.7	20.5	47.0	49.2
	ensemble	79.3	63.2	48.6	36.0	27.6	57.1	115.5	20.8	50.0	49.3
bottom-up+top-down (Anderson et al. 2018)	none*	79.8	-	-	36.3	27.7	56.9	120.1	21.4	-	-
	CNN-GAN	81.1	65.0	50.4	38.3	28.6	58.6	123.2	22.1	53.6	51.1
	RNN-GAN	81.0	64.8	50.2	38.2	28.5	58.4	122.2	22.0	50.9	54.0
	ensemble	81.8	66.1	51.6	39.6	28.9	59.1	125.9	22.3	54.3	54.5
Average Improvements	CNN-GAN	1.71	2.31	1.85	4.44	2.59	2.53	1.50	2.75	13.93	11.17
	RNN-GAN	1.59	2.47	1.85	4.15	2.22	2.38	1.28	2.27	8.92	16.26

hyper-parameters are empirical values and will be clarified in the following subsection.

4.3 Parameters Determination

In order to determine a group of optimal hyper-parameters as mentioned in the above subsection, we design a series of experiments with a variable-controlling approach. We adopt the bottom-up image feature extractor together with top-down attention model as our fixed generator, SCST as our RL optimization method and the CNN structure as our discriminator.

First, we fix the g-steps $g = 1$, d-steps $d = 1$ and language metric Q as CIDEr-D (a smooth modification version of CIDEr). The objective results on the test split with different λ values are shown in Table 2. Notice that when $\lambda = 0$, our algorithm exactly degenerates into the conventional RL method. Statistics reveal that all the metrics evaluation results achieve their optimal scores when $\lambda = 0.3$.

Second, we fix the g-steps $g = 1$, d-steps $d = 1$ and $\lambda = 0.3$. The test results while RL optimizing by different language evaluator Q are shown in Table 3. Notice that here we do not choose SPICE as evaluator since the computation of SPICE is extremely slow and is too time-consuming. When the evaluator is chosen as CIDEr-D, even though some scores (CIDEr and METEOR) are not the highest, the

comprehensive performance still outperforms other evaluators.

Third, we fix $\lambda = 0.3$ and language evaluator Q as CIDEr-D. We try different step-size combinations and list the test results in Table 4. Experimental results demonstrate that the best step-size combination is $g = 1$ and $d = 1$.

Overall, based on the experimental results explained above, the final optimal hyper-parameters of our proposed algorithm are $\lambda = 0.3$, $g = 1$, $d = 1$ and $Q = \text{CIDEr-D}$.









4.4 Comparisons

We compare our framework with some state-of-the-art encoder-decoder models (att2in (Rennie et al. 2017), att2all (Rennie et al. 2017) and top-down attention (Anderson et al. 2018)) with SCST (Rennie et al. 2017) to study the effectiveness of our proposed algorithm. For fair comparisons, we chose Resnet101 (He et al. 2016) and bottom-up mechanism (Anderson et al. 2018) as the CNN feature extractors for all the models mentioned above and the identical decoder parameters as reported in the original papers. Statistics reported in Table 5 reveal that by using our proposed generative adversarial training method, the performance of all the objective evaluation metrics is improved for all three models. Specifically, the relative improvements range from 1.3% to 4.4% on average. Notice that the well-trained CNN-

Table 6: Performance of different models on the MSCOCO evaluation server. All values are reported in percentage (%), with the highest value of each entry highlighted in boldface. It is worth pointing out that almost all the metrics of our method (ensemble of 4 CNN-GAN and 4 RNN-GAN models in the last row) ranked in top two at the time of submission (5 Sep., 2018).

Algorithms	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr		SPICE	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
NIC (Vinyals et al. 2015)	71.3	89.5	54.2	80.2	40.7	69.4	30.9	58.7	25.4	34.6	53.0	68.2	94.3	94.6	18.2	63.6
PG-BCMR (Liu et al. 2017)	75.4	91.8	59.1	84.1	44.5	73.8	33.2	62.4	25.7	34.0	55.0	69.5	101.3	103.2	18.7	62.2
Adaptive (Lu et al. 2017)	74.8	92.0	58.4	84.5	44.4	74.4	33.6	63.7	26.4	35.9	55.0	70.5	104.2	105.9	19.7	67.3
Actor-Critic (Zhang et al. 2017)	77.8	92.9	61.2	85.5	45.9	74.5	33.7	62.5	26.4	34.4	55.4	69.1	110.2	112.1	20.3	68.0
Att2all (Rennie et al. 2017)	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.7	20.7	68.9
Stack-Cap (Gu et al. 2017)	77.8	93.2	61.6	86.1	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3	-	-
LSTM-A ₃ (Yao et al. 2017)	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0	-	-
Up-down (Anderson et al. 2018)	80.2	95.2	64.1	88.8	49.1	79.4	36.9	68.5	27.6	36.7	57.1	72.4	117.9	120.5	21.5	71.5
CAVP (Liu et al. 2018)	80.1	94.9	64.7	88.8	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8	-	-
RFNet (Jiang et al. 2018)	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1	-	-
Ours	81.9	95.6	66.3	90.1	51.7	81.7	39.6	71.5	28.7	38.2	59.0	74.4	123.1	124.3	-	-

Table 7: This table shows some caption examples by different algorithms. The first line is generated by the Up-Down model with SCST. The second and third lines are caption results by adding CNN-GAN and RNN-GAN training, respectively. The last line is the results of ensemble of CNN-GAN and RNN-GAN.

				
Up-Down	a bag of items and other items on a table	a man with a beard and tie wearing a tie	a desk with two laptops and a laptop	a group of zebras and a zebra standing in the water
CNN-GAN	a purse and personal items laid out on a wooden table	a man in a suit and tie looking at the camera	a desk with a laptop computer and a desktop on it	a group of zebras and other animals in the water
RNN-GAN	a purse and other items laid out on a wooden table	a man in a suit and tie is smiling	a desk with a laptop computer and books on it	a group of zebras and other animals standing in the water
ensemble	a purse and other personal items laid out on a wooden table	a man in a suit and tie looking at the camera with smile	a desk with a laptop and a desktop sitting on top of it	a group of zebras and other animals standing near the water
				
Up-Down	a woman holding an umbrella in a brick wall	a woman standing in front of a cell phone	two giraffes standing next to a city in the water	a group of people standing on top of a clock
CNN-GAN	a woman in a yellow jacket holding an umbrella	a woman standing in front of a newspaper sign	two giraffes standing next to a large city in the background	a group of people standing on a building with a clock
RNN-GAN	a woman standing in front of a brick wall holding an umbrella	a woman standing in front of a store holding a cell phone	two giraffes standing next to a city in the background	a group of people standing on a balcony looking at a clock
ensemble	a woman in a yellow jacket near a brick wall holding an umbrella	a woman standing in front of a newspaper sign holding a cell phone	two giraffes standing next to a city near water in the background	a group of people standing on a balcony with a clock

based and RNN-based discriminators (CNN-D and RNN-D in the last two columns of Fig. 5) can also be viewed as two learned evaluators. The experimental results demonstrate that our proposed adversarial training approach can significantly boost the scores compared with traditional RL training method (improvements range from 8.9% to 16.3%).

In terms of CNN-based and RNN-based generative adversarial training framework (called CNN-GAN and RNN-GAN in Table 5), each has its own advantages. Experimental results show that CNN-GAN can improve the performance of image captioning frameworks slightly more as compared with RNN-GAN. However, during training stage, using RNN-GAN can save 30% training time according to

our experimental experience. The most important issue is that the ensemble results of 4 CNN-GAN and 4 RNN-GAN models can largely enhance the performance of a single model as shown in Table 5.

For online evaluation, we use the ensemble of 4 CNN-GAN and 4 RNN-GAN models with different random initializations whose generator structure exactly follows the Up-Down model (Anderson et al. 2018) and the comparisons are provided in Table 6. We can see that our approach achieves the best results compared with some state-of-the-art algorithms with publicly published papers. For all submissions online, almost all the metrics of our method ranked in top two at the time of submission (5 Sep., 2018).

Specifically, when compared with the results of Up-Down model, by using our proposed generative adversarial training method, it can obtain significant improvements.

4.5 Examples

To better understand the CNN-GAN and RNN-GAN framework, Table 7 provides some representative examples for comparisons. The baseline algorithm is the Up-Down model with SCST, whose results are shown in the first line. Our generated captions by CNN-GAN and RNN-GAN training approach with the same generator are listed in the second and third line, respectively. The last row is the results of ensemble of 4 CNN-GAN and 4 RNN-GAN models.

The four cases appearing above in Table 7 indicate that the original model will generate some duplicate words or phrase. Even without any grammar or description errors, these sentences seem rigid and machine generated. After generative adversarial training, the generated sentences are much more like a human style. The four cases appearing below in Table 7 present some logical errors which are inconsistent with given images when using traditional RL training method, *e.g.* “in a brick wall”, “in front of a cell phone”, “city in the water”, “on top of a clock”, while such error will be avoided when employing our proposed method.

5 Conclusion

This paper proposes a novel architecture combining generative adversarial nets and reinforcement learning to improve existing image captioning frameworks. Current RL-based image captioning algorithms directly optimize language evaluation metrics such as CIDEr, BELU and SPICE; however, simply optimizing one metric or combination of these metrics will not consistently improve all evaluation metrics and will also result in some logical errors or unnatural styles when generating descriptions. Therefore, we are motivated to design a discriminator network to judge whether the input sentence is human described or machine generated. Alternatively, the caption generator and the evaluation discriminator are improved by adversarial training. We experimentally show consistent improvements over all language evaluation metrics for different state-of-the-art encoder-decoder based image captioning models optimized by conventional RL training, demonstrating the effectiveness of our proposed generative adversarial training method. In addition, the well-trained CNN and RNN-based discriminators can also be utilized as image captioning evaluators leaned by neural networks.

References

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, 382–398. Springer.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, 6.

Banerjee, S., and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and ex-*

trinsic evaluation measures for machine translation and/or summarization, 65–72.

Bengio, S.; Vinyals, O.; Jaitly, N.; and Shazeer, N. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, 1171–1179.

Dai, B.; Fidler, S.; Urtasun, R.; and Lin, D. 2017. Towards diverse and natural image descriptions via a conditional GAN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2970–2979.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Gu, J.; Cai, J.; Wang, G.; and Chen, T. 2017. Stack-captioning: Coarse-to-fine learning for image captioning. *arXiv preprint arXiv:1709.03376*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jiang, W.; Ma, L.; Jiang, Y.-G.; Liu, W.; and Zhang, T. 2018. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 499–515.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3128–3137.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Liu, S.; Zhu, Z.; Ye, N.; Guadarrama, S.; and Murphy, K. 2017. Improved image captioning via policy gradient optimization of SPI-DEr. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 873–881.

Liu, D.; Zha, Z.-J.; Zhang, H.; Zhang, Y.; and Wu, F. 2018. Context-aware visual policy network for sequence-level image captioning. *arXiv preprint arXiv:1808.05864*.

Lu, J.; Xiong, C.; Parikh, D.; and Socher, R. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 375–383.

Mao, J.; Xu, W.; Yang, Y.; Wang, J.; and Yuille, A. L. 2014. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

- Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.
- Ren, Z.; Wang, X.; Zhang, N.; Lv, X.; and Li, L.-J. 2017. Deep reinforcement learning-based image captioning with embedding reward. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 290–298.
- Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7008–7024.
- Shetty, R.; Rohrbach, M.; Hendricks, L. A.; Fritz, M.; and Schiele, B. 2017. Speaking the same language: Matching machine to human captions by adversarial training. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.
- Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.
- Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4566–4575.
- Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3156–3164.
- Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*. Springer. 5–32.
- Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; and Mei, T. 2017. Boosting image captioning with attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, 4894–4902.
- Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2852–2858.
- Zhang, X., and LeCun, Y. 2015. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.
- Zhang, L.; Sung, F.; Liu, F.; Xiang, T.; Gong, S.; Yang, Y.; and Hospedales, T. M. 2017. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*.