Actor-Critic Sequence Training for Image Captioning

Li Zhang
Queen Mary University of London
david.lizhang@qmul.ac.uk

Flood Sung
Independent Researcher
floodsung@gmail.com

Feng Liu **Tao Xiang** **Shaogang Gong** **Yongxin Yang**
Queen Mary University of London
{feng.liu, t.xiang, s.gong, yongxin.yang}@qmul.ac.uk

Timothy M. Hospedales
The University of Edinburgh
t.hospedales@ed.ac.uk

Abstract

Generating natural language descriptions of images is an important capability for a robot or other visual-intelligence driven AI agent that may need to communicate with human users about what it is seeing. Such image captioning methods are typically trained by maximising the likelihood of ground-truth annotated caption given the image. While simple and easy to implement, this approach does not directly maximise the language quality metrics we care about such as CIDEr. In this paper we investigate training image captioning methods based on actor-critic reinforcement learning in order to directly optimise non-differentiable quality metrics of interest. By formulating a per-token advantage and value computation strategy in this novel reinforcement learning based captioning model, we show that it is possible to achieve the state of the art performance on the widely used MSCOCO benchmark.

1 Introduction

As the classic task of automatic object category recognition is beginning to approach a solved problem [20], interest is growing in solving a more 'end-to-end' task of generating richer descriptions of images in terms of natural language, suitable for communication to human users [22, 23, 27, 10]. This task is extremely topical recently, benefiting from public benchmarks such as MSCOCO [9]. Despite extensive research in recent years, leading performance on the benchmarks has not increased dramatically. We hypothesise that this is mainly due to research focus being on the image understanding aspects of captioning, rather than the language generation aspects, and in this paper investigate reinforcement learning methods for training effective language generation in captioning.

Most existing captioning studies investigate variants of deep learning-based image encoders, that feed into deep sentence decoders. They have two main issues: (i) They are trained by maximising the likelihood of each ground-truth word given the previous ground-truth words and the image using back-propagation [16], termed 'Teacher-Forcing' [3]. This 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. This exposure bias [16], results in error accumulation during generation at test time, since the model has never been exposed to its own predictions. (ii) While sequence models are usually trained using the cross entropy loss, the actual NLP quality metrics of interest – with which we evaluate them at test time – are non-differentiable metrics such as CIDEr [21]. Ideally sequence models for image captioning should be trained to avoid exposure bias and directly optimise metrics for the task at hand.

In this paper we propose to improve image captioning by addressing the above identified two issues through training captioning models with reinforcement learning. In this way, we can optimise the

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

gradient of the expected reward by sampling from the model during training, thus avoiding the train-test mismatch; and we can directly optimise the relevant test-time metrics such as CIDEr, by treating them as reward in a reinforcement learning context.

Specifically we propose an actor-critic model for image captioning. It consists of a policy network (actor) and value network (critic). The actor is trained to predict the caption as a sequential decision problem given the image, where the sequence of actions correspond to tokens. The critic predicts the value of each state (image and sequence of actions so far), which we define as the expected task-specific reward (language metric score) that the network will receive if it outputs the current token and continues to sample outputs according to its probability distribution. The value predicted by the critic can be used to train the actor (captioning policy network). Under the assumption that the critic produces the exact values, the actor is trained based on an unbiased estimate of the gradient of the caption score in terms of relevant language quality metrics. Compared to most reinforcement learning applications [14], image captioning has a much higher dimension action (e.g., 10,000+ token/word actions) space but shorter episodes. The proposed actor-critic approach exploits the shorter episodes and ameliorates the high dimensional action space. Our model achieves state-of-the-art performance: It is ranked third on the MS-COCO testing server leaderbord when submitted, which is the highest rank achieved by reinforcement learning based method without model ensemble.

2 Related Work

Image captioning There is now extensive work on image captioning [22, 6, 26, 5, 12, 25, 23, 10]. The typical pipeline is based on a convolutional neural network (CNN) image encoder and a recurrent neural network (RNN) based sentence decoder [22, 23]. Topical issues addressed in the literature include dynamic attention [27, 26], improving visual feature representations [10]. As mentioned earlier they are typically trained by maximising training caption likelihood through teacher forcing.

Image captioning with reinforcement learning Recently a few studies proposed to use reinforcement learning to address the discrepancy between the standard training objective for image captioning (likelihood/teacher forcing) and the evaluation metrics of interest (CIDEr) [16, 11, 17]. [17] uses the basic REINFORCE algorithm [24] with a reward obtained by the current model under the inference algorithm as the baseline. As a results, for each sampled caption, it has only one sentence level advantage which means that every token makes the same contribution towards the whole sentence – a clearly invalid assumption. [16, 11] add an additional FC layer on top of the RNN output to predict state value function. However, both treat *state* as the RNN output while we treat the *state* as the RNN input (given image and the taken actions), so that we can build an independent value network rather than a shared RNN cell between actor and critic.

Actor-Critic An actor-critic [2] method trains the actor by policy gradient with advantages baselined by the critic. Many state-of-the-art reinforcement learning algorithms [13] are based on actor-critic. For instance, AlphaGo [19] utilised the actor-critic method to do self-learning in the game of Go and achieved great success by beating human world champions. It uses Monte-Carlo rollout and the reward is only set at the end of the game with very long episode.

Sequence generation Our task is related to sequence generation [1, 28, 15]. [1] uses actor-critic for machine translation. Their actor and critic have the same encoder-decoder architecture while the critic outputs state-action value function for each possible actions for policy iteration. Given that the action space is huge for a sequence generation task, the predicted action-value function often have to rely on various tricks to penalising the variance of the outputs of the critic. Without the penalty the values of rare actions can be severely overestimated, introducing bias to the gradient estimates and causing convergence difficulties. [28] uses Monte Carlo rollout to sample actions and uses GAN to compute reward. [15] applies the same method as [17] to improve the performance of abstractive summarisation.

3 Methodology

3.1 Problem formulation

We consider the problem of learning to generate caption sequence $Y = \{y_1, \ldots, y_T\}, y_t \in \mathcal{D}$ given an image I, where $\mathcal{D}$ is the dictionary. To simplify the formulas we always use T to denote the length

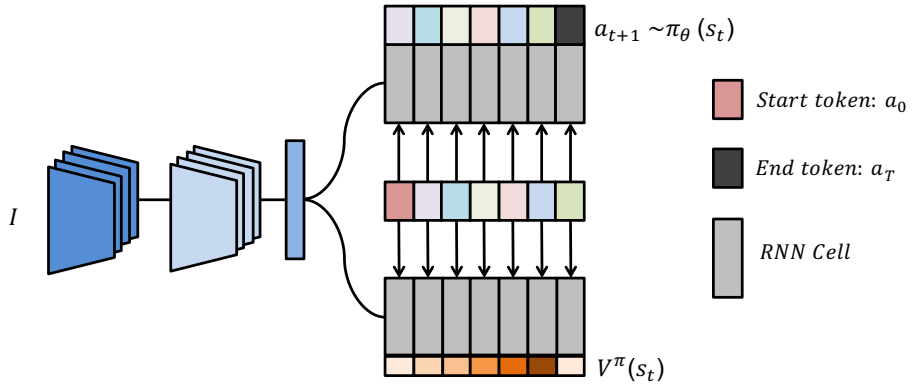


Figure 1: Schematic illustration of our actor-critic based captioning model (with word embedding layer omitted).

of an output sequence, ignoring the fact that the generated caption sequences may have different lengths. Two sets of input-output pairs (I, Y) are assumed to be available for both training and testing. The trained sequence generative model is evaluated by computing the task-specific score $R(\hat{Y}, Y)$ (e.g., BLEU, CIDEr) on the test set, where $\hat{Y}$ is the predicted caption sequence.

In this work we utilise the conventional encoder-decoder architecture (see Figure 1) for image captioning which consists of a Convolutional Neural Network (CNN) as the encoder and a Recurrent Neural Network (RNN) as the decoder. In order to transform the image captioning problem into a reinforcement learning task, we consider the image caption generation process as a finite Markov decision process (MDP) $\{S, A, P, R, \gamma\}$. In the MDP setting, the state S is composed of the image feature I_e encoded by the CNN from image I and the tokens/actions $\{a_0, a_1, \ldots, a_t\}$ that are generated so far. With the definition of the state, the state transition function P is $s_{t+1} = \{s_t,\ a_{t+1}\}$, where the action a_{t+1} becomes a part of the next state s_{t+1} and the reward r_{t+1} is received. $\gamma \in [0, 1]$ is the discount factor. Under the MDP interpretation of the image captioning problem, we can apply standard reinforcement learning algorithms to maximise the cumulated reward.

3.2 Model

We train our model using actor-critic [2] reinforcement learning method which contains a policy network (actor) and a value network (critic). In particular, we use Inception-V3 [20] as the CNN subnet and Long Short-Term Memory [8] as the RNN subnet. Both the policy network and value network are based on LSTM for sequential action or value generation.

Policy network The policy network π is parametrised by θ and at time t it receives a state s_t and generates the categorical distribution over $|\mathcal{D}|$ actions (tokens), i.e. $a_{t+1} \sim \pi_\theta(s_t)$. We encode the given image I to I_e by CNN and treat I_e and the start token a_0 as the initial state s_0:

$$s_0 = \{\mathrm{I}_e,\ a_0\}. \tag{1}$$

With state transfer function mentioned above, we have:

$$s_t = \{\mathrm{I}_e, a_0, a_1, \ldots, a_t\}. \tag{2}$$

We feed state s_t into the LSTM and obtain the LSTM hidden state h_{t+1} (I_e was set as h_0). In order to build a probabilistic model for caption generation with an LSTM, we add a stochastic output layer f (typically with the softmax activation for discrete outputs) that generates outputs $a_{t+1} \in \mathcal{D}$:

$$\begin{aligned} \mathrm{h}_{t+1} &= \mathrm{LSTM}(s_t), \\ a_{t+1} &\sim f(\mathrm{h}_{t+1}). \end{aligned}$$

Thus, the policy network defines a probability distribution $p(a_{t+1}|s_t)$ of the action a_{t+1} given current state s_t. The architecture of the policy network is same as the standard supervised learning. Therefore,

by given a target ground truth sequence $\{y_0, y_1, \ldots, y_T\}$, the supervised learning approach would be to train this network by minimising the cross entropy loss (XE).

$$\mathcal{L}_{\mathrm{XE}}(\theta) = -\sum_{t=1}^{T} \log(\pi_\theta(y_t \mid y_0, \ldots, y_{t-1})). \tag{3}$$

This corresponds to imitation learning of a perfect teacher in a RL context, and we use the pre-trained model as the initial policy network.

Policy gradient training Policy gradient methods maximise the expected cumulated reward by repeatedly estimating the gradient $g := \nabla_\theta \mathbb{E}[\sum_{t=1}^{T} r_t]$, where the environment issues the reward r_t according to the efficacy of the produced actions, rather than the teacher demonstrating the ideal actions directly as in Eq 3. For policy gradient, it is typically better to train an expression of the form:

$$g = \mathbb{E}[\sum_{t=0}^{T-1} A^\pi(s_t,\ a_{t+1}) \nabla_\theta \log \pi_\theta(a_{t+1} \mid s_t)], \tag{4}$$

where $A^\pi(s_t,\ a_{t+1})$ is advantage function yields almost the lowest possible variance, though in practice, the advantage function is not known and must be estimated. This statement can be intuitively justified by the following interpretation of the policy gradient: that a step in the policy gradient direction should increase the probability of better-than-average actions and decrease the probability of worse-than-average actions. The advantage function, by it's definition $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$, measures whether or not the action is better or worse than the policy's default behaviour. So that the gradient term $A^\pi(s_t,\ a_{t+1}) \nabla_\theta \log \pi_\theta(a_{t+1} \mid s_t)$ points in the direction of increased $\pi_\theta(a_{t+1} \mid s_t)$ if and only if $A^\pi(s_t,\ a_{t+1}) > 0$.

Value network Given the policy π, sampled actions and reward function, the value represents the expected future return as a function of the observed state s_t. We use V be an approximate state-value function.

$$V^\pi(s_t) = \mathbb{E}[\sum_{l=0}^{T-t-1} \gamma^l r_{t+l+1} \mid a_{t+1}, \ldots, a_T \sim \pi,\ I], \tag{5}$$

where discount factor γ allows us to reduce variance by down-weighting rewards, at the cost of introducing bias. This parameter corresponds to the discount factor used in discounted formulations of MDPs.

Our value network can be seen as an encoder. We propose to use a separate LSTM parametrised by ϕ with shared CNN. The RNN consumes state $s_t = \{\mathrm{I}_e, a_0, a_1, \ldots, a_t\}$ and produces a single value output to predict the TD target (to be defined later in Sec 3.3).

3.3 Advantage function estimation

We use temporal-difference (TD) learning for advantage function estimation. Specifically, we define $Q^\pi(s_t, a_{t+1})$ in forward-view TD(λ) setting:

$$Q^\pi(s_t, a_{t+1}) = (1 - \lambda) \sum_{n=1}^{\infty} G_t^n, \tag{6}$$

where G_t^n is the n-step expected return:

$$G_t^n = r_{t+1} + \gamma r_{t+2} + ... + \gamma^{n-1} r_{t+n} + \gamma^n V^\pi(s_{t+n}). \tag{7}$$

Therefore we have:

$$A^\pi(s_t, a_{t+1}) = Q^\pi(s_t, a_{t+1}) - V^\pi(s_t) = (1 - \lambda) \sum_{n=1}^{\infty} G_t^n - V^\pi(s_t), \tag{8}$$

which is the same definition as GAE [18] but in a forward view. Then the gradient of policy network has the form:

$$g = \mathbb{E}[\sum_{t=0}^{T-1} \big((1 - \lambda) \sum_{n=1}^{\infty} G_t^n - V^\pi(s_t)\big) \nabla_\theta \log \pi_\theta(a_{t+1} | s_t)]. \tag{9}$$

3.4 Value function estimation

When using a nonlinear function approximator to represent the value function, the simplest approach is to solve a nonlinear regression problem:

$$\min_{\phi} ||Q^{\pi}(s_t, a_{t+1}) - V_{\phi}^{\pi}(s_t)||^2 \tag{10}$$

where $Q^{\pi}(s_t,\ a_{t+1}) = (1-\lambda)\sum_{n=1}^{\infty} G_t^n$.

3.5 λ setting for image captioning

λ setting plays an important role for the whole algorithm. If $\lambda = 0$, the advantage and value function estimations become one-step TD, whereas if $\lambda = 1$, the estimations turn out to be Monte Carlo approach. Since the episode length of image captioning is relatively shorter than popular contemporary RL problems (e.g. Atari and Mujoco games), and we have to sample the whole sequence of captions for rewarding, we set $\lambda = 1$ for our image captioning problem. Under this setting, the estimator for both advantage and value function is unbiased and the limited length of episode restricts the variance of estimation to a limited range. Concretely, with $\lambda = 1$, we have:

$$Q^{\pi}(s_t,\ a_{t+1}) = \sum_{l=0}^{T-t-1} \gamma^l r_{t+l+1} \tag{11}$$

3.6 Reward

For image captioning we can only obtain an evaluation score (e.g. CIDEr) when the caption generation process is finished. Therefore, we define the reward as follows:

$$r_t = \begin{cases} 0 & t < T \\ \text{score} & t = T \end{cases} \tag{12}$$

under such reward setting, we have

$$Q^{\pi}(s_t,\ a_{t+1}) = \gamma^{T-t-1} r_T. \tag{13}$$

Then the gradient of policy network has a sample form:

$$g = \mathbb{E}[\sum_{t=0}^{T-1} \left(\gamma^{T-t-1} r_T - V(s_t)\right) \nabla_{\theta} \log \pi_{\theta}(a_{t+1} \mid s_t)]. \tag{14}$$

4 Experiments

4.1 Implementation details

We use Inception-v3 [20] as the CNN subnet, and an LSTM network is used as the RNN subnet. The number of LSTM cells is 512, equalling to the dimension of the word embedding. The output vocabulary size for sentence generation is 12,000. Note that all these are exactly the same as the NICv2 [22] model ensuring a fair comparison.

For the CNN feature we used, semantic concept [10] feature $\mathrm{I} \in \mathbb{R}^{1000}$ is used. These 1,000 semantic concepts are mined from the most frequents words in a set of image captions. A concept classifier is learned to predict Is as classification scores for the concepts.

Algorithm 1 describes the proposed method in detail. Our preliminary experiments show that training actor-critic from scratch can lead to an early determinisation of the policy and vanishing gradients, because neither the actor nor the critic would provide adequate training signals for one another. The actor would sample completely random tokens that receive very low reward, thus providing a very weak training signal for the critic. A random critic would be similarly useless for training the

actor. To overcome these problems, staged pre-trainings are carried out. More specifically, we first pre-train the actor using standard cross entropy loss (XE) (see Eq. 3). After that, we pre-train the critic network by feeding it with sampled actions from the fixed pre-trained actor. The critic network is pre-trained for 2,000 iterations using Adam with a learning rate of 5e-5. For the final stage of joint training of actor-critic, we weight critic loss by 0.5. We use Adam with an initial learning rate of 5e-5 and decrease it to 5e-6 after 1 million iterations, with minibatch size 16. We set discount factor $\gamma = 1$ in all our experiments. The complete training procedure including pre-training is described by Algorithm 2.

Algorithm 1: Actor-Critic Training for Image Captioning

Require: Actor $\pi(a_{t+1}|s_t)$ and critic $V(s_t)$ with weights θ and ϕ respectively;
for *episode = 1 to max episode* **do**
  Receive a random example (I, Y) and sample sequence of actions $\{a_1, \ldots, a_T\}$ according to current policy π_θ;
  Compute TD target $Q^\pi(s_t,\ a_{t+1}) = \gamma^{T-t-1} r_T$ for $V(s_t)$;
  Update critic weights ϕ by minimising Eq. 10;
  Update actor weights θ using the gradient in Eq. 14;
end

Algorithm 2: Complete Actor-Critic Algorithm for Image Captioning

Initialise actor $\pi(a_{t+1}|s_t)$ and critic $V(s_t)$ with random weights θ and ϕ respectively;
Pre-train the actor to predict ground truth y_t given $\{y_1, \ldots, y_{t-1}\}$ by minimise Eq. 3;
Pre-train the critic to estimate $V(s_t)$ by running Algorithm 1 with fixed actor;
Run Algorithm 1

4.2 Datasets and setting

We evaluate the proposed method on the most widely used MSCOCO [9] dataset. The dataset contains 82,783 training images and 40,504 validation images. Each image is manually annotated with about 5 captions. The comparison against the state-of-the-art is conducted using the actual MS COCO test set comprising 40,775 images. Note that the annotation of the test set is not publicly available, so the results are obtained from the COCO evaluation server. We also follow the setting of [22, 23] by using a held-out set of 4,051 images from the COCO validation set as the development set. The widely used BLEU, CIDEr, METEOR, and ROUGE scores are employed to measure the quality of generated captions.

4.3 Experimental results

Competitors Several state-of-the-art models are selected for comparison: MSRCap: The Microsoft Captivator [5] combines the bottom-up based word generation model [6] with a gated recurrent neural network [4] (GRNN) for image captioning. mRNN: The multimodal recurrent neural network [12] uses a multimodal layer to combine the CNN and RNN. NICv2: The NICv2 [23] is an improved version of the Neural Image Caption generator [22]. It uses a better image encoder, i.e., Inception-v3. In addition, scheduled sampling [3] and an ensemble of 15 models are used; both improved the accuracy of captioning. V2L: The V2L model [25] uses a CNN based attribute detector to firstly generate 256 attributes, and then feed as initial input to an LSTM model to generate captions. ATT: The semantic attention model [27] uses both image features and visual attributes, and introduces an attention mechanism to reweight the attribute context to improve captioning accuracy. Semantic [10] is our base model which uses a semantically regularised embedding layer as the interface between the CNN and RNN.

In addition to the traditional supervised learning method, we compare our method with three reinforcement learning based model. PG [11] and MIXER [16] use policy gradient method with an additional FC layer on top of the RNN as state value network to reduce the high variance. Different from [16], [11] uses the same method as [28] with k-times Monte Carlo rollout to estimate the state

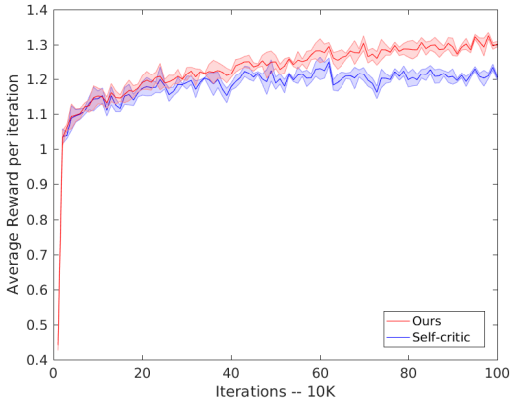


Figure 2: Average training reward curve for ours and [17]. We recorded the reward for 1 million iteration and plot every 10k iteration.

value target. Self-critical [17] uses the basic REINFORCE algorithm with a reward obtained by the current model under the inference algorithm as the baseline. This simple method achieved very high performance which ranked 2nd currently on coco captioning challenge. However, it use a multiple model ensemble. In contrast, only a single model is used for our method.

Results The results on development set are summarised in Table 1. We report a significant improvement from 1.007 to 1.162 on CIDEr over the log-likelihood baseline when single model greedy search is used for decoding. We can also see that our method is better than attention [26] and memory cell [7] which are added on top of the LSTM cell. For fair comparison with the current state-of-the-art method [17], we implement it with same semantic CNN input [10] on development set with single model for evaluation. The training reward curves of our method and [17] are shown in Figure 2. Our method is clearly superior to that of [17] due to the per-token advantage and value computation strategy adopted in our actor-critic based reinforcement learning framework.

Metric	**CIDEr-D**	**BLEU-4**	**METEOR**	**ROUGE-L**
NIC [22]	0.855	0.277	0.237	-
NICv2 [23]	0.998	0.321	0.257	-
Semantic [10]	1.007	0.302	0.256	0.539
Semantic [10]+Attention [26]	1.042	0.311	0.263	0.543
Semantic [10]+Attention [26]+Memory [7]	1.057	0.318	0.266	0.547
Semantic [10]+Self-critical [17]	1.140	0.323	0.266	0.554
Ours	**1.162**	**0.344**	**0.267**	**0.558**

Table 1: Single model greedy search scores on the MSCOCO development set

We also submitted our single model results to the official evaluation server to compare with the eight baselines mentioned above. The evaluation is done with both 5 and 40 reference captions (C5 and C40). Our model is ranked the 3rd on the MSCOCO image captioning challenge leaderboard when submitted. Table 2 shows that, compare to the supervised learning based methods, our approach significantly outperforms all of them in all metrics despite using only a single model rather than model ensemble. Comparing to the other two reinforcement learning based methods [11, 16], our method still achieved better performance except ROUGE-L c40. Figure 3 shows some qualitative examples of our models captioning compared against ground truth (Human) and method using the same encoder-decoder architecture, but with standard cross-entropy (XE) training.

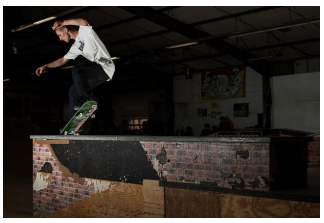
Computational Cost We compare the training time of our method with several alternatives. All algorithms are implemented in *Tensorflow* and run on an NVIDIA P100 card, with a mini-batch size of 16. Table 3 shows that for training, our method is the most efficient one. This is mainly due to the fact that our model does not have attention cell. Furthermore, for the Self-critical [17] model, it needs to sample twice (random sampling + greedy decoding) for each iteration which is expensive.

Metric	B-1		B-2		B-3		B-4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
MSRCap [5]	0.715	0.907	0.543	0.819	0.407	0.710	0.308	0.601	0.248	0.339	0.526	0.680	0.931	0.937
mRNN [12]	0.716	0.890	0.545	0.798	0.404	0.687	0.299	0.575	0.242	0.325	0.521	0.666	0.917	0.935
V2L [25]	0.725	0.892	0.556	0.803	0.414	0.694	0.306	0.582	0.246	0.329	0.528	0.672	0.911	0.924
NICv2 [23]	0.713	0.895	0.542	0.802	0.407	0.694	0.309	0.587	0.254	0.346	0.530	0.682	0.943	0.946
ATT [27]	0.731	0.900	0.565	0.815	0.424	0.709	0.316	0.599	0.250	0.335	0.535	0.682	0.943	0.958
Semantic [10]	0.743	0.917	0.578	0.840	0.434	0.735	0.323	0.621	0.255	0.343	0.540	0.691	0.986	1.002
PG [11]	0.754	0.918	0.591	0.841	0.445	0.738	0.332	0.624	0.257	0.340	0.550	**0.695**	1.013	1.032
MIXER [16]	0.747	-	0.579	-	0.431	-	0.317	-	0.258	-	0.545	-	0.991	-
Ours	**0.778**	**0.929**	**0.612**	**0.855**	**0.459**	**0.745**	**0.337**	**0.625**	**0.264**	**0.344**	**0.554**	0.691	**1.102**	**1.121**

Table 2: Results from the official MS-COCO image captioning challenge leaderboard (`http://cocodataset.org/#captions-leaderboard`)

Model	Time
Semantic [10]+Attention [26]	0.10
Semantic [10]+Self-critical [17]	0.13
Semantic [10]+Self-critical [17]+Attention [26]	0.18
Ours	**0.07**

Table 3: Training time for one minibatch on COCO dataset (in seconds)



Human : A man doing a trick on this skateboard.
XE : a man riding a skateboard on a cement ledge.
Ours : a man doing a trick on a skateboard.

Human : A motorcycle carrying many wheels is parked.
XE : a motorcycle parked next to a yellow wall.
Ours : a yellow motorcycle parked in front of a street.

Human : Large black motorcycle sitting next to a white building.
XE : a motorcycle parked next to a building.
Ours : a black motorcycle parked next to a building.

Human : A small personal pizza sits in a pizza box.
XE : a pizza is sitting on a box with a box of pizza.
Ours : a person holding a box of pizza.

Human : A group of skiers in the mountains reach a sign.
XE : a group of people standing on top of slope.
Ours : a group of people skiing on a snow covered slope.

Human : Old and new trains navigating a rail yard.
XE : a train is on the tracks in a city.
Ours : a black and white photo of trains on a train yard.

Figure 3: Qualitative results of image captioning on the MS COCO dataset.

5 Conclusion

We have investigated the problem of automated image captioning by employing reinforcement learning to optimise the relevant non-differentiable language metrics such as CIDEr. A novel actor-critic based learning strategy is formulated which has the advantage over existing reinforcement learning based captioning models in that a per-token advantage and value computation is enabled leading to better model training. State-of-the-art performance is achieved using our computational efficient model on the MSCOCO benchmark.

References

[1] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. In *ICLR*, 2017.

[2] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983.

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

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[5] Jacob Devlin, Hao Cheng, Hao Fang, Saurabh Gupta, Li Deng, Xiaodong He, Geoffrey Zweig, and Margaret Mitchell. Language models for image captioning: The quirks and what works. In *ACL*, 2015.

[6] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015.

[7] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 2016.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

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

[10] Feng Liu, Tao Xiang, Timothy M Hospedales, Wankou Yang, and Changyin Sun. Semantic regularisation for recurrent image annotation. In *CVPR*, 2017.

[11] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, 2017.

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

[13] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS*, 2013.

[15] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

[16] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *ICLR*, 2016.

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

[18] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *ICLR*, 2016.

[19] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 2016.

[20] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[21] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[22] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[23] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *PAMI*, 2016.

[24] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992.

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

[26] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

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

[28] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: sequence generative adversarial nets with policy gradient. In *AAAI*, 2017.