Image Captioning with SCST-PPO and Attention

Cindy Kuang
Department of Computer Science
Stanford University
Stanford, CA
ckuang@stanford.edu

Charles Lu
Department of Computer Science
Stanford University
Stanford, CA
charleslu@stanford.edu

Abstract

We address the task of image captioning using a new policy gradient algorithm, SCST-PPO, and a deep network with an attention mechanism. SCST-PPO is a policy gradient algorithm which combines ideas from self-critical sequence training (SCST) and proximal policy optimization (PPO).

The model and SCST-PPO are implemented in PyTorch and applied to the MSCOCO image captioning task.

1 Introduction

Increased availability of datasets and continued algorithmic innovation have fostered significant progress in the field of deep learning over the last few years. Nevertheless, sequence generation remains a difficult class of tasks to train a model for. Models for image caption generation (which we will focus on in this paper) have traditionally been trained with a supervised learning method. However, this approach to image captioning makes the model vulnerable to exposure bias due to "Teacher-Forcing" [Bengio et al., 2015]. Furthermore, a supervised learning approach optimizes cross entropy loss as it is differentiable unlike metrics like CIDEr. However, though the latter is not a perfect proxy for human judgment, the former is a significantly worse metric.

Lu [2017] introduced SCST-PPO, an algorithm for training sequence generation models which combines ideas from self-critical sequence training (SCST) [Rennie et al., 2016] and proximal policy optimization (PPO) [Schulman et al., 2017]. On experiments on MSCOCO [Lin et al., 2014] image captioning task [Chen et al., 2015] with a simple model with a ResNet encoder and LSTM decoder, SCST-PPO was shown to achieve higher sample and data efficiency compared to SCST.

In this paper, we apply SCST-PPO to a more complex attention model similar to the Att2in model used in Rennie et al. [2016].

2 Background

2.1 Image Captioning

Image captioning, which involves generating a natural language description of an image, has been a key task in artificial intelligence research. Since good performance on image captioning requires an understanding of a scene, ability to "compose" attributes, objects, and relationships, and expression in natural language, the task is still very much an open problem.

Various datasets related to image captioning have been released in recent years. Most notably, the COCO (Common Objects in Context) dataset [Lin et al., 2014] includes 330,000 images, each of which is annotated with 5 reference captions. Moreover, a web server run by Microsoft provides a platform to evaluate and benchmark image captioning methods [Chen et al., 2015]. This leaderboard

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

has been dominated by deep learning approaches, which generally use a deep convolutional neural network to encode the image followed by a recurrent neural network to generate captions [Karpathy and Fei-Fei, 2015], [Vinyals et al., 2015].

Measuring the performance of an image caption generator–or just sequence generation model in general–remains a contested subject. So far, performance has generally been evaluated with metrics such as CIDEr [Vedantam et al., 2015], the BLEU score [Papineni et al., 2002], METEOR [Banerjee and Lavie, 2005], and ROUGE [Lin, 2004]. These metrics are typically non-differentiable and cannot be optimized directly with backpropagation, rendering them unsuited for supervised learning methods. It is also important to note that research has found that such evaluation metrics do not perfectly correlate with human evaluations [Callison-Burch et al., 2006], [Anderson et al., 2016].

2.2 Supervised Sequence Training

Deep learning models for sequence generation have traditionally been trained using supervised learning methods, in which a cross entropy loss is calculated for each output token and averaged across the entire generated sequence. Such models are often sequence-to-sequence recurrent models, where the model maintains an internal state h_t during generation of a sequence and outputs a single token $\hat{w}_t$ corresponding to an input token at each time step t. During training time, a method called "Teacher-Forcing" [Bengio et al., 2015] is often used, where the model is trained with cross entropy to maximize the probability of outputting a token $\hat{w}_t$ conditioned on the previous ground truth token w_{t-1} (in addition to its internal state h_t. Though training with the cross entropy loss allows the network to be fully differentiable, and thus backpropagation can be used, this creates a schism between training and test time; the model's test-time inference algorithm has no access to the previous ground truth token w_{t-1} and therefore typically feeds in the previous predicted token $\hat{w}_{t-1}$. This may lead to cascading errors during inference and is known as exposure bias. [Ranzato et al., 2015]

Various approaches have been used to address exposure bias, including scheduled sampling [Bengio et al., 2015], where the probability that the model's own predictions are fed back as input is increased over the training process, and "Professor-Forcing" [Lamb et al., 2016], an adversarial training setting where the network is encouraged to behave similarly when conditioned on ground truth inputs as when conditioned on sampled inputs.

2.3 Policy Gradient Methods

The use of policy gradient methods from reinforcement learning is an exciting development in the training of sequence generation models. This class of algorithms allows non-differentiable metrics to be directly optimized and the problem of exposure bias to be reduced [Ranzato et al., 2015]. However, policy gradient methods are themselves flawed, with issues ranging from poor sample efficiency to high variance in gradient estimates.

2.4 Self-Critical Sequence Training (SCST)

Recent breakthroughs in reinforcement learning have introduced various measures to mitigate the weaknesses with policy gradient methods. It has been found that the use of a baseline can lower the variance of gradient estimates. Self-critical sequence training (SCST) [Rennie et al., 2016] extends this idea by using the reward obtained from the model's own test-time inference algorithm as the baseline. By combining this technique with REINFORCE [Williams, 1992], SCST has achieved state-of-the-art results on the MSCOCO image captioning task.

2.5 Proximal Policy Optimization (PPO)

Separately, proximal policy optimization (PPO) [Schulman et al., 2017] is a class of policy gradient methods which are easy to implement, have good sample complexity, and are easy to tune with respect to other methods such as REINFORCE.

2.6 Self-Critical Proximal Sequence Training (SCST-PPO)

Lu [2017] introduces self-critical proximal sequence training (SCST-PPO), a policy gradient algorithm for sequence training combining ideas from SCST and PPO.

As with SCST, SCST-PPO samples the policy π_θ randomly, generating sequences $w_{1:T}$. Each sequence results in a reward R, i.e. the CIDEr score at the end of the episode. The policy π_θ is also sampled greedily to generate baseline sequences $\hat{w}_{1:T}$ to obtain the baselines $b = r(\hat{w}_{1:T})$. The advantage is computed as $\hat{A} = R - b$ using these values. For computation of the probability ratio in the PPO update, the current policy saved as $\pi_{\theta_{old}}$; in practice, the log probabilities for the entire sequences under $\pi_{\theta_{old}}$ are saved rather than copying the entire model's parameters for efficiency. A number of gradient updates are then performed with the PPO update rule, optimizing the loss $L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$. The probability ratio $r_t(\theta)$ is defined as

$$\frac{\sum_{t=1}^{T} \log \pi_\theta(w_t|I, w_{1:t-1})}{\sum_{t=1}^{T} \log \pi_{\theta_{old}}(w_t|I, w_{1:t-1})}$$

for each sampled sequence.

It is important to note that, while PPO calculates the probability ratio per action, the probability ratio is computed across the entire sequence in SCST-PPO. As with SCST, the advantage is applied to all words generated in the sequence.

Algorithm 1: Self-critical proximal sequence training

Input: start states I and reference sequences, scoring metric R, model parameterized by θ
Result: optimize θ to maximize expected reward under R
for *each epoch* **do**
  for *each batch* **do**
    $w_{1:T} \leftarrow$ sample π_θ;
    $R_t \leftarrow R(w_{1:T})$;
    $p_{old} = \sum_{t=1}^{T} \log \pi_\theta(w_t|I, w_{1:t-1})$;
    $\hat{w}_{1:T} \leftarrow$ sample greedily π_θ;
    $b_t \leftarrow R(\hat{w}_{1:T})$;
    $\hat{A} \leftarrow R_t - b_t$;
    $\theta_{old} \leftarrow \theta$;
    for *each PPO iteration* **do**
      $p = \sum_{t=1}^{T} \log \pi_\theta(w_t|I, w_{1:t-1})$;
      $r(\theta) = \frac{p}{p_{old}}$;
      optimize PPO loss $L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$;

3 Experiments

We implement SCST-PPO and apply it to two models, one with attention (Att2in) and one without (FC), for the MSCOCO image captioning task [Kuang and Lu, 2018]. The implementation is based on Ruotian Luo's implementation of SCST [Luo, 2017], and uses PyTorch 0.2.0 [PyT, 2017] in Python 2.7. We train the models using a virtual machine with two NVIDIA Tesla P100 GPUs.

For both models, we first pretrain them in the supervised fashion with cross entropy loss. Then, starting from this point, we train each model with SCST-PPO (with either 4 or 8 PPO iterations) or vanilla SCST.

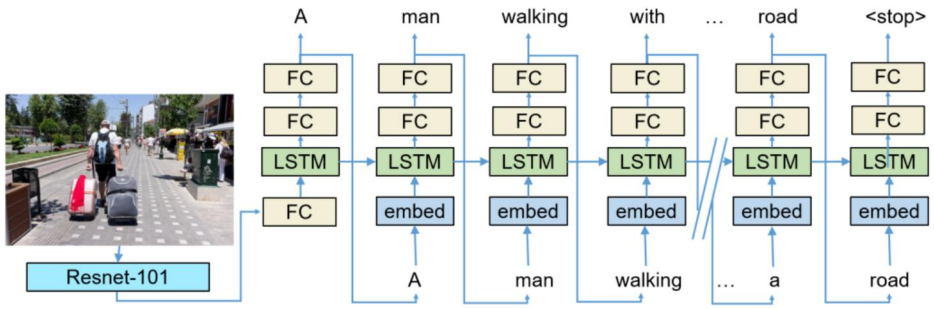


Figure 1: Diagram of the FC model used in the original SCST paper as well as in our experiments. The Att2in model is more complex and uses attention over the spatial features of the image at each time step.

4 Results

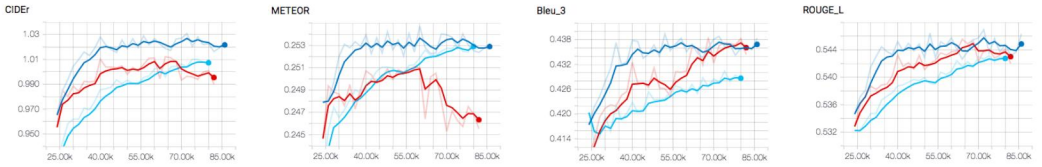


Figure 2: Basic FC without attention, evaluated on CIDEr, METEOR, BLEU-3, and ROUGE-L. SCST graphed in light blue; SCST-PPO with 4 PPO iterations graphed in dark blue, 8 PPO iterations graphed in red.

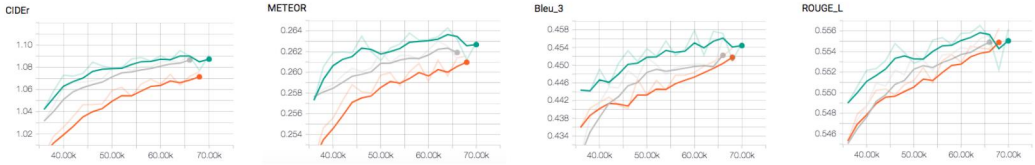


Figure 3: Att2in, evaluated on CIDEr, METEOR, BLEU-3, and ROUGE-L. SCST graphed in orange; SCST-PPO: 4 PPO iterations graphed in teal, 8 PPO iterations graphed in gray.

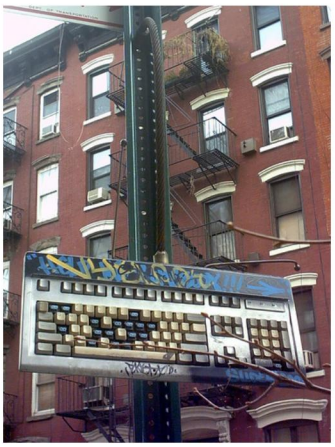
Table 1: Model performance evaluated on CIDEr, METEOR, BLEU-3, and ROUGE-L

Model	CIDEr	METEOR	BLEU-3	ROUGE-L
FC				
Baseline, XE 25e	0.976	0.251	0.429	0.534
+ SCST 35e	1.039	0.256	0.435	0.548
+ SCST-PPO (4 iter.) 35e	1.041	0.255	0.441	0.547
+ SCST-PPO (8 iter.) 35e	1.023	0.252	0.438	0.546
Att2in				
Baseline, XE 20e	1.043	0.260	0.442	0.546
+ SCST 20e	1.095	0.263	0.457	0.558
+ SCST-PPO (4 iter.) 20e	1.112	0.266	0.462	0.558
+ SCST-PPO (8 iter.) 20e	1.111	0.264	0.462	0.557

In both the FC and Att2in models, fine-tuning with SCST-PPO significantly improved model performance and demonstrated significantly better sample complexity. Although SCST-PPO required noticeably more computation time per iteration than SCST, due to the more complex objective function and multiple gradient updates per iteration, we observe better sample efficiency when using SCST-PPO. Furthermore, the model did noticeably better when performing 4 PPO iterations per image, as opposed to 8 iterations, indicating the number of PPO iterations is an important hyperparameter to tune.

5 Analysis

FC, XE: *a bed in a green field with a tree*
FC, SCST: *a bedroom with a bed and in the grass*
FC, SCST-PPO4: *a bed with a blanket and in the of it*
Att2in, XE: *a bed sitting in the middle of a forest*
Att2in, SCST: *a bedroom with a bed and in the middle of*
Att2in, SCST-PPO4: *a bed with a blanket and in the middle of*



FC, XE: *a blue and white clock on a brick building*
FC, SCST: *a clock on the side of a building*
FC, SCST-PPO4: *a clock on the side of a building*
Att2in, XE: *a keyboard with a keyboard on it on a city street*
Att2in, SCST: *a computer keyboard sitting on top of a building*
Att2in, SCST-PPO4: *a computer keyboard sitting on the side of a building*

FC, XE: *a motorcycle parked in front of a building*
FC, SCST: *a motorcycle parked in front of a building*
FC, SCST-PPO4: *a motorcycle is parked on display in a building"*
Att2in, XE: *a bike parked in front of a building*
Att2in, SCST: *a bike is parked in front of a building*
Att2in, SCST-PPO4: *a bike is parked on the side of a building*

Figure 4: Objects from the OOOC dataset where the FC model trained with SCST-PPO generated far better captions than both the models trained with cross entropy only and SCST. The above examples were chosen by finding instances where SCST-PPO was better than XE, without looking at SCST.

We evaluate our models on three images from the SUN 09 out-of-context (OOOC) dataset [Choi et al., 2012], which features images containing objects composed in a way that would not normally be seen in training data. Because image captioning models tend to learn statistical correlations between objects in similar context from training data, it is a difficult task to generate reasonable captions for the strange images from OOOC.

Though SCST-PPO and SCST generated captions of similar quality in most instances, there were certain images where the caption generated by SCST-PPO was significantly better. Furthermore, the Att2in model generally generates better captions than the FC model.

While Rennie et al. [2016] observe that the performances of all captioning models on MSCOCO evaluation data are qualitatively similar, they note that their SCST-trained models exhibit significantly better performance on images with objects out of context. Furthermore, their SCST-trained attention models create captions which are even more accurate. Similarly, we observe in our evaluation set that

our Att2in model trained with SCST-PPO generally produces better captions than the SCST-trained model. However, a more rigorous analysis is certainly necessary.

Contributions

Both team members, Cindy Kuang and Charles Lu, contributed equally. Work included implementation through pair coding, clarifying previous mistakes in the SCST-PPO algorithm, setting up the development and training environments and preprocessing data, and analysis.

References

Pytorch, 2017. URL `http://pytorch.org/`.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72, 2005.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179, 2015.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. In *EACL*, volume 6, pages 249–256, 2006.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Myung Jin Choi, Antonio Torralba, and Alan S Willsky. Context models and out-of-context objects. *Pattern Recognition Letters*, 33(7):853–862, 2012.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

Cindy Kuang and Charles Lu. Pytorch implementation for "self-critical proximal sequence training" for image captioning. `https://github.com/clu8/self-critical-ppo`, 2018.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

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 *European conference on computer vision*, pages 740–755. Springer, 2014.

Charles Lu. Self-critical sequence training. *CS 332*, pages 1–9, 2017.

Ruotian Luo. Unofficial pytorch implementation for self-critical sequence training for image captioning. `https://github.com/ruotianluo/self-critical.pytorch`, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.