Improving Image Captioning with Conditional Generative Adversarial Nets

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

Main | Overview

1 Introduction

2 RL-based image captioning

3 Proposed Algorithm

4 Experiments

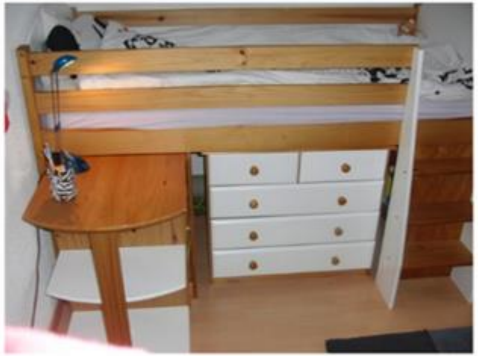
5 Conclusion

Part I

Introduction

Introduction | Image captioning

A horse carrying a large load of hay and two people sitting on it.



Bunk bed with a narrow shelf sitting underneath it.

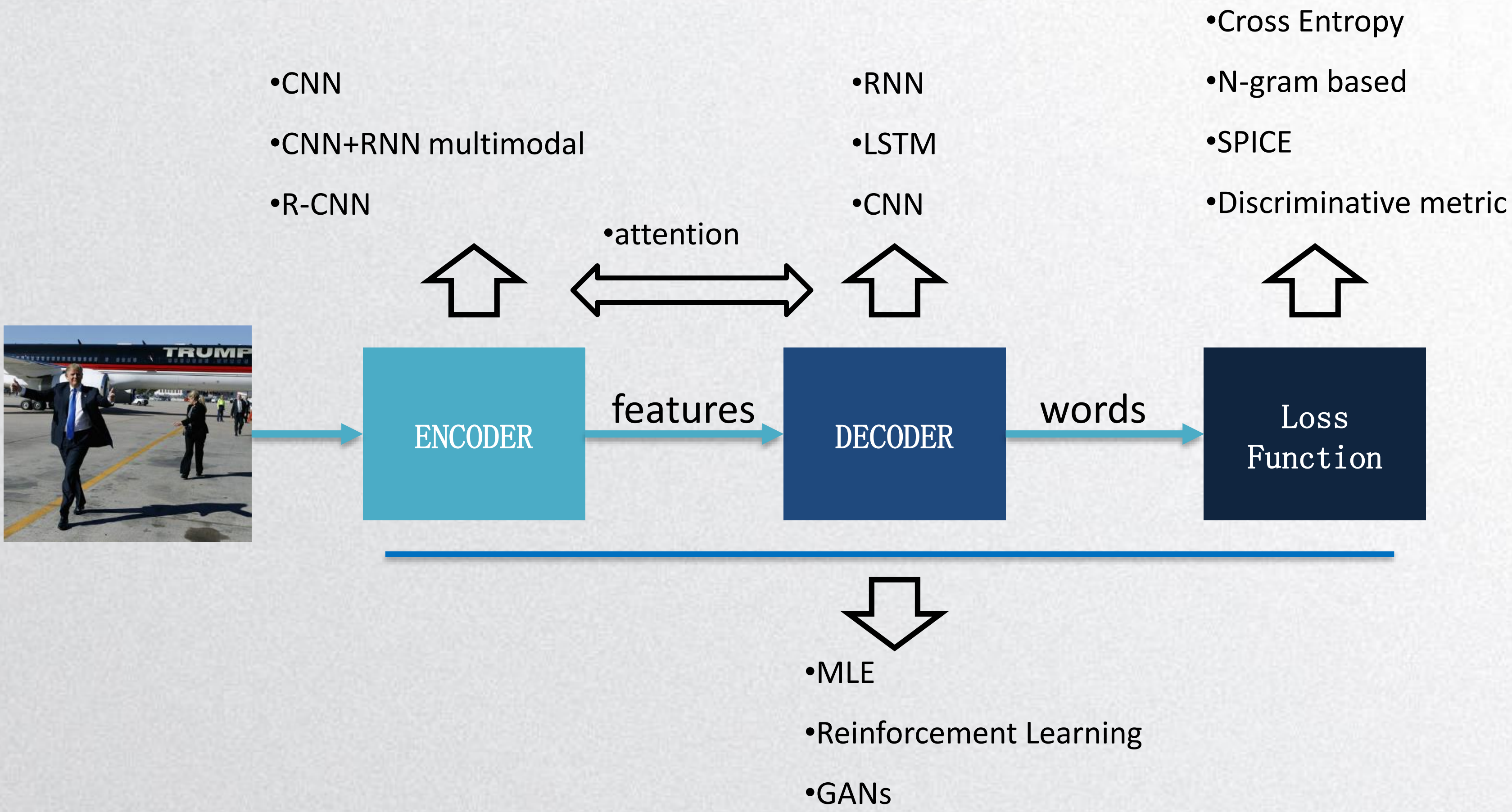
The man at bat readies to swing at the pitch while the umpire looks on.

A female tennis player in action on the court.

A group of young men playing a game of soccer

A man riding a wave on top of a surfboard.

Introduction | Basic framework



Part II

RL-based image captioning

RL-based image captioning | Why use reinforcement learning?

1: exposure bias

2: do not optimize the whole sequence

BLEU-1, BLEU-2, ...

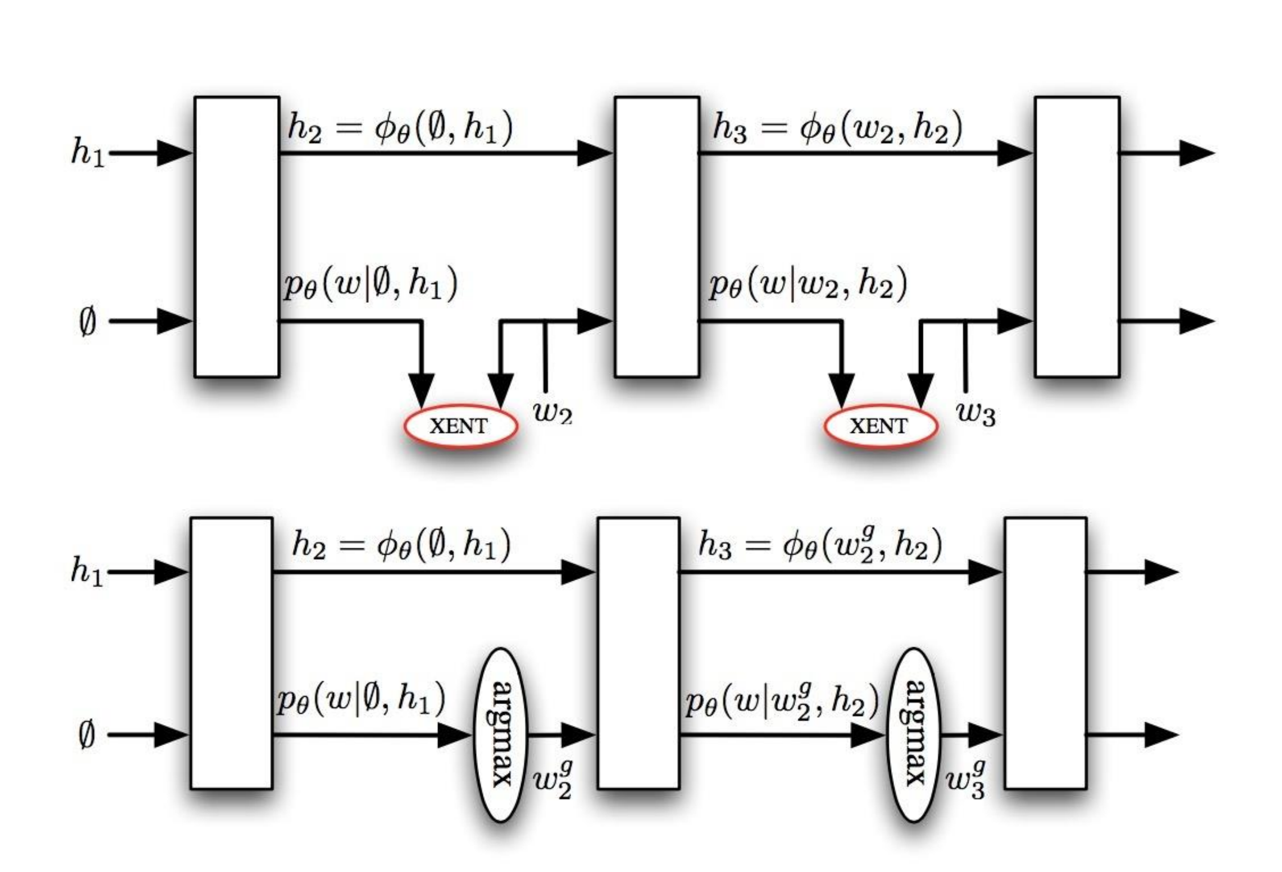
ROUGE

METEOR

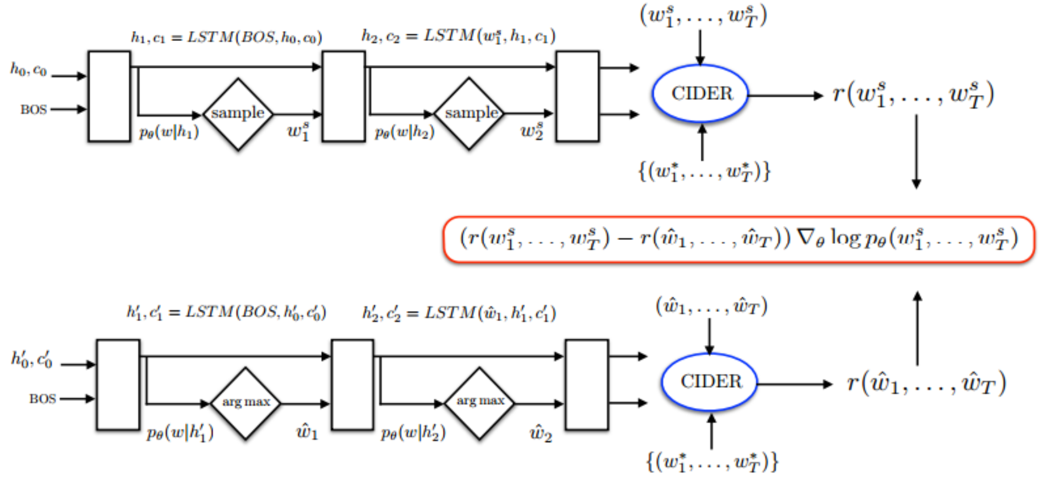
CIDER

SPICE

...



RL-based image captioning | Self-critical sequence training



CE loss

$$\theta^* = \arg\max_\theta \pi_\theta(\omega_1, \omega_2, \cdots \omega_T)$$
$$= \arg\max_\theta \log \pi_\theta(\omega_1, \omega_2, \cdots \omega_T)$$
$$= \arg\max_\theta \sum_{i=1}^{T} \log \pi_\theta(\omega_i \mid \omega_1, \cdots \omega_{i-1})$$
$$\text{loss} = -\sum_{i=1}^{T} \log \pi_\theta(\omega_i \mid \omega_1, \cdots \omega_{i-1})$$

RL loss

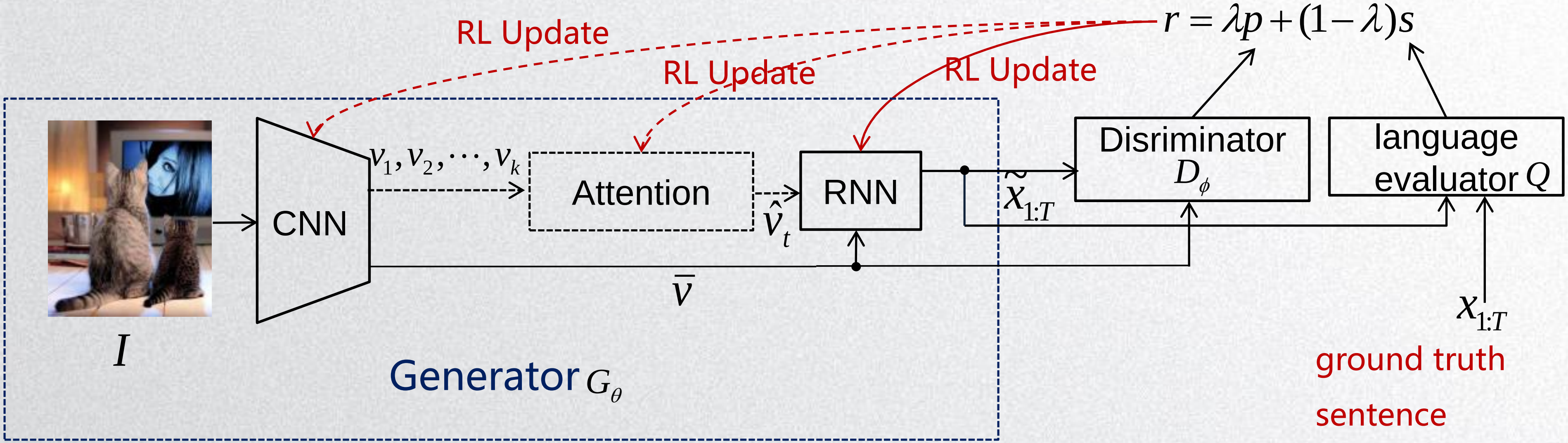
$$\theta^* = \arg\max_\theta \left(r(\omega_1^s, \omega_2^s, \cdots, \omega_T^s) - r(\hat{\omega}_1, \hat{\omega}_2, \cdots, \hat{\omega}_T) \right) \pi_\theta(\omega_1^s, \omega_2^s, \cdots, \omega_T^s)$$
$$= \arg\max_\theta \left(r(\omega_1^s, \omega_2^s, \cdots, \omega_T^s) - r(\hat{\omega}_1, \hat{\omega}_2, \cdots, \hat{\omega}_T) \right) \log \pi_\theta(\omega_1^s, \omega_2^s, \cdots, \omega_T^s)$$
$$= \arg\max_\theta \left(r(\omega_1^s, \omega_2^s, \cdots, \omega_T^s) - r(\hat{\omega}_1, \hat{\omega}_2, \cdots, \hat{\omega}_T) \right) \sum_{i=1}^{T} \log \pi_\theta(\omega_i^s \mid \omega_1^s, \cdots, \omega_{i-1}^s)$$
$$\text{loss} = -\left(r(\omega_1^s, \omega_2^s, \cdots, \omega_T^s) - r(\hat{\omega}_1, \hat{\omega}_2, \cdots, \hat{\omega}_T) \right) \sum_{i=1}^{T} \log \pi_\theta(\omega_i \mid \omega_1, \cdots \omega_{i-1})$$

Rennie, Steven J., et al. "Self-critical sequence training for image captioning." arXiv preprint arXiv:1612.00563 (2016).

Part III

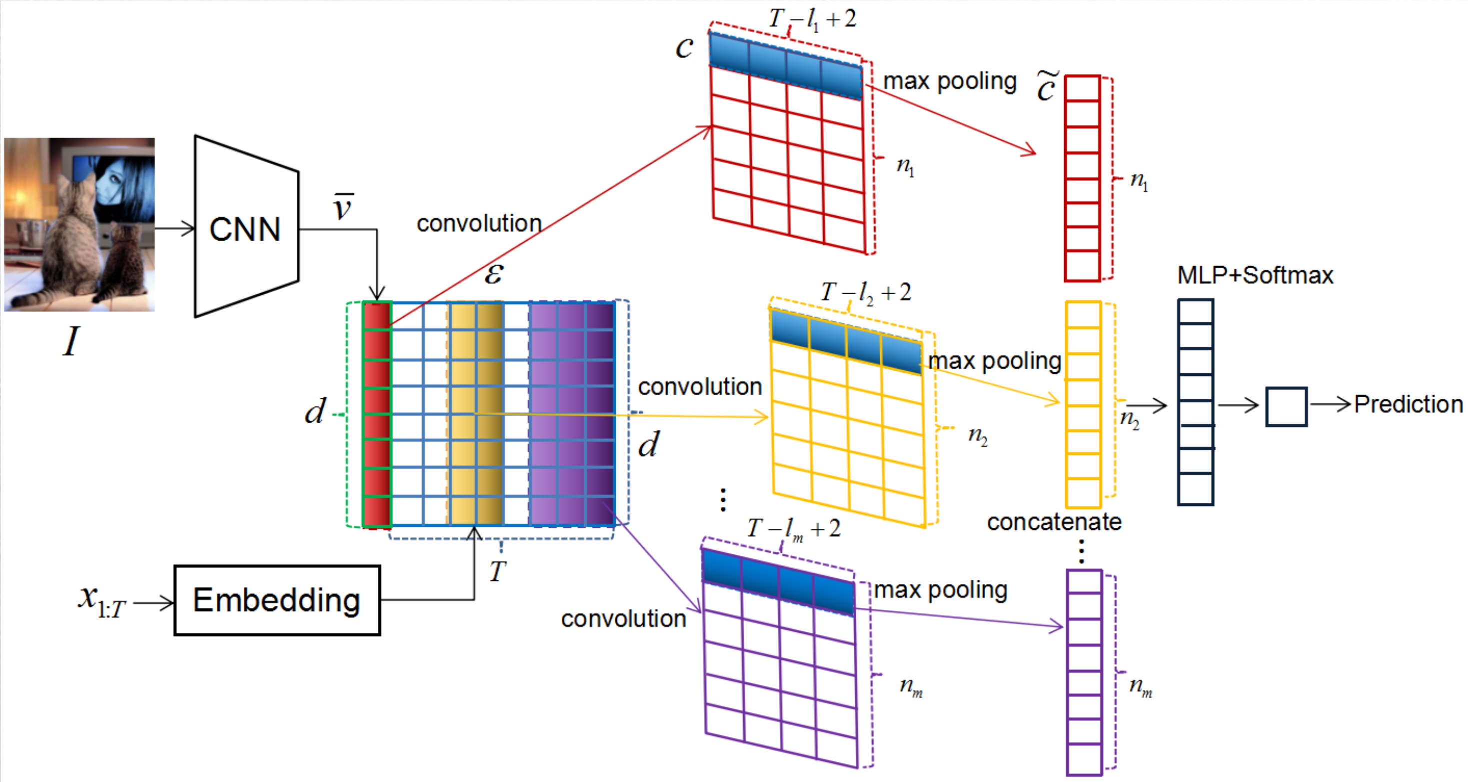
Proposed Algorithm

Proposed algorithm | Overall framework



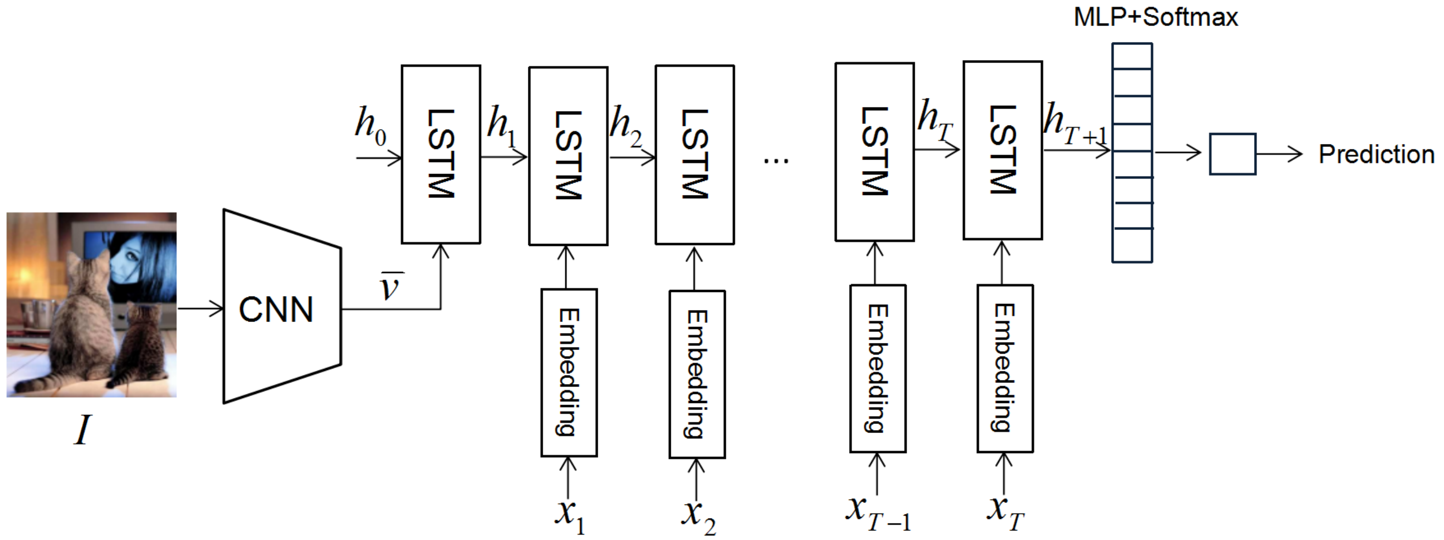
$$r(\tilde{\boldsymbol{x}}|\boldsymbol{I},\boldsymbol{x}) = \lambda \cdot p + (1-\lambda)\cdot s = \lambda \cdot D_\phi(\tilde{\boldsymbol{x}}|\boldsymbol{I}) + (1-\lambda)\cdot Q(\tilde{\boldsymbol{x}}|\boldsymbol{x}),$$

Proposed algorithm | CNN-based discriminator



$$\varepsilon = \bar{v} \oplus \boldsymbol{E} \cdot \boldsymbol{x}_1 \oplus \boldsymbol{E} \cdot \boldsymbol{x}_2 \oplus \cdots \oplus \boldsymbol{E} \cdot \boldsymbol{x}_T.$$

Proposed algorithm | RNN-based discriminator



$$\boldsymbol{h}_{t+1} = \begin{cases} \mathrm{LSTM}(\bar{\boldsymbol{v}}, \boldsymbol{h}_t) & t = 0 \\ \mathrm{LSTM}(\boldsymbol{E} \cdot \boldsymbol{x}_t, \boldsymbol{h}_t) & t = 1, 2, \cdots, T \end{cases}$$

$$p = \sigma(\boldsymbol{W}_R \cdot \boldsymbol{h}_{T+1} + \boldsymbol{b}_R),$$

Proposed algorithm | Training pipeline

Algorithm 1 Image Captioning Via Generative Adversarial Training Method

Require: caption generator $G_{\boldsymbol{\theta}}$; discriminator D_{ϕ}; language evaluator Q, e.g. CIDEr-D; training set $\mathbb{S}_r = \{(\boldsymbol{I}, \boldsymbol{x}_{1:T})\}$ and $\mathbb{S}_w = \{(\boldsymbol{I}, \hat{\boldsymbol{x}}_{1:T})\}$.

Ensure: optimal parameters $\boldsymbol{\theta}$, ϕ.

```
Initial G_θ and D_φ randomly.
Pre-train G_θ on S_r by MLE.
Generate some fake samples based on G_θ to form S_f = {(I, x̃_{1:T})}.
Pre-train D_φ on S_r ∪ S_f ∪ S_w by Eq. (12).
repeat
    for g-steps=1 : g do
        Generate a mini-batch of image-sentence pairs {(I, x̃_{1:T})} by G_θ.
        Calculate p based on Eqs. (7)-(9) or Eqs. (10)-(11).
        Calculate s based on Q.
        Calculate reward r according to Eq. (6).
        Update generator parameters θ by SCST method via Eq. (5).
    end for
    for d-steps=1 : d do
        Generate negative image-sentence pairs {(I, x̃_{1:T})} by G_θ, together with negative samples {(I, x̂_{1:T})} ⊆ S_w and positive samples {(I, x_{1:T})} ⊆ S_r.
        Update discriminator parameters φ via Eq. (12).
    end for
until generator and discriminator converge
```

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

Part IV

Experiments

Experiments | Parameters determination

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

Table 3: Metric selection

fixed parameters: g=1; d=1; λ=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: λ=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

Experiments | Comparisons

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 (%).

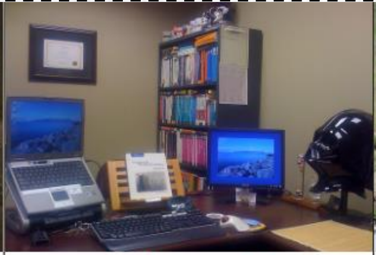
Generator	Discriminator	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE	CNN-D	RNN-D
resnet101+att2in (Rennie et al. 2017)	none*	-	-	-	33.3	26.3	55.3	111.4	-	-	-
	CNN-GAN	78.1	61.3	46.4	34.4	26.6	56.1	112.3	20.3	47.9	45.8
	RNN-GAN	78.0	61.4	46.3	34.3	26.5	56.0	112.2	20.4	46.0	48.1
	ensemble	**78.5**	**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**

Experiments | MSCOCO online test serve

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	-	-

Experiments | Examples

				
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

Part V

Conclusion

Conclusion

Conclusion

1 Generic

A generic framework to improve any RL-based image caption algorithm.

2 Effective

Consistently improve all automatic evaluation metrics, e.g. BLEU, ROUGE, METEOR, CIDER, SPICE

3 Evaluator

Well trained CNN-based and RNN-based discriminators can also be viewed as evaluators for image captioning.

4 Ensemble

Ensemble of CNN-GAN and RNN-GAN model can further improve performance

Q&A

Thanks