IMAGE CAPTIONING WITH TWO CASCADED AGENTS

*Lun Huang, Wenmin Wang *, Gang Wang*

SECE, Shenzhen Graduate School, Peking University, Shenzhen, China
{*huanglun@pku.edu.cn*}, {*wangwm@ece.pku.edu.cn*}, {*wg_alan@pku.edu.cn*}

ABSTRACT

Recent neural models on image captioning usually take a encoder-decoder fashion, where the decoder predicts a single word at one step recently with the encoder providing information. The encoder is a pretrained CNN model typically. Thus the decoder, the input to it, and the output from it become the most important parts of a model. We propose a pipelined image captioning framework consisting of two cascaded agents. The former is named as "semantic adaptive agent" which generates the input to the decoder by consulting the information from the current decoding process, and the latter as "caption generating agent" which select a single word of the vocabulary as the output of the decoder by taking consideration of the input and the current states of the decoder. For the framework of two cascaded agents, we design a multi-stage training procedure to train the two agents with different objectives by fully utilizing reinforcement learning. In experiments, we conduct quantitative and qualitative analysis on MS COCO dataset and our results can significantly outperform baseline methods in terms of several evaluation metrics.

Index Terms— Image captioning, Attention, Deep learning, Reinforcement learning

1. INTRODUCTION

Image captioning, which aims to describe an image using a complete and natural sentence, is a primary goal of image understanding. It's a challenging task, since not only dose it require to understand salient entities in an image, the attributes of them and connections among them, but also require to verbalize with natural language [1, 2, 3, 4, 5, 6].

Inspired by the great development of deep learning and neural machine translation, the use of attention mechanisms on deep encoder-decoder paradigm[7] has yielded impressive results on the task, becoming the mainstream. Methods based on attention mechanisms force the decoder to attend visual image features at every decoding step, which is unnecessary and can be misleading. In [8], Lu et al. appended a "visual sentinel", which is another hidden state of the decoder, to the image feature vectors. And further a sentinel gate is designed to mix the image features and the visual sentinel then input the mixture to the decoder when generating the next word. However the practice of mixing the two kinds of information makes it hard to distinguish whether it's "visual" or "non-visual" and can bring noise to each other. Except that, the value of the sentinel gate can't actually stand for the importance of each one quantitatively since they are not guaranteed to have similar magnitudes.

Models that take a word-level training can involve two problems. The first one is called "exposure bias", and the second problem is about the inconsistency [9]. Recently, it has been shown that the reinforcement learning (RL)[10] can provide a solution to these two issues above[9, 11].

Combining all these different branches of works above, we propose a pipelined framework consisting of two cascaded agents of reinforcement learning for image captioning. In our framework, the first agent, named as "semantic adaptive agent", forms the input to the decoder by consulting the information from the current decoding process. And the second agent, named as "caption generating agent", selects a single word of the vocabulary as the output of the decoder by taking consideration of the input and the current states of the decoder. For training our cascaded captioning model, we design a multi-stage training procedure with different objectives by fully utilizing the policy gradient methods in reinforcement learning.

Main contributions of this work are summarized as follows: **1)** we propose a framework of two cascaded agent for image captioning; **2)** we build a pipelined training mechanism for our cascaded agents with reinforcement learning. Our method can achieve promising improvement of performance on MS COCO dataset.

2. FRAMEWORK OF TWO CASCADED AGENTS

Our model is based on the general encoder-decoder framework for image captioning. Image is first encoded through a CNN, then decoded to a sequence of words recurrently. The decoder in our model consists of two agents, "semantic adaptive agent" notated as A_1 and "caption generation agent" no-

This project was supported by Shenzhen Peacock Plan (20130408-183003656), Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (ZDSYS201703031405467), and National Natural Science Foundation of China (NSFC, No.U1613209).

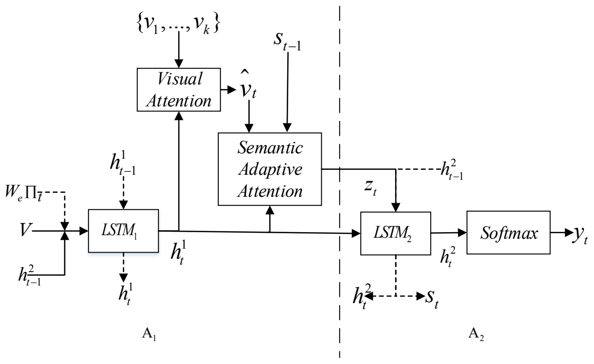

Fig. 1. Overview of the proposed captioning model of two cascaded agents. At each decoding step, A_1 is responsible for generating z_t, the "semantic adaptive vector", as the input to A_2. And A_2 is responsible for predicting a single word to generate a caption.

tated as A_2, as shown in Figure 1.

We now describe the formulation of the two agents.

2.1. Semantic Adaptive Agent

The semantic adaptive agent, A_1, is responsible for generating a "semantic adaptive vector" z_t at each decoding time step according to the current decoding states. z_t is either the attended image feature vector v_t or the visual sentinel s_{t-1} and it's input to A_2 .

An LSTM layer ($LSTM_1$) is included in A_1 to guide the generation of z_t. The input to $LSTM_1$ is a vector that concatenats the mean-pool vector $\bar{v}$ of the image feature set $V = \{v_1, ..., v_k\}$, an encoding $W_e\Pi_t$ of the previously generated word as well as the previous output h^2_{t-1} of $LSTM_2$ (the LSTM layer in A_2), given by:

$$x^1_t = [h^2_{t-1}, \bar{v}, W_e\Pi_t] \tag{1}$$

2.1.1. Semantic Adaptive Attention

We design a "semantic adaptive attention" mechanism to generate z_t. The policy network of deciding the assignment of z_t is comprised of two leaner layers with a $\tanh$ and a softmax activation function respectively:

$$\beta_t = \text{softmax}\left(W_b \tanh\left(W_{hb} h^1_t\right)\right) \tag{2}$$

where h^1_t is the output of $LSTM_1$ at time step t, $\beta_t \in \mathbb{R}^2$, $\beta_t[0]$ and $\beta_t[1]$ stand for the probabilities of sampling $\hat{v}_t$ or s_{t-1} respectively.

Note that unlike the original implementation of adaptive attention [8], the "hard" fashion is adopted, which indicates that the decision is explicit: rather than produces a mixture of the weighted image features and the visual sentinel, the agent selects one of them.

2.1.2. Semantic Adaptive Vector

If image features are chosen to be attended, then we will let $z_t = \hat{v}_t$, where $\hat{v}_t$ is the weighted average vector over the whole image feature set V with the normalized attention weights α_t: $\hat{v}_t = \sum_{i=1}^{K} \alpha_{i,t} v_i$. The weight $\alpha_{i,t}$ for each of the k image features v_i is computated as follows:

$$a_{i,t} = w_a^T \tanh\left(W_{va} v_i + W_{ha} h^1_t\right) \tag{3}$$

$$\alpha_t = \text{softmax}\left(a_t\right) \tag{4}$$

Otherwise, if it's decided not to attend image features, then s_{t-1} will be assigned to z_t. The visual sentinel s is another hidden state of $LSTM_2$, and it's supposed to store some necessary information. From s, a word can be inferred without attending to the visual image. It's given by:

$$g_t = \sigma(W_{xg} x^2_t + W_{hg} h^2_{t-1}) \tag{5}$$

$$s_t = g_t \odot \tanh(c^2_t) \tag{6}$$

where x^2_t is the input to $LSTM_2$ at time step t, h^2_{t-1} is previous output, and g_t is the gate applied on the memory cell c^2_t, $\odot$ represents the element-wise product and σ is the logistic sigmoid activation.

2.2. Caption Generation Agent

As mentioned above, A_2 also includes an LSTM core ($LSTM_2$). The input to $LSTM_2$ consists the out of $LSTM_1$ and the semantic adaptive vector $\hat{z}_t$, given by:

$$x^2_t = [\hat{z}_t, h^1_t] \tag{7}$$

The output h^2_t is used to predict the conditional distribution over possible output words of the vocabulary:

$$p(y_t \mid y_{1:t-1}) = \text{softmax}\left(W_p h^2_t + b_p\right) \tag{8}$$

The notation $y_{1:T}$ refers to a sequence of words $(y_1, ..., y_T)$.

3. TRAINING PROCEDURE AND OBJECTIVES

Training with Cross Entropy Loss. The typical way of training a captioning model is to optimize cross entropy loss L_{XE}. Given the sequence $y^*_{1:T}$ of a target ground truth and the parameters θ of the captioning model, the loss can be expressed as:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y^*_t \mid y^*_{1:t-1})) \tag{9}$$

However, in our case, it's hard for L_{XE} to be directly optimized because there's a sampling operation on z_t. Thus, we adopt the REINFORCE rule [12, 13] to approximate the gradient with the following loss:

$$L_{Stage1} = L_1 + \lambda_e L_2 + \lambda_h L_h \tag{10}$$

where L_1 is the cross entropy loss when z_t is given:

$$L_1(\theta) = -\sum_{t=1}^{T} \log(p_\theta(y_t^* \mid y_{1:t-1}^*, \tilde{z_t})) \tag{11}$$

L_2 is the loss for the "semantic adaptive attention" of A_1:

$$\frac{\partial L_2}{\partial \theta} = -\sum_{t=1}^{T} (\log(p_\theta(y_{1:T}^* \mid \tilde{z_{1:T}})) - b) \frac{\partial \log p(\tilde{z_t})}{\partial \theta} \tag{12}$$

$L_h = -H[\beta]$ is an entropy term on the multinouilli distribution on β. And λ_e and λ_h are two hyper-parameters.

b is a baseline used to reduce variance, and we let b to be a moving average of $-L_1$:

$$b_k = 0.9 \times b_{k-1} + 0.1 \times \log p(y_{1:T}^* \mid \tilde{z_{1:T}}) \tag{13}$$

SCST: Traing Together. Following the approach described as Self-Critical Sequence Training [11] (SCST), we can directly optimize the NLP metrics which are used at test time. The loss can be approximated as the negtive expected score:

$$L_{Stage2} = -\mathbf{E}_{y_{1:T} \sim p_\theta, z_{1:T} \sim p_\theta}[r(y\text{-}z_{1:T})] \tag{14}$$

the notation y-z is the output word of A_2 when the output of A_1 is z, and r is the score function (e.g., CIDEr [14]). The gradient of this loss can be approximated:

$$\nabla_\theta L_{Stage2}(\theta) \approx -(r(y^s\text{-}{z^s}_{1:T}) - r(\hat{y}\text{-}\hat{z}_{1:T})) \nabla_\theta \log p_\theta(y_{1:T}^s, z_{1:T}^s) \tag{15}$$

the notation y^s or z^s indicates that it's a result sampled from probabilities, while $\hat{y}$ or $\hat{z}$ means that it's a result sampled from greedy decoding (sampling items with maximum probabilities).

SCST: Traing Alternatively. Note that y^s-${m^s}_{1:T}$ above can be regarded as the sampled result from the joint decisions of the two agents A_1 and A_2. However, if one of the agents(e.g. A_1) takes a bad action at some step which can ruin the caption generation, then no matter which action A_2 takes could not make the final result any better for the following decoding process.

Hence we propose another 'alternative training' stage: first we train A_2 by fixing the policy of A_1 and perform its actions with greedy decoding, and A_2 sample its actions from probabilities. Then in turn we keep the policy of A_2 fixed and train A_1.

When training A_1, the gradient can be approximated:

$$\nabla_\theta L_{Stage3}^1(\theta) \approx -(r(\hat{y}\text{-}{m^s}_{1:T}) - r(\hat{y}\text{-}\hat{m}_{1:T})) \nabla_\theta \log p_\theta(m_{1:T}^s \mid \hat{y}_{0:T-1}) \tag{16}$$

When training A_2, the gradient can be approximated:

$$\nabla_\theta L_{Stage3}^2(\theta) \approx -(r(y^s\text{-}\hat{m}_{1:T}) - r(\hat{y}\text{-}\hat{m}_{1:T})) \nabla_\theta \log p_\theta(y_{1:T}^s \mid \hat{m}_{1:T}) \tag{17}$$

The objective at this stage is then:

$$L_{Stage3}(\theta) = \lambda_1 L_{Stage3}^1(\theta) + \lambda_2 L_{Stage3}^2(\theta) \tag{18}$$

where if $\lambda_1 = 1, \lambda_2 = 0$, and A_1 will be trained; if $\lambda_1 = 0, \lambda_2 = 1$, and A_2 will be trained. And if we want to train both A_1 and A_2, then we can let $\lambda_1 = 1$ and $\lambda_2 = 1$.

In the sections that follow, we will refer to the three different training stages using the notations: S1, S2 and S3.

4. EXPERIMENTS

4.1. Dadasets

We evaluate our proposed method on the popular MS COCO dataset [15]. The "Karpathy" data split [16] is used for the performance comparisions, where 5,000 images are used for validation, 5,000 images for testing and the rest for training. We convert all sentences to lower case, and drop the words that occur less than 6 times and end up with a vocabulary of 9,487 words. We use different metrics, including BLEU [17], METEOR [18], ROUGE-L [19], CIDEr [14] and SPICE [20], to evaluate the proposed method and compare with other methods. All the metrics are computed with the publicly released code[1].

4.2. Implementation Details

Like in [21], we take Faster-RCNN [22] as our encoder and extract the bottom-up image features with it. We set the size of LSTM cell to 1,000, and the size of the input word embedding to 1,000. As for training process, training stage 1 (S1) takes 20 epochs, and ADAM [23] optimizer is used with a learning rate initialized with 5e-4 and annealed by 0.8 every 3 epochs. We increase the probability of feeding back a sample of the word posterior by 0.05 every 5 epochs [24]. We set the hyper-parameter $\lambda_e = 1$, and $\lambda_h = 0.02$. S2 takes 10 epochs and S3 takes another 10 epochs where ADAM optimizer is used with a fixed learning rate of 5e-5.

4.3. Quantitative Analysis

We report the performance on the MSCOCO Karpathy test split of our model as well as the compared models in Table 1. The compared models inludes: Att2all[11], which

[1] https://github.com/tylin/coco-caption

Table 1. Performance of different image captioning models on the MS COCO ‘Karpathy’ test split. The highest value of each entry has been highlighted in boldface. B@n is short for BLEU-n, M is short for METEOR, R for ROUGE-L, C for CIDEr and S is short for SPICE. Σ indicates an ensemble.

Model	B@1	B@4	M	R	C	S
Att2all [11]	-	34.2	26.7	55.7	114.0	-
Up-Down [21]	79.8	36.3	27.7	56.9	120.1	21.4
Att2all$^{\Sigma}$ [11]	-	35.4	27.1	56.6	117.5	-
Soft Ada (Ours)	79.4	36.6	27.9	57.6	121.5	21.3
CA (Ours)	79.8	37.2	28.2	57.9	125.7	**21.7**
CA$^{\Sigma}$ (Ours)	**80.6**	**38.2**	**28.3**	**58.4**	**126.4**	**21.7**

Table 2. Performance of our model at different stages.

Stage	B@1	B@4	M	R	C	S
S1	75.5	35.5	27.4	56.0	110.9	20.0
S1+S2	78.8	36.3	27.7	57.3	120.7	21.0
S1+S2+S3	**79.8**	**37.2**	**28.2**	**57.9**	**125.7**	**21.7**

employs a modified visual attention; Up-Down, a two-LSTM layer model with bottom-up and top-down attention; and “soft ada”, a model designed by us with a “soft semantic adaptive attention” added in the Up-Down model, where “soft” means the “semantic adaptive vector” is the mixture of the image feature vector and visual sentinel.

For fair comparision, all the models are first trained under XE loss and then trained with REINFORCE. It can be seen from Table 1 that both our single and ensembled model of two cascaded agents(CA) can achieve the best performance in tems of all metrics. Comparing to the Up-Down model, our single cascaded-agents model improves the performance by a large margin across most metrics: BLEU-4, METEOR, ROUGE-L, CIDEr and SPICE. And the ensemble of our model achieves further improvement.

Importance of Two Cascaded Agents. Our proposed baseline “soft ada” performs slightly better than the Up-Down baseline for some metrics: BLEU-4, METEOR, ROUGE-L and CIDEr; but it performs slightly worse for the other metrics: BLEU-1, SPICE. It’s not obvious whether “soft ada” is better than Up-Down. However, the improvement achieved by our cascaded agents(CA) is significant, which shows that the framework of two cascaded agents is important.

Importance of Multi-stage Traing. We report the performances of our cascaded agents(CA) model at different training stages in Table 2. As can be seen, both S2 and S3 boost the performance by a large margin.

4.4. Qualitative Analysis

To qualitatively show the caption generating process, we begin by visualizing the actions of the two agents and the attended image regions.

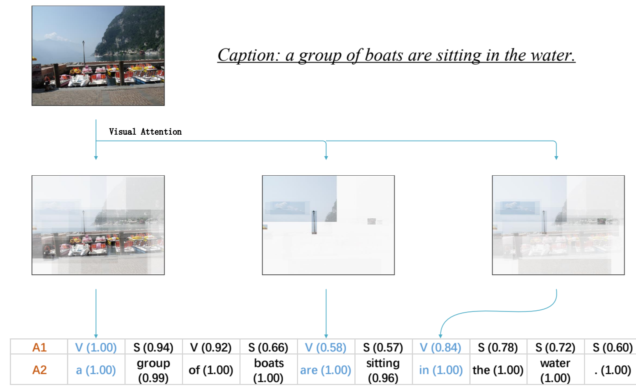

Fig. 2. An examples of generated caption. The action of A_1 is ‘V’ or ‘S’, standing for visual image or sentinel respectively; the action of A_2 is its prediction of a word. The value following the action stands for the confidence.

We note that first, the “semantic adaptive agent” decides to attend the image features for only a few times. A caption generation process can be divided to several phases. For each phase, at the beginning some image features are attended and then the decoder gets some knowledge of the attended features, then for the rest of phase, the “semantic adaptive agent” always decides to not attend the image features. For the example in Figure 2, the caption generation process can be divided to the following phases: generating “a group of boats”, “are sitting” and “in the water”. Secondly, we observe that both the two agents are very confident about their decisions: most of the values of confidence are equivalent or close to 1.

5. CONCLUSION

In this paper, we propose an image captioning model which consists of two cascaded agents. For the training procedure, we design a multi-stage incremental training method to guarantee that two cascaded agents can collaborate well to converge on a good policy. In experiments, we verify the remarkable validity of our model on MS COCO dataset. And through quantitative analysis and qualitative analysis, it has been shown that the performance of our framework is promising. In the future, we will explore more about the potential of the pipelined model of two cascaded agents, and consider designing better reward function. Apart from this, we are conducting some experiments on applying trust region sequence-level optimization on image captioning to achieve a better learning ability.

6. REFERENCES

[1] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L

Berg, “Babytalk: Understanding and generating simple image descriptions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2891–2903, 2013.

[2] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos, “Corpus-guided sentence generation of natural images,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 444–454.

[3] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III, “Midge: Generating image descriptions from computer vision detections,” in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 747–756.

[4] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1473–1482.

[5] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei, “Exploring visual relationship for image captioning,” *European Conference on Computer Vision*, pp. 711–727, 2018.

[6] Zechao Li, Jinhui Tang, and Tao Mei, “Deep collaborative embedding for social image understanding,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[7] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2048–2057.

[8] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3242–3250.

[9] Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba, “Sequence level training with recurrent neural networks,” *arXiv preprint arXiv:1511.06732*, 2015.

[10] Ronald J Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[11] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel, “Self-critical sequence training for image captioning,” *Computer Vision and Pattern Recognition*, pp. 1179–1195, 2017.

[12] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu, “Multiple object recognition with visual attention,” *International Conference on Learning Representations*, 2014.

[13] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu, “Recurrent models of visual attention,” *Neural Information Processing Systems*, pp. 2204–2212, 2014.

[14] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh, “Cider: Consensus-based image description evaluation,” in *Computer Vision and Pattern Recognition*, 2015, pp. 4566–4575.

[15] 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,” *European Conference on Computer Vision*, vol. 8693, pp. 740–755, 2014.

[16] Andrej Karpathy and Fei Fei Li, “Deep visual-semantic alignments for generating image descriptions,” in *Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[17] Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu, “Bleu: a method for automatic evaluation of machine translation,” *Association for Computational Linguistics*, pp. 311–318, 2002.

[18] Banerjee Satanjeev, “Meteor : An automatic metric for mt evaluation with improved correlation with human judgments,” *ACL-2005*, pp. 228–231, 2005.

[19] Carlos Flick, “Rouge: A package for automatic evaluation of summaries,” in *The Workshop on Text Summarization Branches Out*, 2004, p. 10.

[20] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould, “Spice: Semantic propositional image caption evaluation,” *European Conference on Computer Vision*, vol. 11, no. 4, pp. 382–398, 2016.

[21] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” 2018.

[22] S. Ren, K. He, R Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, 2015.

[23] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 2015.

[24] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in *International Conference on Neural Information Processing Systems*, 2015, pp. 1171–1179.