Supplementary Material for "Semi-Autoregressive Transformer for Image Captioning"

1. Background

1.1. Autoregressive Image Captioning

Given an image I and the associated target sentence $y = (y_1, ..., y_T)$, AIC models the conditional probability as:

$$p(y|I;\theta) = \prod_{i=1}^{T} p\left(y_i|y_{<i}, I;\theta\right), \tag{1}$$

where θ is the model's parameters and $y_{<i}$ represents the words sequence before the i-th word of target y. During inference, the sentence is generated word by word sequentially, which causes high inference latency.

1.2. Non-Autoregressive Image Captioning

Non-autoregressive image captioning (NAIC) models are recently proposed to accelerate the decoding speed by discarding the words dependence within the sentence. A NAIC model generates all words simultaneously and the conditional probability can be modeled as:

$$p(y|I;\theta) = \prod_{i=1}^{T} p\left(y_i|I;\theta\right). \tag{2}$$

During inference, all words are parallelly generated in one pass and thus the inference speed is significantly improved. However, these NAIC models inevitably suffer from large generation quality degradation since they remove words dependence excessively.

2. Approach

Image Features Encoder. The encoder takes the image region features as inputs and outputs the contextual region features. It consists of a stack of L identical layers. Each layer has two sublayers. The first is a multi-head self-attention sublayer and the second is a position-wise feed-forward sublayer. Both sublayers are followed by residual connection [10] and layer normalization [2] operations for stable training. Multi-head self-attention builds on scaled dot-product attention, which operates on a query Q, key K and value V as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{3}$$

where d_k is the dimension of the key. Multi-head self-attention firstly projects the queries, keys and values h times with different learned linear projections and then computes scaled dot-product attention for each one. After that, it concatenates the results and projects them with another learned linear projection:

$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V), \tag{4}$$

$$MultiHead(Q, K, V) = Concat(H_1, \dots H_h)W^O, \tag{5}$$

where $W_i^Q, W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. The self-attention in the encoder performs attention over itself, i.e., ($Q = K = V$), which is the image region features in the first layer. After a multi-head self-attention sublayer, the position-wise feed-forward sublayer (FFN) is applied to each position separately and identically:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2, \tag{6}$$

where $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$, $b_1 \in \mathbb{R}^{d_{ff}}$ and $b_2 \in \mathbb{R}^{d_{model}}$ are learnable parameter matrices.

3. Experiments

3.1. Dataset and Evaluation Metrics.

MSCOCO [3] is the widely used benchmark for image captioning. We use the 'Karpathy' splits [12] for offline experiments. This split contains $113,287$ training images, $5,000$ validation images and $5,000$ testing images. Each image has 5 captions. We also upload generated captions of MSCOCO official testing set, which contains $40,775$ images for online evaluation. We evaluate the quality of captions by standard metrics[3] , including BLEU-1/4, METEOR, ROUGE, SPICE, and CIDEr, denoted as B1/4, M, R, S, and C, respectively.

3.2. Implementation Details.

Each image is represented as 10 to 100 region features with $2,048$ dimensions extracted by Anderson *et al.* [1]. The dictionary is built by dropping the words that occur less than 5 times and ends up with a vocabulary of $9,487$. Captions longer than 16 words are truncated.

Models	BLEU-1	BLEU-4	METEOR	ROUGE	SPICE	CIDEr	Latency	Speedup
Autoregressive models								
NIC-v2 [16]	/	32.1	25.7	/	/	99.8	/	/
Up-Down [1]	79.8	36.3	27.7	56.9	21.4	120.1	/	/
AOANet [11]	80.2	38.9	29.2	**58.8**	22.4	129.8	/	/
M2-T† [4]	**80.8**	**39.1**	**29.2**	58.6	22.6	**131.2**	/	/
AIC† (bw = 1)	80.5	38.8	29.0	58.7	22.7	128.0	135ms	**2.25**×
AIC† (bw = 3)	**80.8**	**39.1**	29.1	**58.8**	**22.9**	129.7	304ms	1.00×
Non-autoregressive models								
MNIC† [7]	75.4	30.9	27.5	55.6	21.0	108.1	-	2.80×
FNIC †[6]	/	36.2	27.1	55.3	20.2	115.7	-	8.15×
MIR †[14]	/	32.5	27.2	55.4	20.6	109.5	-	1.56×
CMAL †[9]	**80.3**	**37.3**	**28.1**	**58.0**	**21.8**	**124.0**	-	**13.90**×
Partially Non-autoregressive models								
PNAIC(K=2) †[5]	80.4	38.3	**29.0**	58.4	22.2	**129.4**	-	2.17×
PNAIC(K=5) †[5]	80.3	38.1	28.7	58.3	22.0	128.5	-	3.59×
PNAIC(K=10) †[5]	79.9	37.5	28.2	58.0	21.8	125.2	-	5.43×
SATIC(K=2, bw=3) †	**80.8**	**38.4**	28.8	58.5	**22.7**	129.0	184ms	1.65×
SATIC(K=2, bw=1) †	80.7	38.3	28.8	**58.5**	22.7	128.8	76ms	4.0×
SATIC(K=4, bw=3) †	80.7	38.1	28.6	58.4	22.4	127.4	127ms	2.39×
SATIC(K=4, bw=1) †	80.6	37.9	28.6	58.3	22.3	127.2	46ms	6.61×
SATIC(K=6, bw=3) †	80.6	37.6	28.3	58.1	22.1	126.2	119ms	2.55×
SATIC(K=6, bw=1) †	80.6	37.6	28.3	58.1	22.2	126.2	35ms	**8.69**×

Table 1. Performance comparisons with different evaluation metrics on the MS COCO offline test set. "†" indicates the model is based on Transformer architecture. AIC is our implementation of the Transformer-based autoregressive model, which has the same structure as SATIC models and is used as the teacher model for sequence knowledge distillation. "/" denotes that the results are not reported. "bw" denotes the beam width used for beam search. Latency is the time to decode a single image without minibatching, averaged over the whole test split. The Speedup values are from the corresponding papers. Since latency is influenced by platform, implementation and hardware, it is not fair to directly compare them. A fairer alternative way is to compare speedup, which is calculated based on their own baseline.

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down* [1]	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
AOANet [11]	81.0	95.0	65.8	89.6	51.4	81.3	39.4	71.2	29.1	38.5	58.9	74.5	126.9	129.6
M2-T* [4]	81.6	96.0	66.4	90.8	51.8	82.7	39.7	72.8	29.4	39.0	59.2	74.8	129.3	132.1
CMAL [9]	79.8	94.3	63.8	87.2	48.8	77.2	36.8	66.1	27.9	36.4	57.6	72.0	119.3	121.2
PNAIC [5]	80.1	94.4	64.0	88.1	49.2	78.5	36.9	68.2	27.8	36.4	57.6	72.2	121.6	122.0
SATIC (K=4)	80.3	94.5	64.4	87.9	49.2	78.2	37.0	67.2	28.2	37.0	57.8	72.6	121.5	124.1

Table 2. Results on the online MSCOCO test server. $*$ denotes ensemble model.

Both our SATIC model and autoregressive image captioning base model (AIC) almost follow the same model hyperparameters setting as in [15] ($d_{model} = 512, d_k = d_v = 64, d_{ff} = 2048, L = 6, h = 8, p_{dropout} = 0.1$). As for the training process, we train AIC under cross entropy loss for 15 epochs with a mini batch size of 10, and optimizer in [15] is used with a learning rate initialized by 5e-4 and the warmup step is set to 20000. We increase the scheduled sampling probability by 0.05 every 5 epochs. We then optimize the CIDEr score with self-critical training for another 25 epochs with an initial learning rate of 1e-5. Our best SATIC model shares the same training script with AIC model except that we first initialize the weights (weight-init) of SATIC model with the pre-trained AIC model and replace the ground truth captions in the training set with sequence knowledge (SeqKD) [13, 8] results of AIC model with beam size 5 during cross entropy training stage. Unless otherwise indicate, latency represents the time to decode a single image without batching averaged over the whole test split, and is tested on a NVIDIA Tesla T4 GPU. The time

for image feature extraction is not included in latency.

3.3. Quantitative Results

In this section, we will analyse SATIC in detail by answering following questions.

How does SATIC perform compared with state-of-the-art models? We also show the results of online MSCOCO evaluation in Table 2.

References

[1] Peter Anderson, *et al*. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 1, 2

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv*, 2016. 1

[3] Xinlei Chen, *et al*. Microsoft coco captions: Data collection and evaluation server. *arXiv*, 2015. 1

[4] Marcella Cornia, *et al*. Meshed-memory transformer for image captioning. In *CVPR*, 2020. 2

[5] Zhengcong Fei. Partially non-autoregressive image captioning. In *AAAI*, 2021. 2

[6] Zheng-cong Fei. Fast image caption generation with position alignment. *arXiv*, 2019. 2

[7] Junlong Gao, *et al*. Masked non-autoregressive image captioning. *arXiv*, 2019. 2

[8] Jiatao Gu, *et al*. Non-autoregressive neural machine translation. *arXiv*, 2017. 2

[9] Longteng Guo, *et al*. Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. *arXiv*, 2020. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016. 1

[11] Lun Huang, *et al*. Attention on attention for image captioning. In *ICCV*, 2019. 2

[12] Andrej Karpathy and Li Feifei. Deep visual-semantic alignments for generating image descriptions. *CVPR*, 2015. 1

[13] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv*, 2016. 2

[14] Jason Lee, *et al*. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv*, 2018. 2

[15] Ashish Vaswani, *et al*. Attention is all you need. In *NeurIPS*, 2017. 2

[16] Oriol Vinyals, *et al*. Show and tell: A neural image caption generator. In *CVPR*, 2015. 2