This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Semi-Autoregressive Transformer for Image Captioning

Yuanen Zhou[1,3], Yong Zhang[2], Zhenzhen Hu[1,3], Meng Wang[1,3]
[1]School of Computer Science and Information Engineering, Hefei University of Technology
[2]Tencent AI Lab [3]Key Laboratory of Knowledge Engineering with Big Data (MoE)
{y.e.zhou.hb,zhangyong201303,huzhen.ice,eric.mengwang}@gmail.com

Abstract

Current state-of-the-art image captioning models adopt autoregressive decoders, i.e. *they generate each word by conditioning on previously generated words, which leads to heavy latency during inference. To tackle this issue, non-autoregressive image captioning models have recently been proposed to significantly accelerate the speed of inference by generating all words in parallel. However, these non-autoregressive models inevitably suffer from large generation quality degradation since they remove words dependence excessively. To make a better trade-off between speed and quality, we introduce a semi-autoregressive model for image captioning (dubbed as SATIC), which keeps the autoregressive property in global but generates words parallelly in local . Based on Transformer, there are only a few modifications needed to implement SATIC. Experimental results on the MSCOCO image captioning benchmark show that SATIC can achieve a good trade-off without bells and whistles. Code is available at https://github.com/YuanEZhou/satic.*

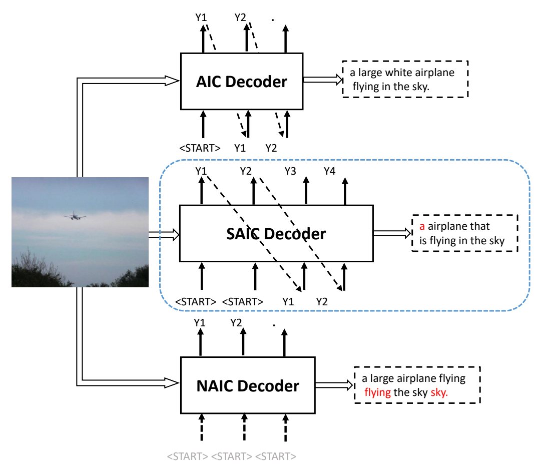

Figure 1. Given an image, autoregressive image captioning (AIC) model generates a caption word by word and Non-Autoregressive Image Captioning (NAIC) model outputs all words in parallel, while Semi-Autoregressive image captioning (SAIC) model falls in between, which keeps the autoregressive property in global but outputs words parallelly in local. We mark error words by red font.

1. Introduction

Image captioning [24, 28, 18], which aims at describing the visual content of an image with natural language sentence, is one of the important tasks to connect vision and language. Most proposed models typically follow the encoder/decoder paradigm. In between, convolutional neural network (CNN) is utilized to encode an input image and recurrent neural networks (RNN) or Transformer [22] is adopted as sentence decoder to generate a caption. Current state-of-the-art models adopt autoregressive decoders which means that they generate one word at each time step by conditioning on all previously produced words. Though impressive results have been achieved, these models suffer from high latency during inference owing to the autoregressive property, which is unaffordable for real-time industrial scenarios sometimes.

To tackle this issue, there is an increasing interest to develop non-autoregressive decoding [9, 15, 6, 11] to significantly accelerate inference speed by generating all target words parallelly. These non-autoregressive models have basically the same structure as the autoregressive Transformer model [22]. The difference lie in that non-autoregressive models generate all words independently (as shown in the bottom of Figure 1) instead of generating one word at each time step by conditioning on the previously produced words as in autoregressive models (as shown in the top of Figure 1). However, these non-autoregressive models suffer from words repetition or omission problem compared to their autoregressive counterparts owing to removing the sequential dependence excessively.

To alleviate the above issue, some methods have been proposed to seek a trade-off between speed and quality. For example, iteration refinement based methods [15, 7, 2] try to compensate for the word independence assumption by taking caption output from preceding iteration as input and then polishing it until reaching max iteration number or no change appears. Nevertheless, it needs multiple times refinement to achieve better quality, which hurts decoding speed significantly. Some works [6, 10] try to enhance the decoder input by providing more target side context information, while they commonly incorporate extra modules and thus extra computing overhead. Besides, partially non-autoregressive models [5, 19] are proposed by considering a sentence as a series of concatenated word groups. The groups are generated parallelly in global while each word in group is predicted from left to right. Though better trade-off is achieved, the training paradigm of such model is somewhat tricky because it must need to incorporate curriculum learning-based training tasks of group length prediction and invalid group deletion [5].

In contrast, the model (SATIC) introduced in this paper can achieve similar trade-off performance but without bells and whistles during training. Specifically, SATIC also considers a sentence as a series of concatenated word groups, as similar with [5, 19]. However, all words in a group are predicted in parallel while the groups are generated from left to right, as shown in the middle of Figure 1. This means that SATIC keeps the autoregressive property in global and the non-autoregressive property in local and thus gets the best of both world. In other words, SATIC can directly inherit the mature training paradigm of autoregressive captioning models and get the speedup benefit of non-autoregressive captioning models.

We evaluate SATIC model on the challenging MSCOCO [3] image captioning benchmark. Experimental results show that SATIC achieves a better balance between speed, quality and easy training. Specifically, SATIC generates captions better than non-autoregressive models and faster than autoregressive models and is easier to be trained than partially non-autoregressive models [5]. Besides, we conduct substantial ablation studies to better understand the effect of each component of the whole model.

2. Related Work

Image Captioning. Over the last few years, a broad collection of methods have been proposed in the field of image captioning. In a nutshell, we have gone through grid-feature [27, 13] then region-feature [1] and relation-aware visual feature [29, 28] on the image encoding side. On the sentence decoding side, we have witnessed LSTM [24], CNN [8] and Transformer [4] equipped with various attention [12, 31, 18] as decoder. On the training side, models are typically trained by step-wise cross-entropy loss and then Reinforcement Learning [21], which enables the use of non-differentiable caption metrics as optimization objectives and makes a notable achievement. Recently, vision-language pre-training has also been adopted for image captioning and show impressive result. These models [30, 16] are firstly pretrained on large image-text corpus

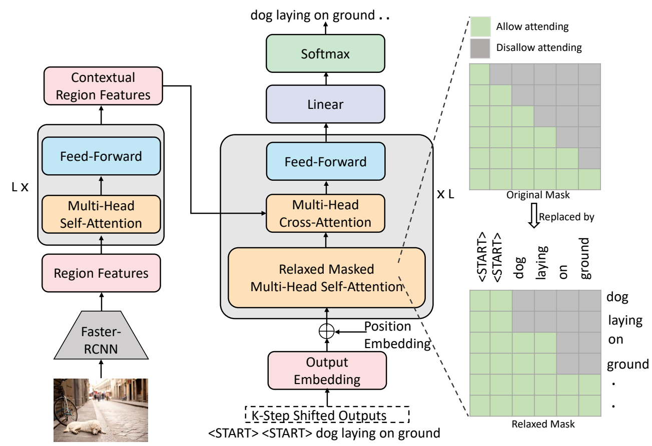

Figure 2. Illustration of Transformer-based semi-autoregressive image captioning model (SATIC), which composes of an encoder and a decoder. Without loss of generality, we set K=2 for convenience. Notice that the Residual Connections, Layer Normalization are omitted.

and then finetuned. It is noteworthy that it's not fair to directly compare them with non-pretraining-finetuning methods. Though impressive performance has been achieved, most state-of-the-art models adopt autoregressive decoders and thus suffer from high latency during inference.

Non-Autoregressive Decoding. Due to the downside of autoregressive decoding, Non-autoregressive decoding has firstly aroused widespread attention in the community of Neural Machine Translation (NMT). Non-autoregressive NMT was first proposed in [9] to significantly improve the inference speed by generating all target-side words in parallel. While the decoding speed is improved, it often suffers from word repetition or omission problem due to removing words dependence excessively. Some methods have been proposed to overcome this problem, including knowledge distillation [9], well-designed decoder input [10], auxiliary regularization terms [26], iterative refinement [15], and partially-autoregressive model [19, 25]. Following similar research roadmap, non-autoregressive decoding has recently been introduced to visual captioning task [7, 11, 6, 2, 5]. This work pursues the semi-autoregressive decoding in NMT [25] for image captioning and further explores its effectiveness under the context of reinforcement training.

3. Approach

In this section, we first present the architecture of SATIC model built on the well-known Transformer [22] and then introduce the training procedure for model optimization.

3.1. Transformer-Based SATIC Model

Given the image region features extracted by a pre-trained Faster-RCNN model [20, 1], SATIC aims to generate a caption in a semi-autoregressive manner. The architecture of SATIC model is shown in Figure 2, which consists of an encoder and decoder.

Image Features Encoder. The encoder, which is basically the same as the Transformer encoder [22], takes the image region features as input and outputs the contextual region features. More details can be found in supplementary material.

Captioning Decoder. The decoder takes contextual region features and previous word embedding features as input and outputs predicted words probability. It is basically the same as the Transformer decoder [22] except that original masked multi-head attention is replaced with the relaxed masked multi-head self-attention. Specifically, the original lower triangular matrix mask is now replaced by the relaxed mask. Formally, given the caption length T and group size K, the relaxed mask $M \in \mathbb{R}^{T\times T}$ is defined as:

$$M[i][j] = \begin{cases} 0 & \text{if } j < ([(i-1)/K]+1) \times K, \text{allow} \\ -\infty & \text{other, } \quad \text{disallow attending} \end{cases}, \tag{1}$$

where $i, j \in [1, T]$ and $[\cdot]$ denotes floor operation. An intuitive example is shown in the right of Figure 2, where $T = 6$ and $K = 2$. As a consequence, the scaled dot-product attention in relaxed masked multi-head self-attention module is modified to:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}} + M)V, \tag{2}$$

where Q/K/V denotes query/key/value and d_k is the dimension of the key. Thanks to the relaxed mask, all words in a group can now access all words in its preceding groups. Different from original autoregressive Transformer model, which outputs a word at each step, SATIC takes a group of words as input and outputs a group of words at each step during decoding. Each group contains K consecutive words. For example, at the beginning of decoding, we feed the model with K <START> symbols to predicate $y_1, ..., y_K$ and then $y_1, ..., y_K$ are fed as input to predicate $y_{K+1}, ..., y_{2K}$ in parallel. This process will continue until the end of sentence. An intuitive example is shown in the middle of Figure 1 with $K = 2$. .

3.2. Training

Since SATIC model keeps the autoregressive property in global and the non-autoregressive property in local, it gets the best of both world and the conditional probability can be formulated as:

$$p(y|I;\theta) = \prod_{t=1}^{[(T-1)/K]+1} p\left(G_t|G_{<t}, I;\theta\right), \tag{3}$$

where θ is the model's parameters and $y = (y_1, ..., y_T)$ is the associated target sentence of image I and $G_{<t}$ represents the groups before t-th group and $G_t = y_{(t-1)K+1}, ..., y_{tK}$ except for the last group which may have less than K words. If the length of padded word sequence can't be divided by K, remaining words only keep in output but not in input. At the first training stage, we optimize the model by minimizing cross-entropy loss (XE):

$$\begin{aligned} L_{XE}(\theta) &= -\sum_{t=1}^{[(T-1)/K]+1} \log p\left(G_t|G_{<t}, I;\theta\right) \\ &= -\sum_{t=1}^{[(T-1)/K]+1} \sum_{i=(t-1)K+1}^{tK} \log p\left(y_i|G_{<t}, I;\theta\right). \end{aligned} \tag{4}$$

At the second training stage, we finetune the model using self-critical training (SC) [21] and the gradient can be expressed as:

$$\nabla_\theta L_{SC}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}(R(\hat{y}^n_{1:T}) - b)\nabla_\theta \log p(\hat{y}^n_{1:T}|I;\theta), \tag{5}$$

where R is the CIDEr [23] score function, and b is the baseline score. We adopt the baseline score proposed in [17], where the baseline score is defined as the average reward of the rest samples rather than original greedy decoding reward. We sample $N = 5$ captions for each image and $\hat{y}^n_{1:T}$ is the n-th sampled caption.

4. Experiments

4.1. Dataset and Implementation Details.

We conduct experiments on the widely used MSCOCO [3] dataset and standard automatic evaluation metrics [3] (including BLEU, METEOR, ROUGE, SPICE, and CIDEr) are used for reporting results. We train model under cross entropy loss for 15 epochs with a mini batch size of 10 and then optimize the CIDEr score with self-critical training for another 25 epochs. Our best SATIC model shares the same training script with autoregressive image captioning model (AIC) 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 distillation (SeqKD) [14, 9] results of AIC model with beam size 5 during cross entropy training stage. More details can be found in supplementary material.

4.2. Quantitative Results

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

How does SATIC perform compared with other models? We compare SATIC model with autoregressive models, non-autoregressive models and a partially non-autoregressive model. As shown in Table 1, SATIC achieves comparable caption quality with autoregressive models but with significant speedup. When

Models	BLEU-4	METEOR	SPICE	CIDEr	Latency	Speedup
Autoregressive models						
NIC-v2 [24]	32.1	25.7	/	99.8	/	/
Up-Down [1]	36.3	27.7	21.4	120.1	/	/
AOANet [12]	38.9	29.2	22.4	129.8	/	/
M2-T† [4]	**39.1**	**29.2**	22.6	**131.2**	/	/
AIC† (bw = 1)	38.8	29.0	22.7	128.0	135ms	**2.25**×
AIC† (bw = 3)	**39.1**	29.1	**22.9**	129.7	304ms	1.00×
Non-autoregressive models						
MNIC† [7]	30.9	27.5	21.0	108.1	-	2.80×
FNIC †[6]	36.2	27.1	20.2	115.7	-	8.15×
MIR †[15]	32.5	27.2	20.6	109.5	-	1.56×
CMAL †[11]	**37.3**	**28.1**	**21.8**	**124.0**	-	**13.90**×
Partially Non-autoregressive models						
PNAIC(K=2) †[5]	38.3	**29.0**	22.2	**129.4**	-	2.17×
PNAIC(K=5) †[5]	38.1	28.7	22.0	128.5	-	3.59×
PNAIC(K=10) †[5]	37.5	28.2	21.8	125.2	-	5.43×
SATIC(K=2, bw=3) †	**38.4**	28.8	**22.7**	129.0	184ms	1.65×
SATIC(K=2, bw=1) †	38.3	28.8	22.7	128.8	76ms	4.0×
SATIC(K=4, bw=3) †	38.1	28.6	22.4	127.4	127ms	2.39×
SATIC(K=4, bw=1) †	37.9	28.6	22.3	127.2	46ms	6.61×
SATIC(K=6, bw=3) †	37.6	28.3	22.1	126.2	119ms	2.55×
SATIC(K=6, bw=1) †	37.6	28.3	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 [14]. "/" 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.

$K = 2$, SATIC achieves about $1.5\times$ speedup while the caption quality only degrades slightly compared with its autoregressive counterpart AIC model. Compared with non-autoregressive models , SATIC obviously achieves a better trade-off between quality and speed by outperforming all the non-autoregressive models except CMAL in speedup metric. Compared with most similar partially non-autoregressive model PNAIC, SATIC achieves similar speedup and caption evaluation results. It is worth nothing that SATIC outperforms PNAIC on SPICE metric, which concerns more on semantic propositional content. What's more, the training of SATIC is more easy and straightforward than PNAIC. Overall, SATIC achieves a better trade-off between quality, speed and easy training.

What is the effect of group size K? We test three different setting of the group size, *i.e.* $K \in \{2, 4, 6\}$. From the bottom of Table 1, we can observe that a larger K brings more significant speedup while the caption quality degrades moderately. For example, the decoding speed increases about $1.5\times$ while the CIDEr score drops about 1.5 when K grows from 2 to 4, and drops no more than 3 when K grows further to 6. This is intuitive since K is the indicator of parallelization and also indicates that SATIC model is relatively stable to K.

Can SATIC benefit from beam search? From the bottom of Table 1, we can also find that SATIC benefits little (CIDEr score only increases 0.2) from beam search compared with its autoregressive counterpart AIC (CIDEr score increases 1.7) after self-critical training. There are two possibilities, the one is self-critical training make its output probability concentrated and the other one is SATIC can not benefit from beam search. To investigate whether SATIC can benefit from beam search, we test its output just after cross entropy training with different beam search width. From Table 2, we can observe that SATIC model can still benefit from beam search and there are two interesting phenomena: 1) SATIC with too large K benefits less from beam search and 2) the effect of beam search is larger when without weight initialization and sequence knowledge distillation. A plausible explanation is that long-distance dependence is hard to capture and sequence knowledge distillation decreases dependence among words.

What is the effect of sequence knowledge distillation and weight initialization? We further investigate the effect of sequence knowledge distillation and weight initialization and results are shown in Table 3. We can find that sequence knowledge distillation plays an important role in both XE and SC training stages and the effect is more significant in XE stage. Basically, the larger the K is, the more obvious the effect is. This is intuitive since the ability of SATIC model to capture conditional probability is undermined when K grows and sequence knowledge distillation compensates for this by reducing the complexity of data sets. SC can also alleviate sentence-level inconsistency by providing sentence-level reward. In addition to accelerate convergence, we can find that weight initialization slightly improves the caption quality when K is small but has important impact when K is large.

What is the effect of batch size on latency? Above latency is measured with batch size set to 1. However, there may be multiple requests at once in real application. So, we further investigate the latency under various batch size setting. From Table 4, we can find that SATIC can basically accelerate decoding even under large batch size. We can also observe that the speedup is decline as batch size increases. This indicates that non-gpu program becomes a bottleneck when the runtime of gpu program is negligible.

Models	bw	B1	B4	M	R	S	C
w/ weight-init and SeqKD:							
K=2	1	79.3	36.2	28.2	57.4	22.1	121.5
	3	80.0	37.3	28.4	57.8	22.3	123.9
K=4	1	77.3	32.9	27.0	56.0	20.5	111.0
	3	78.0	34.4	27.2	56.5	20.9	114.5
K=6	1	77.3	33.3	26.7	56.0	20.4	110.3
	3	77.2	33.7	26.6	56.1	20.4	110.7
w/o weight-init and SeqKD :							
K=2	1	74.2	29.1	25.8	53.8	19.5	100.0
	3	76.0	32.8	26.8	55.2	20.6	107.2
K=4	1	65.8	17.0	21.8	48.3	16.1	73.3
	3	69.7	20.9	22.7	50.2	16.6	79.6
K=6	1	67.6	17.6	21.3	48.6	15.2	73.5
	3	68.8	19.1	21.6	49.6	15.5	76.3

Table 2. The results after XE training when using different beam search width.

4.3. Qualitative Results

We present three examples of generated image captions in Figure 3. From the top example, we can intuitively understand the

Models	XE					SC				
	B1	B4	M	S	C	B1	B4	M	S	C
K=2:										
Base	74.2	29.1	25.8	19.5	100.0	80.3	37.6	28.4	22.0	123.7
+SeqKD	78.8	36.0	28.0	21.8	120.4	80.5	38.4	28.7	22.6	128.1
+Weight-init	79.3	36.2	28.2	22.1	121.5	80.7	38.3	28.8	22.7	128.8
K=4:										
Base	65.8	17.0	21.8	16.1	73.3	79.8	35.3	27.3	20.9	119.5
+SeqKD	69.5	22.2	23.0	16.7	85.5	80.4	37.5	28.3	22.2	126.0
+Weight-init	77.3	32.9	27.0	20.5	111.0	80.6	37.9	28.6	22.3	127.2
K=6:										
Base	67.6	17.6	21.3	15.2	73.5	79.2	32.2	26.7	20.4	116.1
+SeqKD	73.2	27.0	24.6	18.1	96.3	79.9	37.0	28.0	21.6	123.9
+Weight-init	77.3	33.3	26.7	20.4	110.3	80.6	37.6	28.3	22.2	126.2

Table 3. The effect of sequence knowledge distillation (SeqKD) and weight initialization (Weight-init) . Beam width is set to 1.

Model	b=1	b=8	b=16	b=32	b=64
Transformer	135ms	22ms	13ms	11ms	10ms
SATIC, K=2	76ms	13ms	8ms	7ms	7ms
SATIC, K=4	46ms	8ms	6ms	5ms	5ms
SATIC, K=6	35ms	7ms	5ms	5ms	5ms

Table 4. Time needed to decode one sentence under various batch size settings. Beam width is set to 1 since we find that larger beam width brings little performance boost but significant latency to SATIC model after self-critical training.

Figure 3. Examples of the generated captions. GT denotes ground-truth caption. Base here denotes SATIC(k=4) model trained by cross entropy loss using original training set. We mark repeated words by red font.

effect of sequence knowledge distillation (SeqKD) and self-crical training (SC) in reducing repeated words and incomplete content. In general, the final SATIC models with different group size K can generate fluent captions, as shown in the middle example. Nevertheless, repeated words and incomplete content issues still exist, especially when K is large. As shown in the bottom example, '*a*' and '*orange*' are repeated words and '*tray*','*bowl*' are missing.

5. Conclusion

In this paper, we introduce a semi-autoregressive model for image captioning (dubbed as SATIC), which keeps the autoregressive property in global and non-autoregressive property in local. We conduct substantial experiments on MSCOCO image captioning benchmark to better understand the effect of each component. Overall, SATIC achieves a better trade-off between speed,quality and easy training.

References

[1] Peter Anderson, *et al*. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018. 2, 4

[2] Yang Bang, *et al*. Non-autoregressive coarse-to-fine video captioning. In *AAAI*, 2021. 1, 2

[3] Xinlei Chen, *et al*. Microsoft coco captions: Data collection and evaluation server. *arXiv*, 2015. 2, 3

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

[5] Zhengcong Fei. Partially non-autoregressive image captioning. In *AAAI*, 2021. 2, 4

[6] Zheng-cong Fei. Fast image caption generation with position alignment. *arXiv*, 2019. 1, 2, 4

[7] Junlong Gao, *et al*. Masked non-autoregressive image captioning. *arXiv*, 2019. 1, 2, 4

[8] Jiuxiang Gu, *et al*. An empirical study of language cnn for image captioning. In *ICCV*, 2017. 2

[9] Jiatao Gu, *et al*. Non-autoregressive neural machine translation. *arXiv*, 2017. 1, 2, 3

[10] Junliang Guo, *et al*. Non-autoregressive neural machine translation with enhanced decoder input. In *AAAI*, 2019. 2

[11] Longteng Guo, *et al*. Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. *arXiv*, 2020. 1, 2, 4

[12] Lun Huang, *et al*. Attention on attention for image captioning. In *ICCV*, 2019. 2, 4

[13] Huaizu Jiang, *et al*. In defense of grid features for visual question answering. In *CVPR*, 2020. 2

[14] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv*, 2016. 3, 4

[15] Jason Lee, *et al*. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv*, 2018. 1, 2, 4

[16] Xiujun Li, *et al*. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 2

[17] Ruotian Luo. A better variant of self-critical sequence training. *arXiv*, 2020. 3

[18] Yingwei Pan, *et al*. X-linear attention networks for image captioning. In *CVPR*, 2020. 1, 2

[19] Qiu Ran, *et al*. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. *arXiv*, 2020. 2

[20] Shaoqing Ren, *et al*. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[21] Steven J Rennie, *et al*. Self-critical sequence training for image captioning. In *CVPR*, 2017. 2, 3

[22] Ashish Vaswani, *et al*. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 3

[23] Ramakrishna Vedantam, *et al*. Cider: Consensus-based image description evaluation. *Computer Science*, 2015. 3

[24] Oriol Vinyals, *et al*. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1, 2, 4

[25] Chunqi Wang, *et al*. Semi-autoregressive neural machine translation. *arXiv*, 2018. 2

[26] Yiren Wang, *et al*. Non-autoregressive machine translation with auxiliary regularization. *arXiv*, 2019. 2

[27] Kelvin Xu, *et al*. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015. 2

[28] Xu Yang, *et al*. Auto-encoding scene graphs for image captioning. In *CVPR*, 2019. 1, 2

[29] Ting Yao, *et al*. Exploring visual relationship for image captioning. In *ECCV*, 2018. 2

[30] Luowei Zhou, *et al*. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, 2020. 2

[31] Yuanen Zhou, *et al*. More grounded image captioning by distilling image-text matching model. In *CVPR*, 2020. 2