Partially Non-Autoregressive Image Captioning

Zhengcong Fei[1,2]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
feizhengcong@ict.ac.cn

Abstract

Current state-of-the-art image captioning systems usually generated descriptions autoregressively, *i.e.*, every forward step conditions on the given image and previously produced words. The sequential attribution causes a unavoidable decoding latency. Non-autoregressive image captioning, on the other hand, predicts the entire sentence simultaneously and accelerates the inference process significantly. However, it removes the dependence in a caption and commonly suffers from repetition or missing issues. To make a better trade-off between speed and quality, we introduce a partially non-autoregressive model, named PNAIC, which considers a caption 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, and thus the captioner can create multiple discontinuous words concurrently at each time step. More importantly, by incorporating curriculum learning-based training tasks of group length prediction and invalid group deletion, our model is capable of generating accurate captions as well as preventing common incoherent errors. Extensive experiments on MS COCO benchmark demonstrate that our proposed method achieves more than 3.5× speedup while maintaining competitive performance.

1 Introduction

Describing the visual content of an image accurately and quickly is a fundamental goal for the artificial intelligence area, which has a wide range of applications in research and production. Although image captioning has achieved state-of-the-art performance under an encoder-decoder paradigm in recent years (Vinyals et al. 2015), most captioning models still suffer from the slow decoding speed problem due to their autoregressive property, that is, the generation of next target word depends on the given image and all previously produced words, making the decoding process intrinsically nonparallelizable and the inference high latency (Gu et al. 2018; Gao et al. 2019). It is unaffordable for real-time industrial scenarios sometimes. Recently, a flurry of non-autoregressive image captioning methods has been investigated to mitigate the slow decoding speed problem by generating all words independently in parallel (Jason et al. 2018; Guo et al. 2020; Fei 2019). Specifically, non-autoregressive models take basically the same structure as the autoregressive Transformer network (Vaswani et al. 2017). However, instead of conditioning on the previously generated words as the decoder in autoregressive models, they generate all words in one step. Nevertheless, because of the lock of sufficient dependency information about surrounding words, these captioning models suffer from poor captioning quality compared with autoregressive counterparts.

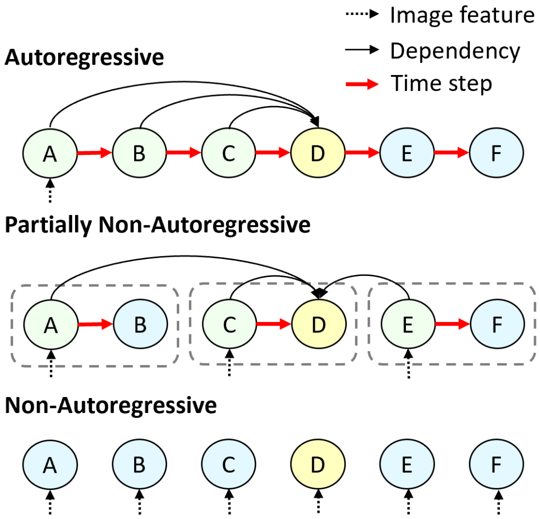


Figure 1: A conceptual overview of autoregressive, partially non-autoregressive, and non-autoregressive caption decoder. Autoregressive decoder generates the next word conditioned on all preceding subsentence, while non-autoregressive decoder outputs all words in parallel. Comparatively, paritally non-autoregressive decoder produces a caption as a series of word groups. The groups are generated simultaneously while each group is generated word-by-word conditioned on both the image feature and history of all groups, *e.g.*, the word “D” in the second group is predicted based on the words “A”,“C”, and “E”.

Numerious works have been proposed to handle the above issues and seeking a trade-off between inference time and caption performance. Generally, previous approaches can be divided into two categories. One kind of works focuses on incorporate iterative refinement framework to get out of the independence dillemma, which takes the caption hypothesis from the preceding iteration as a reference and regularly polishes the new caption until achieving the predefined iteration count or no change appears (Gao et al. 2019; Jason et al. 2018). Nevertheless, comparable captioning performance is based on multiple refinement times, which slows decoding significantly sometimes. The other knid of works tries to adjust the Transfomer structure to better capture dependency and position information by leveraging extra autoregressive layers in the decoder (Fei 2019, 2020b), introducing latent variables to eliminate the modal gap and more powerful probabilistic frameworks to simulate more complicated distributions. Besides, (Guo et al. 2020) introduces a multi-agent reinforcement learning-based training paradigm to improve sentence-level consistency.

Inspired from paragraph generation (Bernardi et al. 2016), which aims to generate multiple sentences parallelly, in this paper, we propose a partially non-autoregressive model named PNAIC for group-level image captioning acceleration. Innovatively, our model incorporates several fundamental diversities with respect to all previous image captioning algorithms: First, we hypothesize that a caption consists of a series of word groups corresponding different image regions, which can be generated parallelly. Second, to better capture dependency and semantic information, the words inside a group are autoregressively generated conditioned not only on the previously generated words in this group but also on those in other groups. Figure 1 provides an illustration of the different autoregressive properties for caption generation. By conditioning on previously generated words in other groups, the captioner can observe what feasible caption candidates have been selected by each group and adapt accordingly, *i.e.*, avoiding from missing and meaningless word errors. Resultingly, our model captures more word dependency such that the consistency issues can be reduced naturally. Third, to make the captioner capable of parallel decoding gradually, we design training curriculum of length prediction and invalid deletion from easy to hard. Accordingly, our model learns to mark a group to be deleted when it finds the content is invalid.

Contributions. To sum up, our contributions are as follows:

- We propose a partially non-autoregressive model to accelerate image captioning generation, splitting each caption into a series of word groups. The captioner keeps the autoregressive property in local but relieves in global. To our knowledge, this is the first work to introduce a partially non-autoregressive paradigm into image captioning.
- We design two curriculum learning-based training tasks: group length prediction and invalid group deletion, which can prevent our captioning model from resulting in incoherent issues with a progressive way.
- We conduct experiments on MS COCO benchmark to evaluate the proposed method. Experimental results show that PNAIC is able to decode over 3.5× faster than the autoregressive counterpart while strikingly narrowing the performance gap. The source code is publicly released on https://github.com/feizc/PNAIC.

2 Background

2.1 Autoregressive Image Captioning

Given an image I as input and a matched target sentence $S = \{w_1, \dots, w_T\}$, autoregressive image captioning (AIC) systems construct the conditional probability as:

$$P(S|I) = \prod_{t=1}^{T} P(w_t|w_{<t}, I), \tag{1}$$

where $w_{<t} = \{w_1, \dots, w_{t-1}\}$ denotes the captioning history. During training, ground-truth words are fed into the caption decoder; While in inference, the preceding sentence generated by decoding strategies, such as greedy search and beam search, is fed into the caption decoder to guide the generation of next word. The prominent feature of the AIC model is that it requires historical information in the decoding procedure, and the target words are produced in the one-by-one style. Due to such a autoregressive property, the decoding speed is limited, which restricts the real-time application of autoregressive image captioning sometimes.

2.2 Non-Autoregressive Image Captioning

Non-autoregressive image captioning (NAIC) (Fei 2019; Gao et al. 2019) is proposed to accelerate the caption decoding process, which can simultaneously generate words by discarding the sequential dependencies within the sentence. The conditional probability can be modeled as:

$$P(S|I) = \prod_{t=1}^{T} P(w_t|I). \tag{2}$$

During decoding, all words could be easily obtained with maximum likelihood parallelly in one pass. Compared to conventional AIC models, non-autoregressive image captioning achieve a significant speedup. However, NAIC models independently predict all words in one caption, which results in a weakness in exploiting the words dependency knowledge for generating accurate descriptions and thus suffering from a large gap in the captioning quality.

3 Approach

In this section, we first present the basic architecture of our PNAIC model built upon the well-known Transformer (Vaswani et al. 2017), and then introduce two progressive training tasks, including group length prediction and invalid group deletion, to boost captioning performance. Finally, we provide a discussion about theoretical complexity of different decoding approaches as well as a unified perspective.

3.1 Transformer-based Model Structure

Provided with the region features of an image extracted from a pre-trained CNN (Anderson et al. 2018), PNAIC aim to generate a descriptive caption in a partially non-autoregressive manner. Referred as (Cornia et al. 2020), our Transformer-based captioning model composes of an image feature encoder and caption decoder. An overview of the architecture of our PNAIC is presented in Figure 2.

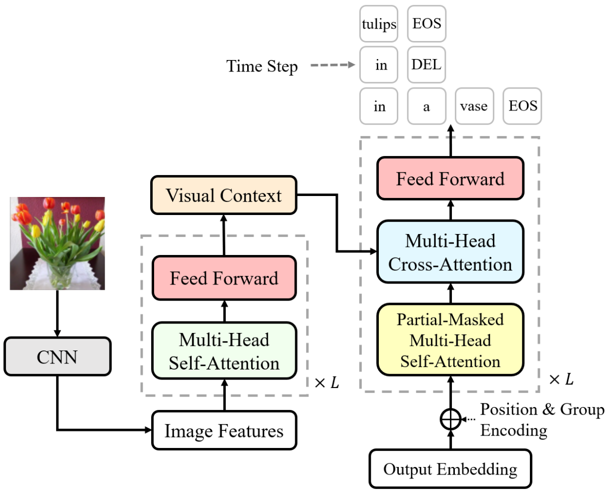

tulips
EOS
Time Step
in
DEL
in
a
vase
EOS
Visual Context
Feed Forward
Multi-Head Self-Attention
× L
CNN
Image Features
Feed Forward
Multi-Head Cross-Attention
Partial-Masked Multi-Head Self-Attention
× L
Position & Group Encoding
Output Embedding

Figure 2: An overview of partially non-autoregressive image captioning system, which consists of an encoder and a decoder. The decoder builds each group inside autoregressive but mutually parallelized. As a result, our method can produce multiple words simultaneously at each time step.

Image Feature Encoder The encoder, which takes the image features as inputs and generates the weighted visual representation, is composed of a stack of L identical layers, and each layer has two sublayers: (1) a multi-head attention layer, and (2) a position-wise feed-forward layer. Both of the sublayers are followed by a residual connection operation and a layer normalization operation. Specifically, multi-head attention adopts dot-product operation which processes a set of queries (Q), keys (K), and values (V) simultaneously as:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \quad (3)$$

where d_k is the dimension of the key. Multi-head attention first computes h different queries, keys, and values with linear projections and computes scaled dot-product attention, then concatenates the results and projects with another linear projection:

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

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, \ldots, H_h), \quad (5)$$

where $W_i^Q, W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ and $W_i^v \in \mathbb{R}^{d_{model} \times d_v}$ are corresponding parameter matrice. Note that the parameters are different each time queries, keys and values undergo a linear transformation. The self-attention in the decoder performs attention over itself, that is, $Q = K = V$. After a multi-head attention sublayer, the feed-forward network (FFN) is applied to further adjust the representations.

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

where $W_1 \in \mathbb{R}^{d_{modl} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$, $b_1 \in \mathbb{R}^{d_{ff}}$ and $b_2 \in \mathbb{R}^{d_{model}}$ denote learnable parameters.

Caption Decoder The decoder is also composed of a stack of L identical layers. For each layer, besides the multi-head sublayer and FFN sublayer, a third sublayer is inserted, called cross-attention layer. The cross-attention sublayer performs multi-head attention over visual context with the multi-head sublayer's output in the same layer as the query. Residual connection and layer normalization are also applied after each sublayer. In addition, unlike the image feature encoder, a mask matrix is applied to ensure that the prediction for position t can depend only on the known outputs less than t at each group.

Position Encoding Since the length of each group is dynamically determined, the positions of the word in the caption can not be represented directly following (Vaswani et al. 2017). To solve this problem, we introduce a two-stage position encoding: (1) an identical method to independently encode the position in the corresponding group of each word and (2) an absolute group embedding method, which uses a distinct trainable vector to represent the position of each group. Formally, the input embedding of the decoder for the t-th target word w of the i-th group is computed as:

$$E_w = E_w^{word} + E_i^{group} + PE_t, \quad (7)$$

where E_v^{word} is the word embedding vector, E_i^{group} is the group embedding vector and PE_t denotes the classical sinusoidal positional embedding vector, respetively.

3.2 Partially Non-Autoregressive Decoding

Overall, our PNAIC model creates caption autoregressively in local and non-autoregressively in global. As illustrated in the Figure 2, the captioner simultaneously generates all groups "tulips EOS", "in DEL" and "in a vase EOS". While at each time step, captioner generates a word for each incomplete group. The special word DEL denotes the group should be deleted, and EOS denotes the end of a word group. Combing all the groups sequentially, we obtain the final caption "tulips in a vase". More broadly, assuming a target sentence S is generated as K groups including $\{G^1, G^2, \ldots, G^K\}$, where G^i is i-th sentence subsequence. For convenience, we assume that all the groups hold the same length. PNAIC predicts a word for each group conditioned on the visual context I_F and all previously generated words at each step. Hence, the probabilty can be computed as:

$$P(S|I) = \prod_{t=1}^{L_g} \prod_{i=1}^{K} P(w_t^i | G_{<t}^1, \ldots, G_{<t}^K; I_F), \quad (8)$$

where w_t^i reprersents the t-th word in the i-th group, $G_{<t}^i = \{w_1^i, \ldots, w_{t-1}^i\}$ denotes the captioning history in the i-th group, and L_g is the group length.

However, there remain two problems for such a decoding process: (1) how to determine the length of group? and (2) how to confirm a group is invalid? We make the PNAIC cover these two problems in a progressive way. Formally, we denote the original word vocabulary is V, we extend it with two extra word EOS and DEL. Then for group G^i, the most probable word $\hat{w}_t^i$ at time step t is determined according to:

$$\hat{w}_t^i = \max_{w_t^i \in V \cup \{\text{EOS}, \text{DEL}\}} P(w_t^i | G_{<t}^1, \ldots, G_{<t}^K; I_F), \quad (9)$$

which leads to three conditions: (1) $\hat{w}_t^i \in V$: the decoding process of group G^i needs to continue; (2) $\hat{w}_t^i = \texttt{EOS}$: group G^i is complete and the decoding process should stop; (3) $\hat{w}_t^i = \texttt{DEL}$: the content of group G^i is invalid. The decoding process should be also stopped and G^i should final be deleted. In summary, the entire decoding process stops when all the groups meet `EOS`/`DEL` or reach the maximum word number. Please note that we do not immediately delete a group when `DEL` is encountered but do it via post-processing after the total caption is completed.

Since there is limited dependency information available in the early stage of the decoding process, error occurrence is inevitable. In this work, to make the captioner capble of parallel decoding sufficiently and gradually, we introduce two training tasks to guide our captioner from easy to difficult.

Group Length Prediction Instead of predicting the group length before decoding, our captioner determines the group lengths by generating the `EOS` automatically. This strategy helps our model removing repetitive and missing errors to some degree. As shown in Equation 8, a word is generated conditioned on the enhanced image features and all the previously produced words in all the groups. Accordingly, the decoder has more abundant dependency information to detect and avoid such errors similar to autoregressive methods, which seldomly suffering these issues.

However, it is unfeasible to train the captioner to hold fast decoding capability while maintaining a comparable speedup. On the one hand, to accelerate the caption decoding, the training samples are supposed to split into uniform word groups to assit the captioner learn to generate as equal-length groups as possible. On the other hand, the captioner should be exposed to different training conditions to enhance its ability to detect and removing the common errors, which suggests that the target sentences of training instances should be divided randomly and uncertainly. To solve this problem, we propose an annealing dividing strategy with curriculum learning (Bengio et al. 2009). To be specific, we randomly decide whether to divide a sentence equally or randomly at each training step and gradually anneal to the equally-dividing method at the end of the training. Formally, given the target sentence S and the group number K, we define the group dividing indice set r as follows:

$$f(m) = \min(1, \sqrt{m\frac{1-c_0^2}{M} + c_0^2}), \qquad (10)$$

$$s \sim \text{Bernoulli}(f(m)), \qquad (11)$$

$$r = \begin{cases} \text{Uniform}(T, K-1) & s = 1 \\ \text{Rand}(T, K-1) & s = 0 \end{cases}, \qquad (12)$$

where $f(m)$ is the progressing function (Platanios et al. 2019), $c_0 > 0$ is set to 0.01, m denotes the training step, Bernoulli($\cdot$) is the Bernoulli distribution with parameter $f(m)$, Uniform(x, y) represents that dividing the sequence with length x into y parts on average, and Rand(x, y) samples y non-duplicate indices from $[1, x]$. Intuitively, a smaller value of $f(m)$ leads to better error detection ability while a larger one encourages the model to generate groups with similar lengths, in other words, better speedup. In this gard, $f(m)$ gradually increases from 0 to 1 automatically during training, which results in a better performance.

Model	Complexity	Acceleration
Autoregressive	$N(D+\Upsilon)+E$	1
Non-autoregressive	$D+\Upsilon+E$	$\approx N$
PNAIC	$\frac{N}{K}(D+\Upsilon)+E$	$\approx K$

Table 1: Complexity and acceleration of different methods.

Invalid Group Deletion Previous curriculum learning-based length prediction helps the captioner deal with incorhent word errors as well as accelerate the decoding effectively, however, the captioner still suffers from errors where incorrect words have already generated. In prior experiment, we observe that most of incoherent errors happen at the early word of each group creates, since it cannot receive enough dependency information. Under this situation, invalid word groups will be easily produced. Toward that end, we prensent a group-wise deletion strategy, which projects a special symbol `DEL` to manifest the current word group is invalid and should be deleted in the final version. To be specific, we integrate extra pseudo repetitive word groups into the training instances to mimic such incorhent cases.

For example, given the description “tulips in a vase”, we first divide it into three word groups “tulips” “in” and “a vase”. Then the second group is copied and appended with the special symbol `DEL` to the end to construct a invalid repetitive group “in `DEL`”. Finally, we insert the repetitive group to the right of the chosen group, which resulting in 4 groups for instance training. More broadly, provided with the group number K and the sentence S, we first divide sentence S into $K-1$ groups, denoted as $\{G^1, \ldots, G^{K-1}\}$, and then build a pseudo invalid group G_r^i by copying the first u words of a randomly chosen G^i and appending special symbol `DEL` to the group end, u is uniformly sampled from $[1, |G^i|]$. Finally, G_r^i is inserted at the right side of G^i to formulate the final training sample G. It is worthy note that inserting such pseudo invalid groups to all training instances will mislead the captioner into a action that always generating then deleting a invalid group, which is not what we want. In this end, we adopt a balancing probability p to determine if construct a invalid group into the training instance.

3.3 Disscusion

Complexity Analysis Here, we first provide a theoretical analysis of time complexity and acceleration of different autoregressive properties. As presented in Table 1, where D denotes the time consumed on the decoder network, *i.e.*, calculating a distribution over the target vocabulary at each time step and Υ denotes the time consumed on searching for top scores, E is the time consumed on image feature encoding, N denotes the average length of caption, and K corresponds to the group number. In practice, (1) D is usually much larger than Υ since the network is deep, and (2) the proportion of E in the total time procedure is pretty small since image feature encoding can be highly parallelized. As

Models	BLEU-1	BLEU-4	METEOR	ROUGE	CIDEr	SPICE	Latency	SpeedUp
Autoregressive models								
NIC-v2 (Vinyals et al. 2015)	-	32.1	25.7	-	99.8	-	-	-
Up-Down (Anderson et al. 2018)	79.8	36.3	27.7	56.9	120.1	21.4	-	-
AoANet† (Huang et al. 2019)	80.2	38.9	29.2	58.8	129.8	22.4	-	-
M2-T†(Cornia et al. 2020)	80.8	39.1	29.2	58.6	131.2	22.6	-	-
AIC†	80.1	38.6	28.9	57.8	128.5	22.1	174ms	1.00×
Non-autoregressive models								
MNIC† (Gao et al. 2019)	75.4	30.9	27.5	55.6	108.1	21.0	-	2.80×
FNIC† (Fei 2019)	-	36.2	27.1	55.3	115.7	20.2	-	8.15×
MIR† (Jason et al. 2018)	-	32.5	27.2	55.4	109.5	20.6	-	1.56×
CMAL† (Guo et al. 2020)	80.3	37.3	28.1	58.0	124.0	21.8	-	13.90×
Partially Non-autoregressive models								
PNAIC† (K = 2)	80.4	38.3	29.0	58.4	129.4	22.2	81ms	2.17×
PNAIC† (K = 5)	80.3	38.1	28.7	58.3	128.5	22.0	49ms	3.59×
PNAIC† (K = 10)	79.9	37.5	28.2	58.0	125.2	21.8	32ms	5.43×

Table 2: Performance comparisons with different evaluation metrics on the MS COCO test set. All values except Latency and SPEED-UP are reported as a percentage (%). "†" denotes the model is based on Transformer architecture. AIC is our implementation of the Transformer-based autoregressive model, which has the same structure as PNAIC and is served as the teacher model for distillation. The SpeedUp values of NAIC models are from the corresponding paper.

mentioned above, we can find that the speed bottleneck of image captioning model performance lies in its autoregressive decoding process.

Unified Perspective By incorporating a group-level parallel generation strategy, we successfully extend the conventional Transformer-based autoregressive captioning model to a partially non-autoregressive captioning model. PNAIC contains a hyperparameter K, and each K value defines a caption decoding process with different degrees of parallelism. In concrete, PNAIC covers AIC and NAIC as its special cases: it reduce to NAIC when K = 1 and to NAIC when $K = N$. In brief, PNAIC is a more general form. A model with a smaller K value is easier to train, achieves higher accuracy, and keep high latency; while that with larger K is harder to train and results in worse accuracy.

4 Experiments

4.1 Experimental Settings

Dataset MS COCO (Chen et al. 2015) is a standard benchmark for the image captioning task. We use the Karpathy split (Karpathy and Fei-Fei 2015) that have been employed extensively for reporting results in prior works. This split contains 113,287 training images equipped with five sentences each, and 5,000 images for validation and test splits, respectively. Following (Huang et al. 2019), all the sentences are converted to lower case, and we omit words which occur less than five times. The vocabulary size is 10,369 words. To be consistent with previous works, we pre-extract image features for all the images following (Anderson et al. 2018).

Implementation Details For model hyperparameters, we follow most of settings in (Vaswani et al. 2017). Specifically, utilizing a base Transformer model (d_{model} = 512, d_h = 512, n_{layer} = 6, n_{head} = 8, $p_{dropout}$ = 0.1) and linearly anneal the learning rate from 3×10^{-4} to 10^{-5}. The AIC model is trained first with XE loss and then with SCST (Rennie et al. 2017). For PNAIC, we utilize the sequence-level distillation (Kim and Rush 2016; Zhou, Neubig, and Gu 2019), which replaces the target sentences in the training dataset with sentences generated by the AIC model, and set the beam size of the technique to 3. In order to accelerate convergence, the encoder of the corresponding AIC model is adopted to initialize the encoder of PNAIC while others are initialized randomly. In addition to accelerating convergence, we find this method also slightly improves the captioning quality. For performance evaluation, we use standard automatic metrics to evaluate the quality of captions, including BLEU-1/4 (Papineni et al. 2002), METEOR (Lavie and Agarwal 2007), ROUGE (Lin 2004), SPICE (Anderson et al. 2016), and CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015). Besides, Latency represents the time to decode a single image averaged over the whole test split, and is tested on a GeForce GTX 1080 Ti GPU.

4.2 Results and Analysis

General Comparisons Table 2 reports the evaluation results of our PNAIC and baselines on image captioning tasks. The compared baselines include both non-autoregressive models and autoregressive models. Among the autoregressive models, AoANet, M2-T, and AIC are based on Transformer architecture as ours, while others are based on

	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Up-Down*	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*	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*	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	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 (K = 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

Table 3: Leaderboard of various methods on the online MS COCO test server. * denotes the ensemble model.

$f(m)$	B4	M	R	C	S	Step
0	38.0	28.6	58.1	128.3	21.9	4.3
0.5	37.9	28.3	58.0	127.9	21.8	3.2
1	37.2	27.8	57.8	124.3	21.2	2.4
Linear	38.1	28.6	58.2	128.2	22.0	3.0
Curriculum	38.1	28.7	58.3	128.5	22.0	2.6

Table 4: Effect of learning to predict the group length evaluated on MS COCO validation set with group number K = 5. $f(m)$ is the ratio parameter in Equation 10. The first three rows indicate that the value of $f(m)$ is constant, while "Linear" denotes annealing $f(m)$ from 0 to 1 linearly. "Step" corresponds to the average number of decoding steps.

p	B4	M	R	C	S	Step
0	37.0	27.5	57.3	122.5	21.0	2.0
0.1	37.8	28.3	57.9	126.8	21.7	2.4
0.3	37.9	28.4	58.0	127.6	21.8	2.6
0.5	38.1	28.7	58.3	128.5	22.0	2.6
0.7	38.0	28.5	58.1	127.9	22.0	2.7
0.9	37.5	28.1	57.8	125.5	21.6	2.8
1.0	37.2	27.8	57.7	124.8	21.3	2.7

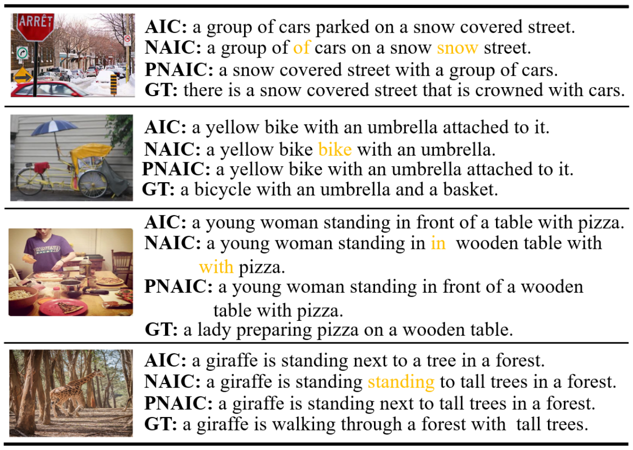
Table 5: Effect of learning to delete the invalid group. Results are evaluated on MS COCO validation set with group number K = 5. p denotes the probability of injecting pseudo repetitive groups to each training instance.

LSTM. For NAIC models, MNIC and MIR adopts an iterative refinement strategy, FNIC orders words detected in the image with an RNN, and CMAL optimzes captions with sentence-level reward. Overall, we can see that our proposed model outperforms all the strong NAIC baselines except CMAL in metric BLEU-1 when K = 10. However, the performance gap is negligible (only about 0.4). Note that our group-based model is one-shot, which is superior in speed to some multi-step refinement method. In particular, not only NAIC, but also we are superised to find that PNAIC surpass the Transformer-based AIC baseline trained from scratch. we attribute it to: 1) PNAIC is fine-tuned on a well-trained AIC model. 2) Combining AIC and NAIC has a better regularization effect. Moreover, we also present the results of the online MS COCO evaluation in Table 3.

Effect of Group Number We test the group size K from $\{2, 5, 10\}$, and the results are listed in the bottom of Table 2. Obviously, we can find that: 1) A larger K has more significant acceleration because fewer autoregressive step are required. 2) As K increase, the performance of PNAIC drops moderately. For example, the CIDEr score drops less than 0.9 when K grows from 2 to 5, and drop no more than 3.3 when K grows further to 10. It indicates that our model has a good trade-off between speed and accuracy.

Effect of Group Length Prediction The curriculum learning-based group length determination is adopted to assist the PNAIC to learn to predict the group length dynamically conditioned on visual context and previously generated words from easy to difficult. Here, we conduct an extensive analysis of the effect of this training task, and the results are presented in Table 4. We can discover that: 1) by incorporating the dynamic length prediction, that is, $f(m) > 0$, all of the evaluation scores are increased, indicating the effectiveness of our training task; 2) As $f(m)$ grows smaller, the average number of decoding step ("Step") increase significantly. The reason is that more training sentences are divided into groups equally with larger $f(m)$ during training, and the model is biased to generate groups with similar lengths. However, if the model is not exposed to randomly divided groups ($f(m)$ = 1), it fails to learn to discriminate incoherent errors, and the CIDEr score drops dramatically. 3) Through utilizing a competence-aware curriculum learning strategy, we achieve a good balance between decoding speed and incoherent issues correction. Thus, we adopt it as the default setting in other experiments.

Effect of Invalid Group Deletion We also investigate the effect of the invalid group deletion, and the performance on various metrics are shown in Table 5, where p represents the probability of integrating pseudo repetitive groups to each training instance. According to the evaluation results, we can observe that: Overall, when p = 0.5, our captioning model achieves the best performance. As p goes from 0.5 to 0 or goes up from 0.5 to 1, the performance drops gradually. In particular, when p = 0, that is, without integrating the group deletion training task, the model obtain lowest CIDEr score, indicating that this deletion task is effective for avoiding from repetitive word errors. On the other hand, the perfor-



AIC: a group of cars parked on a snow covered street.
NAIC: a group of of cars on a snow snow street.
PNAIC: a snow covered street with a group of cars.
GT: there is a snow covered street that is crowned with cars.

AIC: a yellow bike with an umbrella attached to it.
NAIC: a yellow bike bike with an umbrella.
PNAIC: a yellow bike with an umbrella attached to it.
GT: a bicycle with an umbrella and a basket.

AIC: a young woman standing in front of a table with pizza.
NAIC: a young woman standing in in wooden table with with pizza.
PNAIC: a young woman standing in front of a wooden table with pizza.
GT: a lady preparing pizza on a wooden table.

AIC: a giraffe is standing next to a tree in a forest.
NAIC: a giraffe is standing standing to tall trees in a forest.
PNAIC: a giraffe is standing next to tall trees in a forest.
GT: a giraffe is walking through a forest with tall trees.

Figure 3: Examples of the generated captions from AIC, NAIC, and PNAIC models with the same architecture. GT represents a human-annotated ground-truth caption.

mance also drops drastically when $p > 0.7$, we attribute it to: the model is misled that to generate, then delete a repetitive group is expected, meantime, when the pseudo repetitive groups are constructed randomly and universally, it falls into a tight spot to learn the underlying mapping. It can also be proved from the fact that the average number of decoding steps ("Step") increases steadily with p grows.

Qualitative Analysis For more intuitive, we present several examples of generated image captions from AIC, NAIC, and our PNAIC(K = 5), which hold the same model architectures, coupled with human-annotated ground truth sentences (GT) in Figure 3. Compared with As we can be seen, in general, all models hold the capability to reflect the content of the given image accurately, while our model contains more rich attribute information. The incoherent problem, including repeated words and incomplete content, is severe in the sentence generated by pure NAIC, while it can be effectively alleviated by PNAIC, *i.e.*, "bike" in the second sample. This confirms that our proposed training tasks, including the group length prediction and invalid group deletion, can leverage the captioning model to reduce word errors. Other examples also show similar results.

Human Evaluation Since the automatic evaluation metrics do not necessarily correlate with human perception, referring to (Huang et al. 2019), we additionally conducted a human evaluation to compare our PNAIC (K = 5) against two baselines of the same structure, *i.e.*, AIC and NAIC. Specifically, eight humans are invited, and 200 images are selected randomly from the testing set. We show the evaluators each image with three auto-generated sentences plus three human-annotated captions and ask the evaluators: Do the systems produce human-like sentences? Based on the feedback, we calculate the metrics: percentage of captions that are as well as or even better than human annotation. The result scores of PNAIC, AIC, NAIC are 76.3%, 75.8% and 40.2%, respectively. Apparently, our PNAIC is capable of creating high-quality captions.

5 Realted Work

Almost all state-of-the-art image captioning models are autoregressive (Bai and An 2018; Fei 2020a), meaning that the model generates captions word by word from left to right and is not friendly to modern hardware optimized for parallel execution. The pioneering work about parallel generation is (Zheng, Li, and Wang 2019), which generates the selected object words first, and the rests are filled with a two-pass process. Several recent works attempt to accelerate generation by introducing a non-autoregressive image captioning (NAIC) model (Gu et al. 2018; Wei et al. 2019), which produces the entire sentences simultaneously. Although accelerating the decoding process significantly, NAIC models suffer from several repetitive and missing problems. Therefore, more efforts have been devoted to mitigating issues in image captioning. (Fei 2019) reorders words detected in the image with a light RNN to form better latent variables before later decoding. (Jason et al. 2018) introduces an iterative mask refinement strategy to learn the position matching information. (Guo et al. 2020) addresses the inconsistency problem in NAIC with a multi-agent learning paradigm and sentence-level optimization. As far as we are concerned, this is the first work to incorporate partially non-autoregressive solution for the image captioning task and provide a unified perspective about previous AIC and NAIC works.

Most relevant to our proposed method is (Wang, Zhang, and Chen 2018; Ran et al. 2020; Chai and Wan 2020). What we have in common is that both methods have not entirely abandoned autoregressive, but rather shortened the autoregressive path. The difference lies that our decoder is cros-modality, this means that decoding is conditioned on both image and prior txt. Besides, we introduce well-designed curriculums for gradual learning. Other similar works are (Gu, Wang, and Zhao 2019; Stern et al. 2019), which introduce the operations of insertion and deletion to generate an amenable sequence in parallel based on Transformer structure. Different from these work, our model changes one-step non-autoregressive generation to a partial form, which maintains considerable speedup and enables the model to view the local history and future to avoid errors.

6 Conclusion

In this paper, we introduce a novel paradigm, referred to as partially non-autoregressive, for fast and accurate image captioning. Technically, our captioning model generates group-level subsentences non-autoregressively and predicts each word in the group autoregressively. By learning to determine group length and delete invalid groups evolutionarily, the captioning system is capable of eliminating incoherent issues effectively. For an in-depth understanding, we also provide a complexity analysis as well as a unified perspective for various decoding strategies. Experiments on widely-used MS COCO dataset show that the proposed method maintains an advanced performance with more than 3.5× decoding speedup compared with state-of-the-art autoregressive captioning models and obtain a substantial improvement in descriptive quality compared with strong non-autoregressive image captioning models.

References

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *Proc. ECCV*, 382–398.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proc. IEEE CVPR*, 6077–6080.

Bai, S.; and An, S. 2018. A survey on automatic image caption generation. *Neurocomputing* 311: 291–304.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proc. ICML*, 41–48.

Bernardi, R.; Cakici, R.; Elliott, D.; Erdem, A.; Erdem, E.; Ikizler-Cinbis, N.; Keller, F.; Muscat, A.; and Plank, B. 2016. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research* 55: 409–442.

Chai, Z.; and Wan, X. 2020. Learning to Ask More: Semi-Autoregressive Sequential Question Generation under Dual-Graph Interaction. In *Proc. ACL*, 225–237.

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* .

Cornia, M.; Stefanini, M.; Baraldi, L.; and Cucchiara, R. 2020. Meshed-Memory Transformer for Image Captioning. In *Proc. IEEE CVPR*, 10578–10587.

Fei, Z. 2020a. Actor-critic sequence generation for relative difference captioning. In *Proc. ICMR*.

Fei, Z. 2020b. Iterative Back Modification for Faster Image Captioning. In *Proc. ACM Multimedia*, 3182–3190.

Fei, Z.-c. 2019. Fast Image Caption Generation with Position Alignment. *arXiv preprint arXiv:1912.06365* .

Gao, J.; Meng, X.; Wang, S.; Li, X.; Wang, S.; Ma, S.; and Gao, W. 2019. Masked Non-Autoregressive Image Captioning. *arXiv preprint arXiv:1906.00717* .

Gu, J.; Bradbury, J.; Xiong, C.; Li, V. O. K.; and Socher, R. 2018. Non-Autoregressive Neural Machine Translation. In *Proc. ICLR*.

Gu, J.; Wang, C.; and Zhao, J. 2019. Levenshtein transformer. In *Proc. NIPS*, 11181–11191.

Guo, L.; Liu, J.; Zhu, X.; He, X.; Jiang, J.; and Lu, H. 2020. Non-Autoregressive Image Captioning with Counterfactuals-Critical Multi-Agent Learning. *arXiv preprint arXiv:2005.04690* .

Huang, L.; Wang, W.; Chen, J.; and Wei, X.-Y. 2019. Attention on attention for image captioning. In *Proc. IEEE ICCV*, 4634–4643.

Jason, L.; Elman, M.; Neubig, G.; and Kyunghyun, C. 2018. Deterministic Non-Autoregressive Neural Sequence Modeling by Iterative Refinement. In *Proc. EMNLP*, 1138–1149.

Karpathy, A.; and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE CVPR*, 3128–3137.

Kim, Y.; and Rush, A. M. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947* .

Lavie, A.; and Agarwal, A. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proc. ACL Workshop*, 228–231.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of summaries. In *Proc. ACL Workshops*, 74–81.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. J. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. ACL*, 311–318.

Platanios, E. A.; Stretcu, O.; Neubig, G.; Poczos, B.; and Mitchell, T. M. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848* .

Ran, Q.; Lin, Y.; Li, P.; and Zhou, J. 2020. Learning to Recover from Multi-Modality Errors for Non-Autoregressive Neural Machine Translation. *arXiv preprint arXiv:2006.05165* .

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-Critical Sequence Training for Image Captioning. In *Proc. IEEE CVPR*, 1179–1195.

Stern, M.; Chan, W.; Kiros, J.; and Uszkoreit, J. 2019. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249* .

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Proc. NIPS*, 5998–6008.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proc. IEEE CVPR*, 4566–4575.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *Proc. IEEE CVPR*, 3156–3164.

Wang, C.; Zhang, J.; and Chen, H. 2018. Semi-autoregressive neural machine translation. In *Proc. EMNLP*, 479–488.

Wei, B.; Wang, M.; Zhou, H.; Lin, J.; Xie, J.; and Sun, X. 2019. Imitation learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:1906.02041* .

Zheng, Y.; Li, Y.; and Wang, S. 2019. Intention oriented image captions with guiding objects. In *Proc. IEEE CVPR*, 8395–8404.

Zhou, C.; Neubig, G.; and Gu, J. 2019. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint arXiv:1911.02727* .