Neighbours Matter: Image Captioning with Similar Images

Qingzhong Wang[1]
qingzwang2-c@my.cityu.edu.hk
Jiuniu Wang[123]
jiuniwang2-c@my.cityu.edu.hk
Antoni B. Chan[1]
abchan@cityu.edu.hk
Siyu Huang[4]
huangsiyu@baidu.com
Haoyi Xiong[4]
xionghaoyi@baidu.com
Xingjian Li[4]
lixingjian@baidu.com
Dejing Dou[4]
doudejing@baidu.com

[1] Department of Computer Science
City University of Hong Kong
Kowloon, HK

[2] Aerospace Information Research Institute
Chinese Academy of Sciences
Beijing, China

[3] University of Chinese Academy of Sciences
Beijing, China

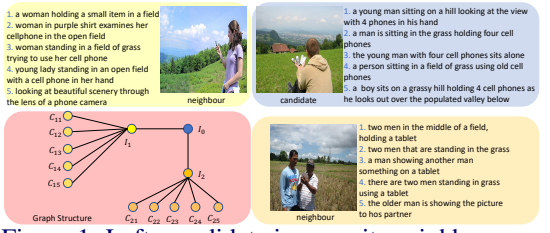
[4] Baidu Research
Beijng, China

Abstract

Most image captioning models aim to generate captions based solely on the input image. However images that are similar to the given input image contain variations of the same or similar concepts as the input image. Thus, aggregating information over similar images could be used to improve image captioning models, by strengthening or inferring concepts that are in the input image. In this paper, we propose an image captioning model based on KNN graphs composed of the input image and its similar images, where each node denotes an image or a caption. An attention-in-attention (AiA) model is developed to refine the node representations. Using the refined features significantly improves the baseline performance, *e.g.*, CIDEr score obtained by the Updown model increases from 120.1 to 125.6. Compared with the state-of-the-art performance, our proposed method obtains 129.3 of CIDEr and 22.6 of SPICE on Karpathy's test split, which is competitive with the models that employ fine-grained image features such as scene graphs and image parsing trees.

1 Introduction

Image captioning is a challenging task that combines computer vision and natural language generation. To achieve the goal of accurately describing images, a wide range of approaches have been developed, most of which pay much attention to the image itself, *i.e.*, using CNN features[3, 38, 40, 41, 44], object-level features [2, 29], object labels and attributes [11, 48], scene graphs [45, 46] and image parsing tree [47]. However, it could be difficult to directly



# Neighbouring images	Max. CIDEr	Avg. CIDEr	Min. CIDEr
1	89.0	51.8	22.1
2	107.6	49.6	12.0
3	118.4	48.6	7.9
4	125.8	47.7	5.9
5	131.1	47.0	4.6

Figure 1: Left: candidate images, its neighbours and the constructed graph. I_0 represents the visual feature of the candidate image, $I_{\{1,2\}}$ represent the visual features of neighbours 1 and 2 and, c_{ij} denotes the representation of the jth caption of the ith neighbouring image. Right: the measurement of useful information using different numbers of neighbours.

translate visual features to concepts [50], *e.g.*, in Figure 1 (left), the concept "use a cell phone" is not easy to be recognized, since there are some ambiguous descriptions, such as "holding a cell phone" and "looking at a cell phone". Yet if we compare the candidate image with its neighbours, the human annotation of the neighbouring images would leak the clue of "use", hence, it could be easier to learn difficult concepts.

To describe a scene, we humans, in particular children who lack knowledge of the scene, generally refer to the descriptions from others or the descriptions of similar scenes. Hence introducing neighbouring images into image captioning models also imitates the ability of humans. Neighbouring images normally contain useful information that benefits image captioning. To measure how much useful information is contained in the neighbours, given an image we first find its top-k nearest neighbours (KNN) based on Euclidean distance in the feature space and then directly use the captions of its neighbours to describe it. Figure 1 (right) shows the CIDEr scores [36] on Karpathy's test split (5,000 images) [19], where Max CIDEr=$\max\{CIDEr(c_{ij}, C_0^{GT}) | i = 1, \cdots, n; j = 1, \cdots, m\}$, and similarly for Min CIDEr. Avg. CIDEr=$\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} CIDEr(c_{ij}, C_0^{GT})$, where and C_0^{GT} denotes the set of ground-truth captions of the candidate image I_0, n and m denote the number of neighbours and the number of captions per neighbour, respectively. The more neighbours we use, the higher the maximum CIDEr we can obtain, indicating more useful information. However, using too many neighbours could introduce noise as well, *e.g.*, the average CIDEr score slightly decreases, therefore, it could be difficult to filter out noisy information.

In this paper, to leverage the useful information underlying similar images and the corresponding human annotations, we first construct KNN graphs, *i.e.*, each image in the dataset has n neighbouring images and each neighbour has m human annotations (see Figure 1). We then use the KNN graph to refine the features of the candidate image using features from the neighbouring images. Each node of the KNN graph is composed of multiple items such as words and objects, thus we need to aggregate the messages from other items in the same node, as well as messages from the neighbouring nodes. In this paper, we propose an attention in attention network (AiA) to refine the candidate image features. The outer attention is used to pass messages over the graph, which is similar to graph attention networks (GATs) [37]. Whereas GATs require that each node in the graph is represented by a vector, in our constructed graphs, each node is a set of vectors. Hence we use an inner attention to refine the node feature. Our proposed AiA feature refiner is a general module that can be plugged into any captioning model and we applied AiA to different baseline models and the experimental results show that AiA is able to improve the baseline performance, *e.g.*, Updown model obtains 120.1 CIDEr score [2], in contrast, using AiA boosts CIDEr score

up to 125.6.

The main contributions of this paper are in threefold. First, we propose a new feature refinement framework that takes similar images and the corresponding human annotations into account, which is different from current feature refinement frameworks that only consider the candidate image itself. Second, we propose an attention in attention network (AiA) to refine features over graph structures, which is a general module that can be plugged into any existing model. Third, we conduct extensive experiments and the results show that our proposed model significantly improves the baseline performance, achieving competitive performance compared with the models that employ image parsing trees to refine features [47], *e.g.*, 129.3 (ours) v.s 130.6 of CIDEr and 22.6 (ours) v.s 22.3 of SPICE on Kaparthy's test split.

2 Related work

End-to-end models dominate the task of image captioning [2, 3, 11, 16, 26, 27, 30, 38, 44, 45, 46, 47, 48]. In [38], a CNN+LSTM framework is proposed, where image features are extracted by a inception network [33] pre-trained on ImageNet [6], and then an LSTM [15] is employed to decode a sentence from the image feature. The connection between CNN and LSTM is a linear transformation, which is simple to learn the correspondence between words and image regions. An attention mechanism is introduced into captioning models by [44], which is able to learn the correspondence between words and image regions. Similarly, [48] applies the attention mechanism to semantics instead of image regions, indicating that the detected concepts play an important role in image captioning. To further improve the performance, [2] uses object-level features provided by Faster-RCNN [13] instead of CNN features. Yao et. al. [46] explore scene graphs [18] in image captioning, where an image is represented by a graph and each node is an object, each edge denotes the relationship between object nodes. Also, [45] use scene graphs for image captioning. Besides using object-level features, [47] employs instance-level features obtained from Mask-RCNN [14] and the image is parsed into a tree structure, where the root is the image, the leaves are the instances and the middle-level nodes denote the object regions. Tree-LSTMs [34] are applied to refine the image features. Another property—diversity of captions also draw much attention [4, 5, 8, 32, 39, 42, 43], which requires a captioning model to generate multiple captions for each image.

In the above related works, researchers pay much attention to obtaining better image representations, *e.g.*, CNN features $\rightarrow$ object-level features $\rightarrow$ scene graphs $\rightarrow$ image parsing trees. However all of these works only consider the candidate image itself but ignore the similar images that could provide useful information for describing the candidate image. J. Devlin et al. explore using the nearest images for image captioning [9], revealing that using the captions of similar images to describe a candidate image could achieve competitive performance compared to existing models, indicating that similar images indeed contain useful information – however, they do not develop a model to leverage the information. Although [10] uses similar images by weighting them, the captions of the similar image are not used. In this paper, we explore using the similar images and their corresponding captions for image captioning.

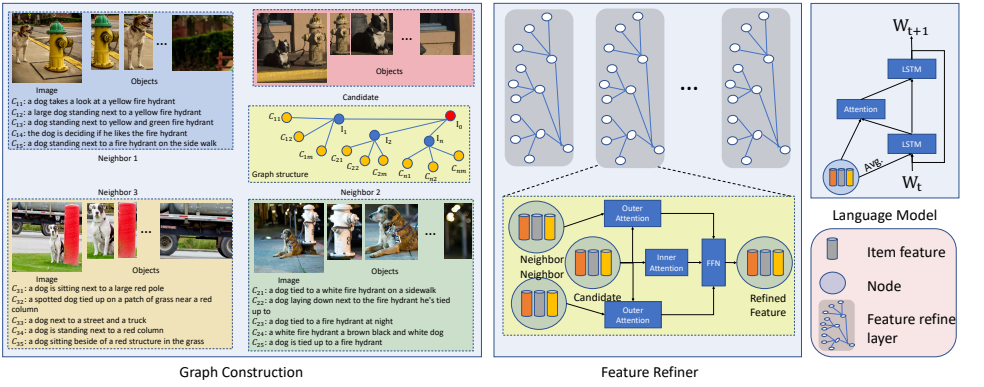


Figure 2: Our proposed model is composed of 3 modules: (1) graph construction, (2) feature refiner, (3) language model. The feature refiner stack a multiple layers of attention-in-attention network, where the outer attention is used to aggregate messages from the neighbouring nodes, and the inner attention is employed to aggregate the items of the same node.

3 Methodology

Notation. Let $I_0 = \{o_1^0, \cdots, o_k^0\}$ be a candidate image, where $o_i^0 \in \mathbf{R}^{D_v}$ denotes the ith object in the candidate image and the ground truth caption $c = \{w_1, \cdots, w_T\}$, where w_i denotes the ith word and T denotes the length of the caption. Define its similar images as $\{(I_1^0, C_1), \cdots, (I_n^0, C_n)\}$ from the training dataset $\mathcal{D}_{train} = \{(I_1, C_1), \cdots, (I_N, C_N)\}$, where $C = \{c_1, \cdots, c_m\}$. A undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is composed of nodes $v \in \mathcal{V}$ and edges $e = (v_i, v_j) \in \mathcal{V} \times \mathcal{V}$. Node v_i is represented by a set of items $\{\eta_1^i, \cdots, \eta_{n_{v_i}}^i\}$, where $\eta_j^i \in \mathbf{R}^{D_v}$ or $\in \mathbf{R}^{D_c}$ based on whether it is a image node or a caption node, D_v and D_c denote the dimensionality of the image and caption embedding spaces, and n_{v_i} denotes the number of items of node v_i. Edges e take binary values, where 0 denotes that two nodes do not connect and 1 denotes there is a connection between two nodes and messages can be passed from one node to its neighbours.

3.1 Graph Construction

In this paper we first construct a KNN graph based on the similarity among images. We use Karpathy's training split of MSCOCO [24] as $\mathcal{D}_{train}$, which contains 113,287 images and each image has 5 captions. For each image, we use Faster-RCNN trained on the Visual Genome dataset [21] to detect k objects, and each object is represented by a vector $o_i \in \mathbf{R}^{D_v}$ provided by the ROI pooling layer. Finally, an image is represented by a vector $\bar{o} = \frac{1}{k}\sum_{i=1}^{k} o_i$ and the distance between two images I_1, I_2 is calculated as follows: $dist(I_1, I_2) = ||\bar{o}_1 - \bar{o}_2||_2$. Given a candidate image I_0, we find n nearest images from $\mathcal{D}_{train}$, not including I_0 itself if it belongs to $\mathcal{D}_{train}$. Note that each image in $\mathcal{D}_{train}$ has m human annotations, which could contain useful information (see Figure 1 (right)), and thus, the constructed KNN graph also takes human annotations into account. In Figure 2, we show an example of the constructed KNN graph, where the node distance between the candidate image I_0 and the captions of similar images is 2.

3.2 Feature Refinement

To refine the image features, we present an attention-in-attention (AiA) model. The outer attention is based on graph attention networks (GATs) [37], and the inner attention is based on self-attention [35], the structure of which is shown as "Feature Refiner" in Figure 2.

Given a KNN graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, there are two types of nodes: (1) image nodes represented by $I_{\{0,1,\cdots,n\}}$, (2) caption nodes represented by $c_{\{11,12,\cdots,nm\}}$, where I_0 is the candidate image without ground-truth captions. Following the graph construction procedure, for each image node, we employ Faster-RCNN to detect k objects and each object is represented by a vector $o_i \in \mathbf{R}^{D_v}$. In terms of caption nodes, we first use a bidirectional LSTM [12] to obtain a representation of a caption, *i.e.*, c_i is represented by $\{s_1^i, \cdots, s_{T_i}^i\}$, where $s_j^i \in \mathbf{R}^{D_c}$ is the jth output of the bidirectional LSTM and T_i denotes the number of words in c_i.

Inner attention. Given a node $v_o = \{\eta_1^o, \cdots, \eta_{n_{v_o}}^o\}$ composed of n_{v_o} items, for the ith item, the inner attention is computed as follows:

$$\hat{\eta}_i^o = \left[\sum_{j=1}^{n_{v_o}} \alpha_{ij}^1 \mathbf{v}_{inner}^1(\eta_j^o), \cdots, \sum_{j=1}^{n_{v_o}} \alpha_{ij}^{n_h} \mathbf{v}_{inner}^{n_h}(\eta_j^o)\right], \tag{1}$$

where $[\cdots]$ denotes concatenation, n_h denotes the number of attention heads, α denotes the attention weights, and $\mathbf{v}_{inner}^h(x) = W_{\mathbf{v}_{inner}}^h x$ denotes a linear transformation and $h = 1, \cdots, n_h$. We define

$$\alpha_{ij}^h = \frac{\exp(a_{ij}^h)}{\sum_{j=1}^{n_{v_i}} \exp(a_{ij}^h)} \quad \text{and} \quad a_{ij}^h = \frac{\mathbf{q}_{inner}^h(\eta_i^o)^{\mathbf{T}} \mathbf{k}_{inner}^h(\eta_j^o)}{\sqrt{D_{qk}^{inner}}}, \tag{2}$$

where $\mathbf{q}_{inner}^h(x) = W_{\mathbf{q}_{inner}}^h x$ and $\mathbf{k}_{inner}^h(x) = W_{\mathbf{k}_{inner}}^h x$ denote query and key functions, respectively. $\mathbf{T}$ represents transpose and D_{qk}^{inner} denotes the dimensionality of the query and key space. Note that node v_i can be either an image node or a caption node.

Outer attention. Outer attention is applied to pass messages over the KNN graph. Given a node $v_o = \{\eta_1^o, \cdots, \eta_{n_{v_o}}^o\}$ and its neighbours $\mathcal{N}_o = \{v_1, \cdots, v_{N_{v_o}}\}$, where n_{v_o} denotes the number of items of v_o, N_{v_o} represents the number of neighbours of v_o, and $v_i = \{\eta_1^i, \cdots, \eta_{n_{v_i}}^i\}, i = 1, \cdots, N_{v_o}$. The outer attention is computed as follows:

$$\tilde{\eta}_{ij}^o = \left[\sum_{p=1}^{n_{v_j}} \alpha_{ijp}^1 \mathbf{v}_{outer}^1(\eta_p^j), \cdots, \sum_{p=1}^{n_{v_j}} \alpha_{ijp}^{n_h} \mathbf{v}_{outer}^{n_h}(\eta_p^j)\right], \tag{3}$$

$$\bar{\eta}_i^o = M\left(\tilde{\eta}_{ij}^o | j = 1, 2, \cdots, N_{v_o}\right), \tag{4}$$

where $\tilde{\eta}_{ij}^o$ denotes the message comes from the jth neighbour for the ith item of node v_o, $\mathbf{v}_{outer}^h(x) = W_{\mathbf{v}_{outer}}^h x$ denotes a linear transformation. $\bar{\eta}_i^o$ denotes the aggregated message for the ith item of node v_o, and $M(\cdot)$ denotes the message aggregation function, such as $\max(\cdot)$ and gated sum function $gate(x_1, \cdots, x_n) = \sum_{i=1}^n \sigma(W_a x_i) \odot (W_b x_i)$, where W_a, W_b are learnable parameters. Similar to (2) the attention weights are calculated by:

$$\alpha_{ijp}^h = \frac{\exp(a_{ijp}^h)}{\sum_{p=1}^{n_{v_j}} \exp(a_{ijp}^h)} \quad \textit{and} \quad a_{ijp}^h = \frac{\mathbf{q}_{outer}^h(\eta_i^o)^{\mathbf{T}} \mathbf{k}_{outer}^h(\eta_p^j)}{\sqrt{D_{qk}^{outer}}}, \tag{5}$$

where $\mathbf{q}_{outer}$ and $\mathbf{k}_{outer}$ represent query and key functions, which are both linear functions, and D_{qk}^{outer} denotes the dimensionality of the query and key space.

Feed-forward networks. Like transformer [35], we use a feed-forward network (FFN) to further refine the attention features:

$$\eta_i^o|_{l+1} = ReLU\left(FFN(\hat{\eta}_i^o|_l + \bar{\eta}_i^o|_l) + \eta_i^o|_l\right), \tag{6}$$

where $\eta_i^o|_l$ denotes the lth layer representation of the ith item in node v_o. If $\eta_i^o|_0$ is from Faster-RCNN or bidirectional LSTM depends on whether v_o is an image node or a caption node. $\hat{\eta}_i^o|_l$ and $\bar{\eta}_i^o|_l$ are computed using (1) and (4), respectively. And $FFN(\cdot) = Seq(IN, FC, ReLU, IN, FC)$, where IN denotes instance normalization layer and FC represents fully-connected layer. We can stack multiple layers to obtain the final representations of the nodes.

3.3 Language Model, Training and Inference

In Figure 2, we show the structure of the Updown language model [2]. For the candidate image I_0, the refined representation is $v_0|_l = \{\eta_1^0|_l, \cdots, \eta_k^0|_l\}$ and $\bar{v}_0 = \frac{1}{k}\sum_{i=1}^{k}\eta_i^0|_l$. The bottom LSTM in the Updown model takes $\{\bar{v}_0, w_t, h_{t-1}^2\}$ as input in the tth step, where w_t is the tth word and h_{t-1}^2 is the $(t-1)$th output of the top LSTM. The top LSTM takes $\{h_t^1, Att(v_0|_l, h_t^1)\}$ as input and the output h_t^2 is applied to predict next word w_{t+1}, where $Att(\cdot,\cdot)$ denotes the attention module [2]. Note that the proposed feature refinement module can be applied to any language model.

To train the model, the cross-entropy loss is applied, which is defined as:

$$\mathcal{L}_{XE} = -\sum_{t=1}^{T} \log p\left(w_t | w_{1:t-1}, I\right), \tag{7}$$

where w_t represents the tth word of the ground-truth caption of image I. To further improve the performance, we can directly optimize CIDEr [36] score using reinforcement learning [30] and we define the loss as follows:

$$\mathcal{L}_{RL} = -\mathbf{E}_{c^* \sim p_\theta}\left[CIDEr(c^*)\right], \tag{8}$$

where c^* denotes the caption sampled from the model p_θ, $CIDEr(c^*)$ is the CIDEr score of c^* and $\mathbf{E}[\cdot]$ represent expectation. The gradient can be approximated as:

$$\bigtriangledown_\theta \mathcal{L}_{RL} = -(CIDEr(c*) - b) \bigtriangledown_\theta \log p_\theta(c^*), \tag{9}$$

where b represents the baseline that is able to reduce the variance of the gradient. In this paper $b = \frac{1}{n_s}\sum_{i=1}^{n_s} CIDEr(c_i^*)$, where c_i^* is the ith sampled caption, which could be different from c^*. Note that in [30] $b = CIDEr(c^g)$, where c^g denotes the caption obtained by greedy search, however, in the beginning of RL, $CIDEr(c^*)$ is generally lower than $CIDEr(c^g)$, hence the samples are suppressed in most cases. In contrast, using $b = \frac{1}{n_s}\sum_{i=1}^{n_s} CIDEr(c_i^*)$ can mitigate this problem.

During inference, given a test image I, we first find its n similar images from $\mathcal{D}_{train}$ and then use the proposed AiA module to refine the image feature, finally the language model is applied to decode a caption from the refined feature.

Model	Cross-entropy loss					CIDEr-D optimization				
	B-4	M	R	C	S	B-4	M	R	C	S
FC [30]	30.0	25.2	52.9	96.1	-	32.4	25.6	54.7	106.6	-
Updown [2]	36.2	27.0	56.4	113.5	20.3	36.3	27.7	56.9	120.1	21.4
RFNet [17]	35.8	27.4	56.8	112.5	20.5	36.5	27.7	57.3	121.9	21.2
GCN-LSTM [40]	36.8	27.9	57.0	116.3	20.9	38.2	28.5	58.3	127.6	22.0
GCN-LSTM‡ [40]	37.1	28.1	57.2	117.1	21.1	38.3	28.6	58.5	128.7	22.1
SGAE [43]	-	-	-	-	-	38.4	28.4	58.6	127.8	22.1
SGAE‡ [43]	-	-	-	-	-	39.0	28.4	58.9	129.1	22.2
GCN-LSTM+HIP‡ [42]	38.0	28.6	57.8	120.3	21.4	39.1	28.9	59.2	130.6	22.3
AoA [16]	37.2	28.4	57.5	119.8	21.3	38.9	29.2	58.8	129.8	22.4
AoA⋆ [16, 30]	36.9	-	57.3	118.4	21.6	39.1	-	58.9	128.9	22.7
FC-base† [30]	32.8	25.9	54.4	101.4	18.9	33.9	26.2	55.6	110.7	19.4
Updown-base† [2]	36.4	27.7	56.6	113.6	20.7	37.3	27.9	57.8	123.3	21.3
AoA-base† [16]	36.4	27.9	56.7	115.1	21.0	37.9	28.5	58.1	125.5	22.2
FC-I9-c5 (ours)	33.7	26.3	54.9	105.4	19.3	34.7	26.9	56.1	115.6	20.0
Updown-I9-c5 (ours)	36.3	27.8	56.8	114.3	20.9	37.7	28.2	58.0	125.6	21.6
AoA-I9-c5 (ours)	36.3	28.0	56.9	115.4	21.2	38.3	28.6	58.3	127.0	22.5
AoA-I9-c5‡ (ours)	-	-	-	-	-	39.1	28.9	58.9	129.3	22.6

Table 1: Performance on Karpathy's test split. We use 3-layer AiA to refine image features. ‡ denotes ensemble model, ⋆ denotes using the publicly available pre-trained model, † denotes the models trained under our experimental settings. I9-c5 means that for a candidate image, we use 9 similar images and each similar image has 5 ground-truth captions.

4 Experiments

4.1 Implementation Details

We conduct all experiments on the MSCOCO dataset [24], which contains 123,287 images (82,783 for training and 40,504 for validation) and each image has 5 human annotations. We use Karpathy's split to train and test the models, *i.e.*, 113,287 image for training, 5,000 for validation and 5,000 for testing. All captions are used to build the dictionary, and we omit the word that occurs less than 5 times, resulting in a dictionary composed of 10,369 words. For each image we use Faster-RCNN to detect 36 objects and each object is represented by a 2048-D vector, which is provided by the ROI pooling layer of Faster-RCNN. The dimensionality of the word embedding space is 300.

In our implementation $D_v = 2048$, $D_c = 1024$, the number of LSTM hidden units is 1024, and the number of captions for each similar image is $m = 5$. To train the model, we set the batch size to 128 and use cross-entropy loss to train the model for 30 epochs. During training Adam [20] optimizer with a initial learning rate 5×10^{-4} and annealed by 0.8 every 3 epochs is employed. After that, we train the model using RL for another 20 epochs using Adam with a fixed learning rate 5×10^{-5}.

The evaluations metrics we use are BLEU (B-1,2,3,4) [28], METEOR (M) [2], ROUGEL (R) [23], CIDEr (C) [36] and SPICE (S)[1].

4.2 Quantitative and Qualitative Results

Table 1 shows the performance of our proposed method on Karpathy's test split. The models with AiA refined features have improved performance over their counterparts that employ the original Faster-RCNN features. For example, FC-base [30] model trained by cross-entropy loss obtains 101.4 of CIDEr, while using AiA refined features with 9 similar images, each

Model	B-1		B-2		B-3		B-4		M		R		C	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
Updown [2]	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
RFNet [11]	80.4	95.0	64.9	89.3	50.1	80.1	38.0	69.2	28.2	37.2	58.2	73.1	122.9	125.1
GCN-LSTM [46]	80.8	95.2	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
GCN-LSTM+HIP [46]	81.6	95.9	66.2	90.4	51.5	81.6	39.3	71.0	28.8	38.1	59.0	74.1	127.9	130.2
CAVP [45]	80.1	94.9	64.7	88.8	50.0	79.7	37.9	69.0	28.1	37.0	58.2	73.1	121.6	123.8
SGAE [45]	80.6	95.0	65.0	88.9	50.1	79.6	37.8	68.7	28.1	37.0	58.2	73.1	122.7	125.5
AoA-I9-c5(ours)	80.1	94.2	64.7	88.1	50.2	79.1	38.2	68.5	28.5	37.4	58.1	73.0	123.5	125.0

Table 2: Performance on the online MSCOCO test server.

GT: a small elephant stands alone in an enclosure
GT: an elephant in a fenced in enclosure packed with dirt
GT: A small elephant stands alone in an enclosure
FC: a couple of elephants standing next to a fence
Updown: a baby elephant standing in front of a zoo
AoA: a baby elephant standing in front of a zoo
FC-I9-c5: a baby elephant walking on a dirt road
Updown-I9-c5: a baby elephant standing in front of a zoo
AoA-I9-c5: a baby elephant walking in the dirt in a zoo

GT: a couple of men riding horses down a street with tall buildings
GT: a man dressed in red riding a horse through town
GT: people in costume riding down the road on horses
FC: a group of people riding horses down a city street
Updown: a man riding a horse down a city street
AoA: two men riding horses down a city street
FC-I9-c5: a group of people riding horses down a city street
Updown-I9-c5: a man riding a horse down a city street
AoA-I9-c5: two men riding horses in a parade

GT: a cat sitting by a window watching the rain
GT: a cat sitting on a sill looking out a window
GT: a close up of a cat on a window sill looking out the window
FC: a cat is sitting on top of a window
Updown: a black cat sitting in front of a window
AoA: a black cat looking out of a window
FC-I9-c5: a cat standing in front of a window
Updown-I9-c5: a black cat sitting in front of a window
AoA-I9-c5: two black cats sitting on a window sill

GT: two people holding surf boards on a beach
GT: a couple people that are walking on a beach
GT: two people with surfboards standing by the water
FC: two people walking on the beach with surfboards
Updown: a couple of people walking on the beach with surfboards
AoA: two people walking on the beach with surfboards
FC-I9-c5: a couple of people walking on the beach with a surfboard
Updown-I9-c5: two surfers are walking on the beach with surfboards
AoA-I9-c5: two surfers walking on the beach with surfboards

Figure 3: Examples of the generated captions.

of which has 5 captions, FC-I9-c5 obtains 105.4 of CIDEr, and SPICE score increases from 18.9 to 19.3 as well. In terms of CIDEr optimization, using refined features also improves the CIDEr score of FC model by 4.4%. When it comes to other models, such as Updown[1] [2] and AoA[2] [16], AiA refined features are capable of consistently improving CIDEr and SPICE scores, *e.g.*, 123.3 v.s 125.6 of CIDEr for Updown model without and with AiA refined features, 125.5 v.s 127.0 of CIDEr for AoA model without and with AiA refined features [3].

Compared with the state-of-the-art performance [45, 46, 47], our proposed model obtains competitive performance based on CIDEr, *e.g.*, the single AoA-I9-c5 model obtains 127.0, and GCN-LSTM [46] and SGAE [45] obtain 127.6 and 127.8, respectively. In contrast, the proposed single model AoA-I9-c5 achieves 22.5 of SPICE, which beats all the counterparts. In terms of the ensemble model (we ensemble 4 models in this paper), our proposed ensemble AoA-I9-c5 model achieves 129.3 of CIDEr, which performs slightly better than the ensemble GCN-LSTM (128.7) and SGAE (129.1). Although GCN-LSTM+HIP, which employs image parsing trees, achieves better performance on CIDEr (130.6 v.s 129.3), our proposed model obtains a higher SPICE score (22.6 v.s 22.3). Normally, SPICE score reflects the similarity between the scene graph provided by a generated caption and the scene graph obtained from human annotations, which takes the relationships between objects into account, and SPICE has relatively strong correlation to human judgment [1]. Hence, a higher SPICE score could indicate that a model is able to recognize relationships between objects. Looking at the differences between the proposed model and its counterparts, AiA considers each pair of objects in an image, the objects in the similar images and the descriptions to similar images.

[1] We use the publicly available code from this repository: https://github.com/ruotianluo/self-critical.pytorch.

[2] We use the code released by the authors in the repository: https://github.com/husthuaan/AoANet, but the training setting is different from [16]. Our experimental settings are in Section 4

[3] Note that for fair comparison, we train the baseline models and our proposed models under the same settings, thus the metric scores could be different from those reported in the published papers [2, 16, 30]

Model	B-1	B-2	B-3	B-4	M	R	C	S
Updown-I1-c5	79.1	63.2	48.6	36.7	27.8	57.4	122.3	21.4
Updown-I5-c5	80.0	64.3	49.7	37.8	28.1	58.0	125.3	21.7
Updown-I9-c5	80.0	64.2	49.6	37.7	28.2	58.0	125.6	21.6

Table 3: The influence of using different numbers of neighbours. A 3-layer AiA is employed to refine image features and all models are trained with CIDEr optimization. I*n*-c*m* represents using *n* similar images and each similar image has *m* human annotations.

Model	# layers	B-1	B-2	B-3	B-4	M	R	C	S
Updown-I5	1	79.9	64.1	49.4	37.4	28.0	57.9	123.4	21.5
	2	79.9	64.2	49.6	37.8	28.1	58.0	124.7	21.6
	3	80.2	64.4	50.0	38.1	28.2	58.1	125.6	21.7
	4	79.9	64.3	50.0	38.1	28.2	58.2	125.5	21.7
Updown-I5-c5	2	79.7	63.9	49.4	37.6	28.1	57.9	124.0	21.5
	3	80.0	64.3	49.7	37.8	28.1	58.0	125.3	21.7
	4	80.0	64.3	49.7	37.7	28.1	57.9	125.0	21.6

Table 4: The influence of stacking different numbers of AiA layers. I*n* means only using visual information of similar images and I*n*-c*m* means using both visual and descriptive information of similar images.

Thus the representation of each object is refined via considering the context and it is believed that context could benefit object and relationship recognition [49, 50].

Our proposed model also obtains competitive performance on MSCOCO online test server (see Table 2). Compared with SGAE [45], which employs scene graphs, the proposed model achieves 123.5 of CIDEr using 5 ground-truth captions, which performs better than the counterparts. Using 40 ground-truth captions, the proposed model performs slightly worse than SGAE. The possible reason is that using too many visually similar images and their corresponding human annotation could introduce noisy and ambiguous descriptions (see Figure 1 (right)).

Figure 3 shows examples of captions generated by different models. Our proposed model is able to generate semantically correct words that do not occur in the ground-truth captions, *e.g.*, "walking", "parade", "window sill" and "surfers", which could slightly reduce BLUE and CIDEr that are based on the overlap between captions. However, these words are able to describe the images and to some extent, they could benefit scene understanding. *E.g.*, looking at the word "parade", we could imagine the scene of many people and happiness. Also, "surfers" normally inspires us to imagine sea and beach, which could tell us more information than the word "people".

4.3 Ablation Study

In this section conduct ablation studies on the number of neighbours, number of layer of AiA, and the message aggregation functions.

Different numbers of neighbours. Table 3 shows the performance of using different numbers of neighbours. Using 9 neighbouring images achieves 125.6 of CIDEr, which is better than using 5 and 1 neighbouring images. In contrast, using 5 neighbours obtains the highest SPICE score (21.7) and BLEU scores, which indicates that employing more neighbours does not mean better performance, since using too many neighbours leads to noisy and ambiguous descriptions. Moreover, applying more neighbours takes more time to train a model.

Different numbers of AiA layers. Table 4 shows the performance of models that use differ-

Model	B-1		B-2		B-3		B-4		M		R		C		S	
	gate	max	gate	max	gate	max	gate	max	gate	max	gate	max	gate	max	gate	max
AoA-I1	79.6	79.7	64.0	64.2	49.6	49.8	37.7	37.9	28.5	28.6	58.1	58.2	125.3	126.8	22.1	22.4
AoA-I5	79.9	79.6	64.1	64.1	49.9	49.7	38.0	37.9	28.5	28.5	58.1	58.1	125.9	126.3	22.3	22.3
AoA-I9	79.5	79.6	64.1	64.1	49.8	49.8	37.9	37.8	28.5	28.5	58.0	58.0	125.7	125.4	22.4	22.2
AoA-I1-c5	79.7	79.7	64.2	64.3	49.9	50.0	38.1	38.1	28.6	28.5	58.2	58.2	126.4	126.3	22.3	22.2
AoA-I5-c5	79.8	80.0	64.3	64.3	49.9	49.8	38.0	37.8	28.6	28.5	58.2	58.1	126.5	126.2	22.3	22.3
AoA-I9-c5	79.7	79.9	64.3	64.3	50.1	49.9	38.3	38.0	28.6	28.5	58.3	58.2	127.0	125.6	22.5	22.1

Table 5: The influence of message aggregation functions.

ent numbers of AiA layers. Stacking more AiA layers could improve the performance *e.g.*, the 1-layer AiA obtains 123.4 of CIDEr, while the 4-layer AiA achieves 125.5. Interestingly, using a 4-layer AiA slightly reduces CIDEr and SPICE, which could be because stacking more graph attention layers could lead to over smoothing [22], *i.e.*, the representations of nodes tend to be similar, hence, the nodes could become less distinguishable. In addition, stacking too many AiA layers is time-consuming and costs more computational resources.

Different message aggregation functions. Message aggregation function plays an important role in graph neural networks [22]. In this paper we explore two aggregation functions (1) gated aggregation $gate(x_1,\cdots,x_n) = \sum_{i=1}^{n} \sigma(W_a x_i) \odot (W_b x_i)$, where W_a, W_b are learnable parameters, $\sigma(\cdot)$ is sigmoid function and (2) maximum aggregation $max(x_1,\cdots,x_n) = MP(x_1,\cdots,x_n)$, where MP denotes max-pooling operation. Table 5 shows the performance of models that use different aggregation functions. The maximum aggregation function could benefit the models that employ fewer neighbours, *e.g.*, AoA-I1 obtains 126.8 of CIDEr using maximum aggregation, while it gradually decreases to 125.4 with the increase of neighbours. In contrast, using gated aggregation function has a different trend: the performance improves with the increase of neighbours, which is because gated aggregation function introduces more learnable parameters, thus it is able to model more complicated graphs.

5 Conclusion

In this paper we proposed a framework that is able to employ visually similar images for image captioning, as well as an attention-in-attention (AiA) model to refine the candidate image features using the information from its neighbours, which significantly improves the baseline performance. In the future, one possible research direction is exploring different methods to construct KNN graphs, such as using semantic similarity between images. Furthermore, using KNN graphs and graph attention mechanism could be time-consuming, thus another possible direction could be to speed up the model. In addition, hubness is common in KNN graphs, which could lead to lack of distinctiveness of the generated captions. Therefore, another possible research direction could be reducing hubness to generate more distinctive captions.

Acknowledgment

This work was partially finished when Qingzhong Wang did the internship in Baidu Research. The work is also supported by a Strategic Research Grant from City University of Hong Kong (Project NO. 7004682 and 7005218). We are grateful for the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, 2016.

[2] 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 vqa. In *CVPR*, 2018.

[3] Jyoti Aneja, Aditya Deshpande, and Alexander Schwing. Convolutional image captioning. In *CVPR*, 2018.

[4] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, control and tell: A framework for generating controllable and grounded captions. In *CVPR*, June 2019.

[5] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional gan. In *ICCV*, 2017.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[7] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *EACL Workshop*, 2014.

[8] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David Forsyth. Fast, diverse and accurate image captioning guided by part-of-speech. In *CVPR*, June 2019.

[9] Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C. Lawrence Zitnick. Exploring nearest neighbor approaches for image captioning. *arXiv*, 2015.

[10] Guiguang Ding, Minghai Chen, Sicheng Zhao, Hui Chen, Jungong Han, and Qiang Liu. Neural image caption generation with weighted training and reference. *Cognitive Computation*, (11):763–777, 2019.

[11] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *CVPR*, 2017.

[12] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *ICANN*, pages 799–804. Springer, 2005.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, pages 4634–4643, 2019.

[17] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *ECCV*, pages 499–515, 2018.

[18] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Ayman Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *CVPR*, 2015.

[19] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[20] Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[22] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *ICLR*, 2016.

[23] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *ACL Workshop*, 2004.

[24] 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. In *ECCV*, 2014.

[25] Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for sequence-level image captioning. In *ACM MM*, pages 1416–1424, 2018.

[26] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, 2017.

[27] Ruotian Luo, Brian Price, Scott Cohen, and Gregory Shakhnarovich. Discriminability objective for training descriptive captions. In *CVPR*, 2018.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[29] Marco Pedersoli, Thomas Lucas, Cordelia Schmid, and Jakob Verbeek. Areas of attention for image captioning. In *ICCV*, 2017.

[30] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[31] F. Sammani and L. Melas-Kyriazi. Show, edit and tell: a framework for editing image captions. In *CVPR*, 2020.

[32] Rakshith Shetty, Marcus Rohrbach, and Lisa Anne Hendricks. Speaking the same language: Matching machine to human captions by adversarial training. In *ICCV*, 2017.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[34] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, pages 1556–1566, 2015.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[36] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[37] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[38] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[39] Liwei Wang, Alexander Schwing, and Svetlana Lazebnik. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In *NIPS*, 2017.

[40] Qingzhong Wang and Antoni B. Chan. Cnn+cnn: Convolutional decoders for image captioning. *arXiv*, 2018.

[41] Qingzhong Wang and Antoni B. Chan. Gated hierarchical attention for image captioning. In *ACCV*, 2018.

[42] Qingzhong Wang and Antoni B. Chan. Describing like humans: on diversity in image captioning. In *CVPR*, June 2019.

[43] Zhuhao Wang, Fei Wu, Weiming Lu, Jun Xiao, Xi Li, Zitong Zhang, and Yueting Zhuang. Diverse image captioning via grouptalk. In *AAAI*. AAAI Press, 2016.

[44] 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 *ICML*, 2015.

[45] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, pages 10685–10694, 2019.

[46] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, pages 684–699, 2018.

[47] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Hierarchy parsing for image captioning. In *ICCV*, pages 2621–2629, 2019.

[48] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[49] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *CVPR*, pages 5831–5840, 2018.

[50] Junjie Zhang, Qi Wu, Jian Zhang, Chunhua Shen, and Jianfeng Lu. Mind your neighbours: Image annotation with metadata neighbourhood graph co-attention networks. In *CVPR*, pages 2956–2964, 2019.